Emilio Corchado Václav Snášel
Ajith Abraham Michał Woźniak
Manuel Graña Sung-Bae Cho (Eds.)

LNAI 7208

Hybrid Artificial Intelligent Systems

7th International Conference, HAIS 2012
Salamanca, Spain, March 2012
Proceedings, Part I

Part I

HAIS 2012

Springer

Lecture Notes in Artificial Intelligence 7208

Subseries of Lecture Notes in Computer Science

Emilio Corchado Václav Snášel
Ajith Abraham Michał Woźniak
Manuel Graña Sung-Bae Cho (Eds.)

Hybrid Artificial Intelligent Systems

7th International Conference, HAIS 2012
Salamanca, Spain, March 28-30, 2012
Proceedings, Part I

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Emilio Corchado
University of Salamanca, Spain
E-mail: escorchado@usal.es

Václav Snášel
VŠB-TU Ostrava, Czech Republic
E-mail: vaclav.snasel@vsb.cz

Ajith Abraham
Machine Intelligence Research Labs, Washington, DC, USA
E-mail: ajith.abraham@ieee.org

Michał Woźniak
Wroclaw University of Technology, Poland
E-mail: michal.wozniak@pwr.wroc.pl

Manuel Graña
University of the Basque Country, San Sebastian, Spain
E-mail: ccpgrrom@si.ehu.es

Sung-Bae Cho
Yonsei University, Seoul, Korea
E-mail: sbcho@cs.yonsei.ac.kr

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-28941-5 e-ISBN 978-3-642-28942-2
DOI 10.1007/978-3-642-28942-2

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012933241

CR Subject Classification (1998): I.2, H.3, F.1, H.4, I.4, I.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume of Lecture Notes in Artificial Intelligence (LNAI) includes accepted papers presented at the 7th HAIS held in the beautiful and historic city of Salamanca, Spain, in March 2012.

The International Conference on Hybrid Artificial Intelligence Systems (HAIS 2012) has become a unique, established and broad interdisciplinary forum for researchers and practitioners who are involved in developing and applying symbolic and sub-symbolic techniques aimed at the construction of highly robust and reliable problem-solving techniques to present the most relevant achievements in this field.

Hybridization of intelligent techniques, coming from different computational intelligence areas, has become popular because of the growing awareness that such combinations frequently perform better than the individual techniques such as neurocomputing, fuzzy systems, rough sets, evolutionary algorithms, agents and multiagent Systems, among others.

Practical experience has indicated that hybrid intelligence techniques might be helpful to solve some of the challenging real-world problems. In a hybrid intelligence system, a synergistic combination of multiple techniques is used to build an efficient solution to deal with a particular problem. This is, thus, the setting of HAIS conference series, and its increasing success is proof of the vitality of this exciting field.

HAIS 2012 received 293 technical submissions. After a rigorous peer-review process, the International Program Committee selected 118 papers, which are published in these conference proceedings. In this edition a special emphasis was put on the organization of special sessions and workshops. Eight special sessions and one workshop, containing 67 papers in total, were organized on the following topics:

Special Sessions:

- Systems, Man, and Cybernetics by HAIS
- Methods of Classifier Fusion
- HAIS for Computer Security
- Data Mining: Data Preparation and Analysis
- Hybrid Artificial Intelligence Systems in Management of Production Systems
- Hybrid Artificial Intelligent Systems for Ordinal Regression
- Hybrid Metaheuristics for Combinatorial Optimization and Modelling Complex Systems
- Hybrid Computational Intelligence and Lattice Computing for Image and Signal Processing

Workshops:

- Nonstationary Models of Pattern Recognition and Classifier Combinations

The selection of papers was extremely rigorous in order to maintain the high quality of the conference, and we would like to thank the Program Committee for their hard work in the reviewing process. This process is very important to the creation of a conference of high standard and the HAIS conference would not exist without their help.

The large number of submissions is certainly not only testimony to the vitality and attractiveness of the field but an indicator of the interest in the HAIS conferences themselves.

HAIS 2012 enjoyed outstanding keynote speeches by distinguished guest speakers: Tom Heskes, Radboud Universiteit Nijmegen (The Netherlands) and Xindong Wu, University of Vermont (USA).

HAIS 2012 teamed up with the *Neurocomputing* (Elsevier) and the *Applied Soft Computing* (Elsevier) journals for special issues and fast-track publication including selected papers from the conference.

Particular thanks also go to the conference main sponsors, IEEE-Sección España, IEEE Systems, Man and Cybernetics -Capítulo Español, AEPIA, World Federation of Soft Computing, MIR Labs, IT4Innovation Centre of Excellence, The International Federation for Computational Logic, Ministerio de Economía y Competitividad, Junta de Castilla y León, Ayuntamiento de Salamanca, University of Salamanca, who jointly contributed in an active and constructive manner to the success of this initiative. We also want to extend our warm gratitude to all the Special Session and Workshop Chairs for their continuing support of the HAIS series of conferences.

We would like to thank Alfred Hofmann and Anna Kramer from Springer for their help and collaboration during this demanding publication project.

March 2012

Emilio Corchado
Václav Snášel
Ajith Abraham
Michał Woźniak
Manuel Graña
Sung-Bae Cho

Organization

Honorary Chairs

Alfonso Fernández Mañueco	Mayor of Salamanca
Antonio Bahamonde	President of the Spanish Association for Artificial Intelligence (AEPIA)
Pilar Molina	Chair IEEE Spanish Section
Hojjat Adeli	The Ohio State University, USA
Manuel Castro	Past Chair IEEE Spanish Section

General Chair

Emilio Corchado	University of Salamanca, Spain

International Advisory Committee

Ajith Abraham	Machine Intelligence Research Labs, Europe
Antonio Bahamonde	President of the Spanish Association for Artificial Intelligence, AEPIA
Andre de Carvalho	University of São Paulo, Brazil
Sung-Bae Cho	Yonsei University, Korea
Juan M. Corchado	University of Salamanca, Spain
José R. Dorronsoro	Autonomous University of Madrid, Spain
Michael Gabbay	King's College London, UK
Ali A. Ghorbani	UNB, Canada
Mark A. Girolami	University of Glasgow, UK
Manuel Graña	University of the Basque Country, Spain
Petro Gopych	Universal Power Systems USA-Ukraine LLC, Ukraine
Jon G. Hall	The Open University, UK
Francisco Herrera	University of Granada, Spain
César Hervás-Martínez	University of Córdoba, Spain
Tom Heskes	Radboud University Nijmegen, The Netherlands
Dusan Husek	Academy of Sciences of the Czech Republic, Czech Republic
Lakhmi Jain	University of South Australia, Australia
Samuel Kaski	Helsinki University of Technology, Finland
Daniel A. Keim	University of Konstanz, Germany
Isidro Laso	D.G. Information Society and Media, European Commission

Marios Polycarpou	University of Cyprus, Cyprus
Witold Pedrycz	University of Alberta, Canada
Václav Snášel	VSB-Technical University of Ostrava, Czech Republic
Xin Yao	University of Birmingham, UK
Hujun Yin	University of Manchester, UK
Michał Woźniak	Wroclaw University of Technology, Poland
Aditya Ghose	University of Wollongong, Australia
Ashraf Saad	Armstrong Atlantic State University, USA
Bernadetta Kwintiana	Universität Stuttgart, Germany
Fanny Klett	German Workforce Advanced Distributed Learning Partnership Laboratory, Germany
Ivan Zelinka	VSB-Technical University of Ostrava, Czech Republic

Industrial Advisory Committee

Rajkumar Roy	The EPSRC Centre for Innovative Manufacturing in Through-life Engineering Services, UK
Amy Neustein	Linguistic Technology Systems, USA
JaydipSen	Innovation Lab, Tata Consultancy Services Ltd., India

Program Committee

Emilio Corchado	University of Salamanca, Spain (Co-chair)
Václav Snášel	VSB-Technical University of Ostrava, Czech Republic (Co-chair)
Ajith Abraham	Machine Intelligence Research Labs, Europe (Co-chair)
Michał Woźniak	Wroclaw University of Technology, Poland (Co-chair)
Manuel Grana	University of the Basque Country/EHU, Spain (Co-chair)
Sung-Bae Cho	Yonsei University, Korea (Co-chair)
Abdel-Badeeh M. Salem	Ain Shams University, Egypt
Aboul Ella Hassanien	Cairo University, Egypt
Adolfo Rodríguez	University of León, Spain
Alberto Fernández	Universidad Rey Juan Carlos, Spain
Alberto Ochoa	Juarez City University, Mexico
Aldo Franco Dragoni	Università Politecnica delle Marche, Italy
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria, Italy
Alicia Troncoso	Pablo de Olavide University, Spain
Álvaro Herrero	University of Burgos, Spain
Amelia Zafra	University of Córdoba, Spain

Ana M. Bernardos	Universidad Politécnica de Madrid, Spain
Ana María Madureira	Polytechnic University of Porto, Portugal
Anca Gog	University of Babes-Bolyai, Romania
André de Carvalho	University of São Paulo, Brazil
Andreea Vescan	University of Babes-Bolyai, Romania
Andrés Ortiz	University of Málaga, Spain
Ángel Arroyo	University of Burgos, Spain
Angelos Amanatiadis	Democritus University of Thrace, Greece
Anna Burduk	Wroclaw University of Technology, Poland
Antonio Bahamonde	University of Oviedo, Spain
António Dourado	University of Coimbra, Portugal
Arkadiusz Kowalski	Wroclaw University of Technology, Poland
Arturo de la Escalera	University Carlos III de Madrid, Spain
Arturo Hernández-Aguirre	CIMAT, Mexico
Barna Iantovics	PetruMaior University of Tg. Mures, Romania
Belén Vaquerizo	University of Burgos, Spain
Bernardete Ribeiro	University of Coimbra, Portugal
Bingyang Zhao	Tsinghua University, China
Blanca Cases Gutierrez	University of the Basque Country/EHU, Spain
Bogdan Trawinski	Wroclaw University of Technology, Poland
Borja Fernandez-Gauna	University of the Basque Country/EHU, Spain
Bożena Skolud	Silesian University of Technology, Poland
Bruno Baruque	University of Burgos, Spain
Camelia Chira	University of Babes-Bolyai, Romania
Camelia Pintea	North University of Baia-Mare, Romania
Carlos Carrascosa	Universidad Politécnica de Valencia, Spain
Carlos D. Barranco	Pablo de Olavide University, Spain
Carlos G. Puntonet	Universidad de Granada, Spain
Carlos Pereira	University of Coimbra, Portugal
Carmen Hernández	University of the Basque Country/EHU, Spain
Carmen Vidaurre	Berlin Institute of Technology, Germany
César Hervás	University of Córdoba, Spain
Cezary Grabowik	Silesian University of Technology, Poland
Constantin Zopounidis	Technical University of Crete, Greece
Cristóbal José Carmona	University of Jaén, Spain
Damian Krenczyk	Silesian University of Technology, Poland
Daniel Mateos-García	University of Seville, Spain
Dante I. Tapia	University of Salamanca, Spain
Dario Landa-Silva	University of Nottingham, UK
Darya Chyzhyk	University of the Basque Country/EHU, Spain
David Iclanzan	Sapientia Hungarian University of Transylvania, Romania
Diego Pablo Ruiz	University of Granada, Spain
Diego Salas-Gonzalez	University of Granada, Spain
Dimitris Mourtzis	University of Patras, Greece
Dominik Slezak	University of Regina, Canada

Donald Davendra	VSB TU Ostrava, Czech Republic
Dragan Simic	University of Novi Sad, Serbia
Dragos Horvath	Université de Strassbourg, France
Eiji Uchino	Yamaguchi University, Japan
Elías Fernández-Combarro	University of Oviedo, Spain
Emilio Corchado	University of Salamanca, Spain
Estefania Argente	University of Valencia, Spain
Eva Lucrecia Gibaja	University of Córdoba, Spain
Fabricio Olivetti de França	University of Campinas, Brazil
Federico Divina	Pablo de Olavide University, Spain
Feifei Song	Peking University, China
Fermín Segovia	University of Granada, Spain
Fernando De La Prieta	University of Salamanca, Spain
Fidel Aznar	University of Alicante, Spain
Florentino Fdez-Riverola	University of Vigo, Spain
Francisco Bellas	University of Coruña, Spain
Francisco Cuevas	CIO, Mexico
Francisco Fernández-Navarro	University of Córdoba, Spain
Francisco Herrera	University of Granada, Spain
Francisco Martínez	University of Córdoba, Spain
Francisco Martínez-Álvarez	Pablo de Olavide University, Spain
Frank Klawonn	Ostfalia University of Applied Sciences, Germany
George Dounias	University of the Aegean, Greece
George Papakostas	Democritus University of Thrace, Greece
Gerardo M. Méndez	Instituto Tecnológico de Nuevo León, Mexico
Gerhard Ritter	University of Florida, USA
Giancarlo Mauri	University of Milan-Bicocca, Italy
Giorgio Fumera	University of Cagliari, Italy
Gloria Cerasela Crisan	Vasile Alecsandri University of Bacau, Romania
Gonzalo A. Aranda-Corral	University of Huelva, Spain
Guiomar Corral	Ramon Llull University, Spain
Guoyin Wang	Chongqing University of Posts and Telecommunications, China
Han Pingchou	Peking University, China
Henrietta Toman	University of Debrecen, Hungary
Honghai Liu	University of Portsmouth, UK
Huiyu Huiyu Zhou	Queen's University Belfast, UK
Ignacio Turias	University of Cadiz, Spain
Indre Zliobaite	Bournemouth University, UK
Inés Galván	University Carlos III de Madrid, Spain
Ingo Keck	University of Regensburg, Germany
Ioannis Hatzilygeroudis	University of Patras, Greece
Irene Díaz	University of Oviedo, Spain
Isabel Barbancho	University of Málaga, Spain
Isabel Nepomuceno	University of Seville, Spain

Ivan Zelinka	Tomas Bata University, Czech Republic
IvicaVeza	University of Split, Croatia
Jacino Mata	University of Huelva, Spain
Jaume Bacardit	University of Nottingham, UK
Javier Bajo	Universidad Pontificia de Salamanca, Spain
Javier de Lope	Universidad Politécnica de Madrid, Spain
Javier R. Pérez	Universidad de Granada, Spain
Javier Sedano	University of Burgos, Spain
Jerzy Grzymala-Busse	University of Kansas, USA
Jerzy Sas	Wroclaw University of Technology, Poland
Jerzy Stefanowski	Poznan University of Technology, Poland
Jesús Alcala-Fernandez	University of Granada, Spain
Joaquín Derrac	University of Granada, Spain
Jorge Díez	University of Oviedo, Spain
Jorge García	University of Seville, Spain
José Dorronsoro	Universidad Autónoma de Madrid, Spain
José García	University of Alicante, Spain
José L. Álvarez	Universidad de Huelva, Spain
Jose Luis Calvo	University of Coruña, Spain
José Luis Martínez	Universidad de Castilla-La Mancha, Spain
José Luis Verdegay	University of Granada, Spain
José M. Armingol	University Carlos III de Madrid, Spain
José M. Molina	University of Seville, Spain
José Manuel López	University of the Basque Country/EHU, Spain
José R. Villar	University of Oviedo, Spain
José Ramón Cano	University of Jaén, Spain
Jose Ranilla	University of Oviedo, Spain
José Riquelme	University of Seville, Spain
Jovita Nenortaite	Kaunas University of Technology, Lithuania
Juan Álvaro Muñoz	University of Almería, Spain
Juan F. De Paz Santana	University of Salamanca, Spain
Juan Humberto Sossa	CIC-IPN, Mexico
Juan José Flores	University of Michoacana, Mexico
Juan M. Corchado	University of Salamanca, Spain
Juan Manuel Gorriz	University of Granada, Spain
Juan Pavón	Universidad Complutense de Madrid, Spain
Julián Luengo	University of Granada, Spain
Julio César Ponce	Universidad Autónoma de Aguascalientes, Mexico
Kamil Krot	Wroclaw University of Technology, Poland
Karmele López de Ipina	University of the Basque Country/EHU, Spain
Katya Rodríguez-Vázquez	Universidad Nacional Autónoma de México, Mexico
Keshav Dahal	University of Bradford, UK
Kevin Knuth	University at Albany, USA
Khaled Ragab	King Faisal University, Saudi Arabia

Konrad Jackowski	Wroclaw University of Technology, Poland
Krzysztof Kalinowski	Silesian University of Technology, Poland
Lars Graening	Honda Research Institute Europe, Germany
Lauro Snidaro	University of Udine, Italy
Lenka Lhotská	Czech Technical University in Prague, Czech Republic
Leocadio González	University of Almería, Spain
Leticia Curiel	University of Burgos, Spain
Li Cheng	University of North Carolina, USA
Lina Petrakieva	Glasgow Caledonian University, UK
Lourdes Sáiz	University of Burgos, Spain
Luis Alonso	University of Salamanca, Spain
Luis Búrdalo	Universitat Politècnica de València, Spain
Maciej Grzenda	Warsaw University of Technology, Poland
Maite García-Sebastián	Fundación CITA-Alzheimer, Spain
Marcilio de Souto	Universidade Federal do Rio Grande do Norte, Brazil
Marcin Zmyślony	Wroclaw University of Technology, Poland
Marco Mora	Universidad Católica del Maule, Chile
María del Mar Martínez	University of Seville, Spain
María Dolores Torres	Universidad Autónoma de Aguascalientes, Mexico
María Guijarro	Universidad Complutense de Madrid, Spain
María José del Jesús	University of Jaén, Spain
María Sierra	University of Oviedo, Spain
Mario Köppen	Kyushu Institute of Technology, Japan
Marta Arias	Universidad Politécnica de Cataluña, Spain
Martí Navarro	Universidad Politécnica de Valencia, Spain
Matjaz Gams	Jozef Stefan Institute Ljubljana, Slovenia
Michał Kuliberda	Wroclaw University of Technology, Poland
Mieczysław Jagodziński	Silesian University of Technology, Poland
Miguel A. Patricio	University Carlos III de Madrid, Spain
Miguel Ángel Veganzones	University of the Basque Country/EHU, Spain
Mohammed Chadli	UPJV, France
Neveen Ghali	Al-Azhar University, Egypt
Nicola Di Mauro	University of Bari Aldo Moro, Italy
Nikos Thomaidis	University of the Aegean, Greece
Nima Hatami	University of Cagliari, Italy
Norberto Díaz	Pablo de Olavide University, Spain
Óscar Ibañez	European Centre for Soft Computing, Spain
Otoniel López	Miguel Hernandez University, Spain
Ozgur Koray Sahingoz	Turkish Air Force Academy, Turkey
Pablo González	University of the Basque Country/EHU, Spain
Paola Mello	University of Bologna, Italy
Paula Castro	University of Coruña, Spain
Pedro Antonio Gutiérrez	University of Córdoba, Spain

Peter Rockett	The University of Sheffield, UK
Peter Sussner	University of Campinas, Brazil
Petrica Pop	North University of Baia-Mare, Romania
Petro Gopych	Universal Power Systems USA, Ukraine
Przemysław Kazienko	Wroclaw University of Technology, Poland
Rafael Alcalá	University of Granada, Spain
Rafael Corchuelo	University of Seville, Spain
Ramón Moreno	University of the Basque Country/EHU, Spain
Ramón Rizo	University of Alicante, Spain
Ricardo del Olmo	University of Burgos, Spain
Richard Duro	University of Coruña, Spain
Richard Freeman	Capgemini, Spain
Robert Burduk	Wroclaw University of Technology, Poland
Roberto Uribeetxeberria	Mondragon University, Spain
Rodica I. Lung	University of Babes-Bolyai, Romania
Rodolfo Zunino	University of Genoa, Italy
Roman Senkerik	Tomas Bata University in Zlin, Czech Republic
Ronald Yager	Iona College, USA
Roque Marin	University of Murcia, Spain
Rubén Fuentes-Fernández	Universidad Complutense de Madrid, Spain
Salvador García	University of Jaén, Spain
Sean Holden	University of Cambridge, UK
Sebastián Ventura	University of Córdoba, Spain
Shanmugasundaram Hariharan	Anna University, India
Soo-Young Lee	Brain Science Research Center, Korea
Stella Heras	Universidad Politécnica de Valencia, Spain
Talbi El-Ghazali	University of Lille, France
Teresa Ludermir	Federal University of Pernambuco, Brazil
Theodore Pachidis	Technological Educational Institution of Kavala, Greece
Tom Heskes	Radboud University Nijmegen, The Netherlands
Tomasz Kajdanowicz	Wroclaw University of Technology, Poland
Ulf Johansson	University of Borås, Sweden
Urko Zurutuza	Mondragon University, Spain
Urszula Markowska-Kaczmar	Wroclaw University of Technology, Poland
Urszula Stanczyk	Silesian University of Technology, Poland
Vasile Palade	Oxford University, USA
Vassilis Kaburlasos	Technological Educational Institution of Kavala, Greece
Vicente Julián	Universidad Politécnica de Valencia, Spain
Waldemar Malopolski	Cracow University of Technology, Poland
Wei-Chiang Samuelson Hong	Oriental Institute of Technology, Taiwan
Wei Yang Dai	Fudan University, China
Wieslaw Chmielnicki	Jagiellonian University, Poland
Yannis Marinakis	Technical University of Crete, Greece

Ying Tan	Peking University, China
Yusuke Nojima	Osaka Prefecture University, Japan
Zuzana Oplatkova	Tomas Bata University in Zlin, Czech Republic

Special Sessions

Systems, Man, and Cybernetics by HAIS

Emilio Corchado	University of Salamanca, Spain
Manuel Graña	University of the Basque Country/EHU, Spain
Richard Duro	University of Coruña, Spain
Juan M. Corchado	University of Salamanca, Spain
Vicent Botti	Polytechnical University of Valencia, Spain
Ramón Rizo	University of Alicante, Spain
Juan Pavón	University Complutense of Madrid, Spain
José Manuel Molina	University Carlos III of Madrid, Spain
Francisco Herrera	University of Granada, Spain
César Hervás	University of Córdoba, Spain
Sebastian Ventura	University of Córdoba, Spain
Álvaro Herrero	University of Burgos, Spain
Bruno Baruque	University of Burgos, Spain
Javier Sedano	University of Burgos, Spain
Sara Rodríguez	University of Salamanca, Spain
Lourdes Sáiz Barcena	University of Burgos, Spain
Ana Gil	University of Salamanca, Spain
Héctor Quintián	University of Salamanca, Spain
José Luis Calvo Rolle	University of Coruña, Spain
María Dolores Muñoz	University of Salamanca, Spain
Ángel Arroyo	University of Burgos, Spain

Methods of Classifier Fusion

Emilio Corchado	University of Salamanca, Spain
Bruno Baruque	University of Burgos, Spain
Michał Woźniak	Wroclaw University of Technology, Poland
Václav Snášel	VSB-Technical University of Ostrava, Czech Republic
Bogdan Trawinski	Wroclaw University of Technology, Poland
Giorgio Fumera	University of Cagliari, Italy
Konrad Jackowski	Wroclaw University of Technology, Poland
Konstantinos Sirlantzis	University of Kent, UK
Robert Burduk	Wroclaw University of Technology, Poland
Urszula Stanczyk	Silesian University of Technology, Poland
Przemysław Kazienko	Wroclaw University of Technology, Poland
Jerzy Stefanowski	Poznan University of Technology, Poland
Julián Luengo	University of Burgos, Spain

Balint Antal	University of Debrecen, Hungary
Hadju Andras	University of Debrecen, Hungary
Tom Heskes	Radboud University Nijmegen, The Netherlands
Leticia Curiel	University of Burgos, Spain

HAIS for Computer Security (HAISFCS)

Emilio Corchado	University of Salamanca, Spain
Álvaro Herrero	University of Burgos, Spain
Ángel Arroyo Puente	University of Burgos, Spain
Carlos Laorden	University of Deusto, Spain
Ignacio Arenaza	Mondragon University, Spain
Igor Santos Grueiro	University of Deusto, Spain
Manuel Jacinto Martínez	Ibermática, Spain
Valentina Casola	Università degli Studi di Napoli Federico II, Italy
Juan Álvaro Muñoz Naranjo	University of Almería, Spain
Amparo Fúster-Sabater	Institute of Applied Physics, Spain
Petro Gopych	Universal Power Systems USA, Ukraine
Raquel Redondo	University of Burgos, Spain
Urko Zurutuza	Mondragon University, Spain
Xiuzhen Chen	Shanghai Jiaotong University, China
Wenjian Luo	University of Science and Technology of China, China
Héctor Alaiz Moretón	University of León, Spain
Juan Jesús Barbarán Sánchez	University of Granada, Spain
Luis Hernández Encinas	Consejo Superior de Investigaciones Científicas, CSIC, Spain
Juan Tapiador	University of York, UK
Belén Vaquerizo	University of Burgos, Spain
Bernardete Ribeiro	University of Coimbra, Portugal
Joaquín García-Alfaro	Carleton University, Canada
Juan Manuel González Nieto	Queensland University of Technology, Australia
Ricardo Contreras Arriagada	Universidad de Concepción, Chile
Wei Wang	Norwegian University of Science and Technology, Norway
Paul Axayacatl Frausto	Mediscs, France
SeemaVerma	Banasthali University, India

Data Mining: Data Preparation and Analysis

Salvador García	University of Jaén, Spain
Julián Luengo	University of Burgos, Spain
Francisco Herrera	University of Granada, Spain

Antonio Rivera	University of Jaén, Spain
Cristóbal J. Carmona	University of Jaén, Spain
Isaac Triguero	University of Granada, Spain
José A. Sáez	University of Granada, Spain
Mikel Galar	Public University of Navarra, Spain
Victoria López	University of Granada, Spain
Alberto Fernández	University of Granada, Spain
Jose Antonio Sanz	Public University of Navarra, Spain
Ana M. Martínez	Universidad de Castilla-La Mancha, Spain
Habiba Drias	USTHB, Algeria
Jesús Alcalá-Fdez	University of Granada, Spain
Joaquín Derrac Rus	University of Granada, Spain
Jose R. Villar	University of Oviedo, Spain
Sergio Esparcia	Universidad Politécnica de Valencia, Spain
Stefanos Ougiaroglou	University of Macedonia, Greece
José García Moreno	University of Granada, Spain
Nenad Tomasev	Jozef Stefan Institute, Stovenia
Rafael del Hoyo	Technological Institute of Aragón, Spain
Krystyna Napierala	Poznan University of Technology, Poland
Jose Ramón Cano	University of Jaén, Spain
Aida Gema de Haro	University of Córdoba, Spain
Ana Palacios	University of Oviedo, Spain
Antonio Jesus Rivera	University of Jaén, Spain
Kim Hee-Cheol	Inje University, Korea
Miguel García Torres	Pablo de Olavide University, Spain
Núria Macià	Universitat Ramon Llull, Spain
Rubén Jaramillo	LAPEM-CIATEC, Spain
Olgierd Unold	Wroclaw University of Technology, Poland
Pablo Bermejo	Universidad de Castilla-La Mancha, Spain
Philippe Fournier-Viger	University of Moncton, Canada
Yong Shi	Kennesaw State University, USA

Hybrid Artificial Intelligence Systems in Management of Production Systems

Edward Chlebus	Wroclaw University of Technology, Poland
Milan Gregor	University of Žilina, Slovak Republic
Ulrich Günther	Dresden University of Technology, Germany
Adam Hamrol	Poznan University of Technology, Poland
Bożena Skolud	Wroclaw University of Technology, Poland
Anna Burduk	Wroclaw University of Technology, Poland
Arkadiusz Kowalski	Wroclaw University of Technology, Poland
Cezary Grabowik	Wroclaw University of Technology, Poland
Kamil Krot	Wroclaw University of Technology, Poland
Krzysztof Kalinowski	Wroclaw University of Technology, Poland
Miczyslaw Jagodzinski	Wroclaw University of Technology, Poland

Tomasz Chlebus	Wroclaw University of Technology, Poland
Michał Kuliberda	Wroclaw University of Technology, Poland
Damian Krenczyk	Wroclaw University of Technology, Poland
DimitrisMourtzis	University of Patras, Greece

Hybrid Artificial Intelligent Systems for Ordinal Regression

César Hervás	University of Córdoba, Spain
Pedro Antonio Gutiérrez-Peña	University of Córdoba, Spain
Jaime S. Cardoso	University of Porto, Portugal
Francisco Fernández-Navarro	University of Córdoba, Spain
Francisco Martínez-Estudillo	University of Córdoba, Spain
Javier Sánchez-Monedero	University of Córdoba, Spain
Manuel Cruz-Ramírez	University of Córdoba, Spain
Ricardo Sousa	INESC, Portugal
Arie Ben David	University of Córdoba, Spain
David Becerra-Alonso	University of Córdoba, Spain

Hybrid Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

José RamónVillar	University of Oviedo, Spain
Camelia Chira	University of Babes-Bolyai, Romania
Enrique de la Cal	University of Oviedo, Spain
Anca Gog	University of Babes-Bolyai, Romania
Camelia Pintea	North University Baia-Mare, Romania
Gerardo Méndez	Instituto Tecnológico Nuevo León, Mexico
Javier Sedano	Instituto Tecnológico de Castilla y León, Spain
José Luis Calvo Rolle	University of Coruña, Spain
Petrica Pop	North University Baia-Mare, Romania
Adolfo Rodríguez	University of León, Spain
María Sierra	University of Oviedo, Spain
Óscar Ibañez	European Centre of Soft Computing, Spain
André Carvalho	University of São Paulo, Brazil
Luciano Sánchez	University of Oviedo, Spain
Paola Mello	University of Bologna, Italy
Nima Hatami	University of Cagliari, Italy

Hybrid Computational Intelligence and Lattice Computing for Image and Signal Processing

Manuel Graña	University of the Basque Country/EHU, Spain
Alexandre Savio	University of the Basque Country/EHU, Spain
Borja Fernandez-Gauna	University of the Basque Country/EHU, Spain

Darya Chyzhyk	University of the Basque Country/EHU, Spain
Ekaitz Zulueta	University of the Basque Country/EHU, Spain
Ion Marques	University of the Basque Country/EHU, Spain
Josu Maiora	University of the Basque Country/EHU, Spain
Miguel Ángel Veganzones	University of the Basque Country/EHU, Spain
Ana I Gonzalez	University of the Basque Country/EHU, Spain
Dragan Simic	University of Novi Sad, Serbia
Iñigo Barandiaran	Vicomtech, Spain
Israel Rebollo Ruiz	University of the Basque Country/EHU, Spain
Maite Termenon	University of the Basque Country/EHU, Spain
Ivan Macia	Vicomtech, Spain
Borja Ayerdi	University of the Basque Country/EHU, Spain
Elsa Fernández	University of the Basque Country/EHU, Spain
Andoni Beristain	Vicomtech, Spain
Ramón Moreno	University of the Basque Country/EHU, Spain

Workshop Committees

Nonstationary Models of Pattern Recognition and Classifier Combinations

Michał Woźniak	Wroclaw University of Technology, Poland
Emilio Corchado	University of Salamanca, Spain
Boguslaw Cyganek	AGH University of Science and Technology, Poland
Francisco Herrera	University of Granada, Spain
Giorgio Fumera	University of Cagliari, Italy
Ioannis Katakis	University of Cyprus, Greece
Manuel Graña	University of the Basque Country/EHU, Spain
Robert Burduk	Wroclaw University of Technology, Poland
Jerzy Stefanowski	Poznan University of Technology, Poland
Przemysław Kazienko	Wroclaw University of Technology, Poland
Álvaro Herrero	University of Burgos, Spain
Bruno Baruque	University of Burgos, Spain
Piotr Sobolewski	Wroclaw University of Technology, Poland
Konrad Jackowski	Wroclaw University of Technology, Poland
Václav Snášel	VSB-Technical University of Ostrava, Poland
Piotr Cal	Wroclaw University of Technology, Poland
Marcin Zmyślony	Wroclaw University of Technology, Poland
Konstantinos Sirlantzis	University of Kent, UK

Organizing Committee

Emilio Corchado	University of Salamanca, Spain (Co-chair)
Bruno Baruque	University of Burgos, Spain (Co-chair)
Álvaro Herrero	University of Burgos, Spain (Co-chair)

José Luis Calvo	University of Coruña, Spain (Co-chair)
Leticia Curiel	University of Burgos, Spain
M^a Dolores Muñoz	University of Salamanca, Spain
Ángel Arroyo	University of Burgos, Spain
Javier Sedano	University of Burgos, Spain
Fernando De la Prieta	University of Salamanca, Spain
Ana Gil	University of Salamanca, Spain
M^a Araceli Sánchez	University of Salamanca, Spain
Héctor Quintián	University of Salamanca, Spain
Héctor Casado	University of Salamanca, Spain
Antonio J. Sánchez	University of Salamanca, Spain

Table of Contents – Part I

Special Sessions

Agents and Multi Agents Systems

HAIS Applications

Cluster Analysis

Data Mining and Knowledge Discovery

Evolutionary Computation

Learning Algorithms

Systems, Man, and Cybernetics by HAIS

Table of Contents – Part II

Special Sessions

Methods of Classifier Fusion

HAIS for Computer Security (HAISFCS)

Data Mining: Data Preparation and Analysis

Hybrid Artificial Intelligence Systems in Management of Production Systems

Hybrid Artificial Intelligent Systems for Ordinal Regression

Hybrid Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

Hybrid Computational Intelligence and Lattice Computing for Image and Signal Processing

Workshop

Nonstationary Models of Pattern Recognition and Classifier Combinations

An Agent Model for Incremental Rough Set-Based Rule Induction in Customer Relationship Management

Yu-Neng Fan* and Ching-Chin Chern

Department of Information Management, National Taiwan University, Taiwan

d96725003@ntu.edu.tw, cchern@im.ntu.edu.tw

Abstract. Compared to other methods, rough set (RS) has the advantage of combining both qualitative and quantitative information in decision analysis, which is extremely important for customer relationship management (CRM).

In this paper, we introduce an application of a multi-agent embedded incremental rough set-based rule induction to CRM, namely Incremental Rough Set-based Rule Induction Agent (IRSRIA). The rule induction is based on creating agents within the main modeling processes. This method is suitable for qualitative information and also takes into account user preferences. Furthermore, we designed an incremental architecture for addressing dynamic database problems of rough set-based rule induction, making it unnecessary to re-compute the whole dataset when the database is updated. As a result, huge degrees of computation time and memory space are saved when executing IRSRIA. Finally, we apply our method to a case study of a cell phone purchase. The results show the practical viability and efficiency of this method, and thus this paper forms the basis for solving many other similar problems that occur in the service industry.

Keywords: Rough sets, Incremental Algorithm, Rule Induction, Intelligent Agents, Customer Relationship Management.

1 Introduction

Today, a numerous and diverse number of enterprises are rushing to become more customer focused. A key component of many such initiatives is the implementation of Customer Relationship Management (CRM). CRM, defined as a widely implemented strategy for managing a company's interactions with customers, uses technology to organize, automate, and synchronize business processes. The purpose of CRM is to increase business effectiveness and efficiency, and thereby increase profitability [1].

Accordingly, many data mining approaches have been devised to analyze the CRM data, such as association rules, classification, clustering, and prediction, and so forth. However, in CRM, information related to customer-preferred features are sometimes collected by using research, interviews, meetings, questionnaires, sampling, and other techniques. These types of data are often discrete and frequently in qualitative format.

* Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 1–12, 2012.

Analyzing this qualitative data and extracting information are useful for promoting sales is critical to CRM [2]. Rough Set Theory (RST), which is introduced by Pawlak in 1982 [3], is a knowledge discovery tool that can be used to help induce logical patterns hidden in massive data. Compared to other methods, the rough set approach has the advantage of combining both qualitative and quantitative information in the decision analysis, and it acts as a very usable tool to analyze data involving preference[4]. All these benefits are extremely important to CRM. In addition, rough set-based rule induction is a novel data mining technique that can automatically explore and analyze large amounts of data in transaction databases, and then can discover potentially significant patterns and rules underlying the database [5]. Since it develops synthetic and easily understandable representations of knowledge by using if-then format decision rules, it is useful in decision support and strategy formulation of CRM.

Besides issues with managing qualitative data, another challenge of CRM is the quickly growing rate of data in dynamic databases. Once a system is installed and used for daily operations within a business, tremendous amounts of data is updated. As the amount of data changes over time, using this information without refreshing could lead to inaccurate analytical results and thus unacceptable decisions, because the extracted knowledge is based on previous behaviors of the analyzed objects [6]. Because the issue occurs frequently, an efficient incremental rough set-based rule induction to CRM is required. In addition, when discovering the preference-based decision rule by rough set theory in CRM, this process is complex and computationally intensive, and involve many optimization problems, especially in reduct computation of rules generation phase, it is NP-hard [7]. These problems result in high process times and the generation of large numbers of rules. Therefore, we solve these problems by embedding an agent-based approach within the incremental rough set-based rule induction framework.

In summary, considering the aforementioned issues, we embedded an agent-based model within the incremental rough set-based rule induction, named Incremental Rough Set-based Rule Induction Agent (IRSRIA). The purpose of this paper is to illustrate that the use of these agents within rule induction can improve mining speed and maintain the quality of knowledge. In addition, this method is suitable to deal with qualitative information and takes the user's preferences into account. Moreover, IRSRIA can deal with incremental data solely instead of re-computing the entire dataset when the database is updated. As a result, huge computing time and memory space are saved. A case study of CRM is applied to show the validity and efficiency of this method. Since this subject is rarely considered in previous literature, we believe that this study will open a new avenue for CRM.

2 Literature Review

2.1 Rough Set-Based Rule Induction

In order to gain meaningful decision rules, two approaches are introduced. The first one is feature reduction algorithm, which is a pre-processing method of rule induction. It removes redundant information or features and selects a feature subset that has the same discernibility as the original set of features. Secondly, the rule induction

algorithms generate decision rules that potentially reveal profound knowledge and provide new insight. These decision rules are more useful for experts to analyze and gain understanding of the problem at hand [8].

However, some conventional rough set-based approaches cannot produce rules containing preference order; namely, they cannot achieve more meaningful and general rules [9]. Also, the knowledge discovery literature [10] indicates that using rough set-based rule induction often generates too many rules without focus. These rough set approaches cannot guarantee that the classification of a decision table is credible [11]. Besides, the conventional rough set-based rule inductions assume that all features are equally important. This assumption is not likely to prevail, as the weighting of each feature may not necessarily be equal to the real-world case [12]. For example, some features might be more important than others and therefore may be assigned a higher weight. In order to resolve these problems, Tseng, Huang, and Jiang [11] proposed the Rule-Extraction Algorithm (REA). REA was presented for discovering preference-based rules according to the reducts that contain the maximum strength index (SI). Also, REA takes into consideration features that are of unequal significance, and their respective weighting.

In conclusion, this study develops the IRSRIA based on the concept of Rule-Extraction Algorithm (REA) of Tseng, Huang, and Jiang [11]. It is a reliable rule induction approach that meets the requirements of CRM and is capable of managing qualitative data. Moreover, the client's preferences with respect to product features are taken into account. The resulting decision rules are concise and understandable, and thus they are useful in decision support and strategy formulation of CRM.

2.2 Related Incremental Algorithm

In data mining, an incremental technique is a way to solve the issue of newly-added data without re-implementing the DM algorithm in a dynamic database. Various methods have been developed to deal with the problem of dynamic databases in the fields of classification, association rules, clustering methods, and so on [13]. For example, Agrawal and Bala [14] built an incremental Bayesian classification for multivariate normal distribution data. Awad and Motai [15] proposed an incremental multi-classification method and applied to video stream data, which consists of an articulated humanoid model monitored by a surveillance camera.

Despite the wealth of prior researches, some of these approaches are not suitable for processing qualitative information. In addition, some of the aforementioned approaches are a type of population-based approach that may require several statistical assumptions, and thus have limitations in handling qualitative types of data. Furthermore, a number of them require calculating the decision matrix for each decision attribute, even though the number of decision attributes are usually very large (e.g. in mass data information systems). This leads to the consumption of large amounts of time and memory [16].

2.3 Hybrid Artificial Intelligent Agent Systems

In this study, the proposed Incremental Rough Set-based Rule Induction Agent (IRSRIA) is based on RS-based rule induction and incremental algorithm. The RS-based rule induction aims at generating the meaningful decision rules for decision

makers to analyze CRM database. The incremental algorithm is a way to solve the issue of newly-added data without re-implementing the overall algorithm in a dynamic database. Since both of the processes in RS-based rule induction and incremental algorithm are computationally intensive, we embedded an agent-based method into this incremental rule induction framework. Ideally, such hybrid architecture would be beneficial for decision support by incorporating the advantages of rough set theory, incremental approach and multi-agent system. Practical experience has also indicated that hybrid intelligence techniques might be helpful to solve some of the challenging real world problems [17]. In a hybrid intelligence system, a synergistic combination of multiple techniques is used to build an efficient solution to deal with a particular problem [18]. For example, Pedrycz and Aliev [19] proposed a logic-oriented neural networks, which benefits from the establishment of highly synergistic links between the technology of fuzzy sets and neural networks. Bakar, Othman, and Hamdan [7] proposed a classifier based on creating agents within the main modeling processes such as reduct computation, rules generation and attribute projections, assisting the execution of a rough set classifier.

The agent is a program that performs a specific task intelligently without any human supervision and can communicate with other agents cooperatively [20]. The agent can assist or act on behalf of users to do multiple tasks or repetitive tasks. This hybrid intelligent agent system is a natural distributed computing environment, allowing agents running on different computing hosts simultaneously to achieve high availability [21]. As a result, it is suitable for addressing the fast-changing database of CRM. However, few agent-based approaches are applied on rough set theory. Therefore, the proposed IRSRIA is a prototype method and forms the basis for solving the rough set-base rule induction.

3 Solution Approach

3.1 The Incremental Architecture of Rough Set-Based Rule Induction

By employing Rough Set Theory, decision rules can be extracted for each homogeneous cluster of data records and the relationships between different clusters. However, because the data in most databases are ever changing, a good method for real applications must handle data change conveniently [22]. Since our approach is a typically incremental one, it can handle data change conveniently. Herein, the situation of data change can be divided into four cases:

- Case1: Data change causes the original rules and reducts to be incomplete and insufficient.
- Case2: Data change causes contradictions among the original rules and reducts.
- Case3: Data change causes the original rules insufficient and incomplete but not the original reducts.
- Case4: Data change does not cause any contradictions, and the original rules can cover new objects.

Case 1 indicates that when a new data set is added, the original rules and reducts can not cover the new data, that is, the new added-in data neither be covered nor contradict the original reducts and rules. Case 2 denotes that the added data set causes con-

tradictions among the original rules and reducts. For instance, object A collides with object B, if and only if object A and object B have the same condition feature values, and their corresponding decision feature values are different. Case 3 signifies that the newly added data set does not collide with the original rules and can also not be covered by the original rules. However, it can be covered by the original reducts. Lastly, Case 4 illustrates newly added data that does not collide with the original rules and that can be covered by the original rules. The case 3 and 4 are both the situation that the original database is able to deal with the new added in data and the whole database re-computing is unnecessary. The aforementioned four cases are used to exhibit situations when a new object is added into the database. In the following section, an agent-based approach IRSRIA is developed and employed to deal with the incremental rough set-based rule induction.

3.2 The Agent-Based Solution Approach

In this study, the rule induction is based on creating agents within the main modeling processes such as status determination, reduct generation, significance index (SI) computation, and rule induction. Four main agents are introduced: the status determination agent, the reduct generation agent, the SI computation agent, and the rule induction agent. First, the status determination agent is used to deal with the incremental issue. When a new data set is added into the CRM database, this agent will determine which case this new object belongs to and is responsible for identifying which objects in the original database are also affected. Then the remaining three agents are designed to handle the process of rough set-based rule induction in the sequence of the reduct generation agent, the SI computation agent, and the rule induction agent. These three are dedicated to inducting the preference-based decision rules. The details of these four agents will be discussed in the next sections.

3.2.1 The Status Determination Agent

The status determination agent acts as a program for determining in which case the new data object belongs. Also, this agent is responsible for identifying which objects in the original database are affected. The procedures of the status determination agent are illustrated in Figure 1.

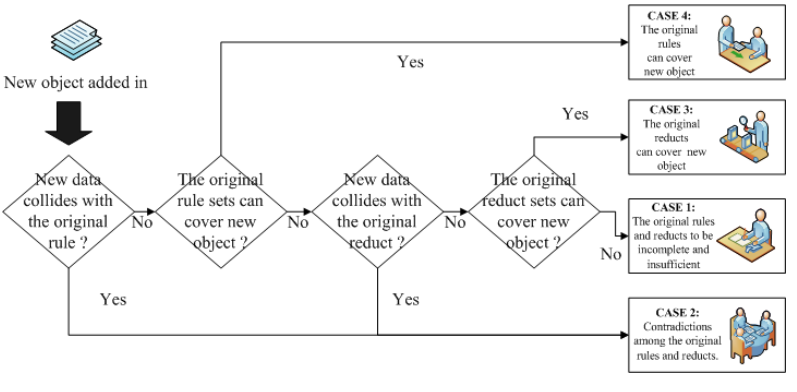


Fig. 1. The procedure of the status determination agent

The incremental techniques applied to rough sets have two bases: a reduct table and a rule table. When comparing the incremental algorithm based on the reduct table and the incremental algorithm based on the rule table, the second approach is more efficient [16]. The rule table, which contains information on the raw data, is deduced from the reduct table. Hence, there are fewer items in the rule table than in the reduct table, which is why the agent checks the rule table first.

As mentioned previously, the status determination agent is also responsible for identifying which objects in the original database are affected, thus the remaining three agents will execute tasks only on the objects that are affected by the newly added data. It does not re-compute the reducts or rules that are not influenced by the incremental data set; hence IRSRIA is less time-consuming than other traditional methods.

3.2.2 The Reduct Generation Agent

The reduct generation agent acts as a program of reduct computation. In Rough set theory, a reduct is generally defined as a minimal subset of features that can classify the same domain of objects as unambiguously as the original set of features [23]. Basically, the reducts represent necessary condition features to make a decision. By using feature reduction techniques, the decision rules extracted from an information system can be simplified without reducing the accuracy of the classification. The key notions in rough set-based feature reduction are reduct and core. However, the time complexity of the original algorithm for feature reduction proposed by Pawlak [23] is $O(m^2 2^n)$, where n is the number of features and m is the number of objects. Thus the algorithm is very limited in actual applications. For this concern, we adopt a complete algorithm for feature reduction in rough set theory proposed by Jiang and Du [24]. This algorithm not only finds the Pawlak reduction, but also guarantees that the reduction set found is the smallest available. Also, it has a polynomial complexity, which in the worst case is $O(m^2 n^2)$. Since finding all reducts is definitely a repetitive and painstaking task, an autonomous reduct generation agent-based on the incremental architecture is proposed. The procedures of the reduct generation agent are listed as follows.

The Procedure of the Reduct Generation Agent

```
Input:   The influenced objects
Output: The reducts of influenced objects
Reduct generation agent start
If the new object belongs to case 1 or 2
Then go to Step 1
Step 1. Initialization: List all objects
Step 2. Generate the reducts for each object
     If [Vij]Aj ⊂ [Vik]Oki
     Then the reducts for Xi is formed
     Else
If C−Ai ∩ [Vij]Ai ⊂ [Vik]Oki
Then the reducts for Xi is formed
Else the reducts for Xi is not formed
Else (case 3 or 4): The new object will be assigned the original reduct,
which can cover the newly added object
Reduct generation agent closed: Stop and output the results
```

Where each *Aj* feature contains *Vij*, while $[V_{ik}]_{O_{kli}}$ includes the objects with each *Ok*|*i* outcome (decision) feature containing *Vik*. In order to find dispensable features, the examination of each feature of the object is required.

3.2.3 The SI Computation Agent

The SI (significance index) computing agent computes the SI, which represents the importance of reduct in the decision table. The direct use of the results provided by the reduct generation agent may lead to many reducts that contain condition features that are not meaningful. SI facilitates extracting meaningful rules, because it is derived from the weight of each feature. The measurement of the weight is based on domain experts' judgment and external assessment in the specific industry [12]. In order to solve large-scale reduct selection problems, provide flexible weight assignment, and reduce the burden of developing sophisticated mathematical models, an SI computation agent is employed and presented. Through SI computation, the reduct with highly correlated features to customers' characteristics are identified.

An SI helps identify meaningful reducts. A reduct with a higher value in the significance index is more preferable to a reduct with a lower value index, because the higher the SI, the more desirable the features contained in the reduct. Moreover, the comparison of reducts is restricted to the same decision features and the number of features selected in the reducts. Finally, for obtaining the rules from the reducts, the SI computing agent needs to compute all reducts of each object, and then the rule induction agent is able to select the most preferred reduct (with maximum SI) as the final rule.

The significance index (SI) of reduct r is defined as follows.

$$\mathrm{SI}(r) = \sum_{j=1}^{m} \mathrm{W}_j \mathrm{v}_j \times \mathrm{n}_r \tag{1}$$

Where, r = the reduct number, $r = 1, ..., n$; n_r = the number of identical reducts r; Fj = the j-th feature, $j = 1, \ldots, m$; Wj = weight of the j-th feature; vj = 1 if condition feature j is selected, 0 otherwise (Fj = "x"). For example, if reduct 1 is {1, x, 3, (0)}, the first three element "1, x, 3" means the corresponding features 1 to 3 of this reduct. The last element (0) means that the outcome of reduct 1 is "0." When an expert assigns the weight of feature 1 to 3 is {0.1, 0.2, 0.3}. The SI of reduct 1 will be $0.1\times1 + 0.2\times0 + 0.3\times3 = 1$.

3.2.4 The Rule Induction Agent

The rule induction agent acts as a program for extracting the preferences-based decision rules from the reduct table. In order to acquire meaningful decision rules, two approaches are required. The first is a reduct generation procedure, and the second is a rule induction algorithm. The reduct generation procedure is pre-processing method of rule induction, which removes redundant information or features and selects a feature subset that has the same discernability as the original set of features. However, there are too many reducts generated from the procedure, and hence the rule induction algorithm is required to induct the concise decision rules that contain the maximum significance index. These decision rules may potentially reveal profound knowledge and provide new insight, which makes them more useful for experts to analyze and gain understanding into the current problem [25]. The following procedure of induct-

ing rules is performed by the rule induction agent. The original algorithm is adopted from the Rule-Extraction Algorithm (REA) of Tseng, Huang, and Jiang [11].

3.3 The Time Complexity of the Proposed IRSRIA

In order to present the time complexity of the proposed IRSRIA and the traditional REA [11], the following notations are used: n: the number of features, m: the number of objects, t: the number of reducts. The time complexity of the traditional REA algorithm in the worst case is $O(m^2 2^n + t^2 n)$, which can be solved in exponential time. However, the time complexity of the proposed IRSRIA in the worst case is $O(m^2 n^2 + t^2)$, which can be solved in polynomial time. Hence, in comparing the complexity of IRSRIA with REA, the proposed IRSRIA is more efficient.

4 Case Study

A cell phone vendor company HJK Inc. plans to develop a strategic plan to promote their products. Because the marketing manager desires to learn its customers' purchasing patterns through customer profiles, HJK Inc. collected data on consumers' preferred features that are related to cell phone purchasing decisions. The customer preferred features of the cell phone purchase are illustrated in Table 1.

Table 1. Customer preferred features of the cell phone purchase

	Description	0	1	2	3	4
F1	Amount of application	Matterless	Few	Medium	Large	—
F2	Operating system	Symbian	Android	iOS	Other	—
F3	Physical styles	Bar	Clamshell	Flip	Slide	Swivel
F4	Multimedia options & memory	Matterless	Low	Medium	High	—
F5	Battery life	Matterless	Low	Medium	High	—
F6	Personalization options	Matterless	Low	Medium	High	—
O	Outcome	Phone N	Phone H	Phone A	Phone B	—

To achieve this goal, HJK Inc. has employed a rough set base rule induction approach to extract the meaningful decision rules to understand the customer's preference upon the feature of the products (as Table 2). HJK Inc. has realized that this method is helpful to launch a desirable and effective marketing strategy and advertisement. Based on the derived decision rules, it is beneficial for HJK Inc. to offer the right products or services to the right segments of customers, and then recommend the appropriate product or service to the right person.

Table 2. The resulting concise rules of HJK Inc.

Concise Decision Rules
D1: F6="3" → Outcome="0"
D2: F3="3" → Outcome="1"
D3: F5="0" → Outcome="2"
D4: F3="0" → Outcome="3"
D5: F1="3" ∩ F2="2" ∩ F3="1" ∩ F5="2" ∩ F6="1" → Outcome="3"
D6: F2="2" ∩ F3="1" ∩ F4="1" ∩ F5="2" ∩ F6="1" → Outcome="3"
D7: F1="0" ∩ F2="3" ∩ F3="2" ∩ F5="3" ∩ F6="1" → Outcome="3"
D8: F2="3" ∩ F3="2" ∩ F4="1" ∩ F5="3" ∩ F6="1" → Outcome="3"

However, there are still challenges ahead. The data in the CRM database of HJK Inc. has been increasing, and will shortly be updated with a large amount of new information as a result of daily use and operations. In order to verify the efficiency of the proposed IRSRIA, an application has been developed for executing the Rule-Extraction Algorithm (REA) [11] and the proposed IRSRIA. The run time is obtained under the following environment: a Lenovo X200 7458RW8 notebook with 3 GB RAM, CPU (Intel® Core™ 2 Duo 2.40GHz), and Windows XP Service Pack 3. Also, a toolkit for rule induction, RESE (Rough Set Exploration System [26]) was used in the experiment. We measure the performance of our method by run time and classification accuracy.

4.1 Experiment 1

For experiment 1, the original dataset has 168 customers' data. Four incremental data sets in the CRM database of HJK Inc., belonging to cases 1 through 4 (in the incremental architecture of rough set-based rule induction), are tested. Table 3 shows the results of the comparison between traditional REA and IRSRIA applied on four different cases.

Table 3. The comparison between REA and IRSRIA applied on four different cases

	REA Run Time	IRSRIA Run time	The reductive percentage
Data set I (Case 1)	1.925	0.475	75.32%
Data set II (Case 2)	1.825	0.625	65.75%
Data set III (Case 3)	1.675	0.2	88.06%
Data set IV (Case 4)	1.675	0.15	91.04%
Overall	7.1	1.45	79.58%

The results of experiment 1 show significant improvement in the run time by applying IRSRIA on the four different cases. Especially on cases where the original rules or reduct can cover the incremental data set, the reductive percentage of the run time is higher than 88%. As for case 1 and 2, the reductive percentages of run time are both higher than 65%.

4.2 Experiment 2

For experiment 2, the original dataset has 168 customers' data. In order to test the validity of IRSRIA when applied to large-scale incremental data, we compared both IRSRIA and REA using 100 to 1000 newly added incremental objects. Table 4 shows the results of the comparison between traditional REA and IRSRIA.

Table 4 shows the ratios of the run times of REA to IRSRIA for different numbers of incremental objects. It reveals that IRSRIA performs 4.9 times better than REA when the number of incremental objects is 100, and 11.1 times better than REA when the number of incremental objects is 400. Most importantly, when the number of incremental objects is greater than 600, IRSRIA outperforms REA by more than 20 times.

Table 4. The run times for the different number of incremental objects

New Objects #	REA Run Time	IRSRIA Run Time	Ratio of REA to IRSRIA
100	246	50	4.9 : 1
200	595	86	6.9 : 1
400	1815	164	11.1 : 1
600	3555	176	20.2 : 1
800	7088	271	26.2 : 1
1000	10233	395	25.9 : 1

4.3 Experiment 3

For experiment 3, there are 168 original objects and 1393 incremental objects, a total of 1561 objects are used from the HJK's CRM database. In order to test the validity of the rules, which are extracted from the data set, we compared the classification accuracy of REA and IRSRIA with three traditional rule-based classification methods, including genetic algorithm (GA), LEM2 Algorithm (A new version of LERS), and Covering algorithm. In order to compare these different methods, the same test data was used. A toolkit for rule induction, RESE (Rough Set Exploration System) was also used in the experiment. The descriptions of these methods can refer to RESE tutorial [26]. The results of classification accuracy are provided in Table 5.

Table 5. Summary of classification accuracy results

Method	Classification Accuracy
REA and IRSRIA	89.6%
Genetic Algorithm (GA)	87.0%
LEM2 Algorithm	85.3%
Covering algorithm	88.2%

From the comparison between REA & IRSRIA and other rule-based classification methods with respect to the classification accuracy, these results show that REA & IRSRIA is the best performance in those rule-based methods. Our work shows that using the agent model can reduce the mining time while maintaining the accuracy of generated rules.

5 Conclusion

The proposed IRSRIA is an agent-based model within incremental rough set-based rule induction. It has been used to analyze user profiles for CRM. By employing IRSRIA, the meaningful decision rules behind customer purchasing patterns can be extracted. Since a client's preferences with respect to the features of a product have been taken into account, IRSRIA shows great promise for CRM, where businesses can precisely offer the right products or services to the right segments of customers.

As to the issue of dynamic databases, since IRSRIA is designed based on incremental architecture, it is capable of addressing dynamic database issues of CRM. IRSRIA aims at extracting rules by partially modifying the reducts, when new data is added to the database, rendering it unnecessary to re-compute the whole database

from the beginning. The case study in section 4 shows that the proposed method significantly reduces run time over all datasets and is able to maintain a high accuracy of generated rules simultaneously. Therefore, our work shows that using an agent model can considerably reduce the computation time of inducing the decision rules in a CRM database while maintaining the accuracy of rules.

References

1. Handen, L.: Putting CRM to work: The rise of the relationship. In: Brown, S.A. (ed.) Customer Relationship Management: A Strategic Imperative in the World of e-Business, pp. 7–18. Wiley, Toronto (2000)
2. Tseng, T.-L.B., Huang, C.-C., Ho, J.C.: Autonomous Decision Making in Customer Relationship Management: A Data Mining Approach. In: Proceeding of the Industrial Engineering Research 2008 Conference, Vancouver, British Columbia, Canada (2008)
3. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
4. Kusiak, A.: Feature transformation methods in data mining. IEEE Transactions on Electronics Packaging Manufacturing 24, 214–221 (2001)
5. Changchien, S.W., Lu, T.C.: Mining association rules procedure to support on-line recommendation by customers and products fragmentation. Expert Systems with Applications 20, 325–335 (2001)
6. Crespo, F., Weber, R.: A methodology for dynamic data mining based on fuzzy clustering. Fuzzy Sets and Systems 150, 267–284 (2005)
7. Bakar, A.A., Othman, Z.A., Hamdan, A.R., Yusof, R., Ismail, R.: An agent model for rough classifiers. Applied Soft Computing 11, 2239–2245 (2011)
8. Wang, X., Yang, J., Jensen, R., Liu, X.: Rough set feature selection and rule induction for prediction of malignancy degree in brain glioma. Computer Methods and Programs in Biomedicine 83, 147–156 (2006)
9. Chengmin, S., Dayou, L., Chunxiao, F., Shuyang, S.: Algorithm Studies of Rules Generation in CORS. In: 5th IEEE International Conference on Cognitive Informatics, ICCI 2006, pp. 572–577 (2006)
10. Pattaraintakorn, P., Cercone, N.: Integrating rough set theory and medical applications. Applied Mathematics Letters 21, 400–403 (2008)
11. Tseng, T.-L.B., Huang, C.-C., Jiang, F., Ho, J.C.: Applying a hybrid data mining approach to prediction problems: A case of preferred suppliers prediction. International Journal of Production Research 44, 2935–2954 (2006)
12. Billtseng, T., Huang, C.: Rough set-based approach to feature selection in customer relationship management. Omega 35, 365–383 (2007)
13. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishers (2006)
14. Agrawal, R.K., Bala, R.: Incremental Bayesian classification for multivariate normal distribution data. Pattern Recognition Letters 29, 1873–1876 (2008)
15. Awad, M., Motai, Y.: Dynamic classification for video stream using support vector machine. Applied Soft Computing 8, 1314–1325 (2008)
16. Liu, Y., Xu, C., Pan, Y.: An incremental rule extracting algorithm based on Pawlak reduction. In: 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 5966, pp. 5964–5968 (2004)

17. Corchado, E., Graña, M., Woźniak, M.: Editorial: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75, 61–63 (2012)
18. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. Inf. Sci. 180, 2633–2634 (2010)
19. Pedrycz, W., Aliev, R.A.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73, 10–23 (2009)
20. Liang, W.-Y., Huang, C.-C.: Agent-based demand forecast in multi-echelon supply chain. Decision Support Systems 42, 390–407 (2006)
21. Luo, P., Lü, K., Huang, R., He, Q., Shi, Z.: A heterogeneous computing system for data mining workflows in multi-agent environments. Expert Systems 23, 258–272 (2006)
22. Zhong, N., Dong, J.-Z., Ohsuga, S., Lin, T.Y.: An incremental, probabilistic rough set approach to rule discovery. In: IEEE International Conference on Fuzzy Systems, Anchorage, AK, USA, pp. 933–938 (1998)
23. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston (1991)
24. Jiang, Y., Du, B.: A efficiency complete algorithm for attribute reduction. In: 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009, pp. 377–379 (2009)
25. Wang, C.-Y., Tseng, S.-S., Hong, T.-P.: Flexible online association rule mining based on multidimensional pattern relations. Information Sciences 176, 1752–1780 (2006)
26. Bazan, J.G., Szczuka, M.S.: RSES and RSESlib - A Collection of Tools for Rough Set Computations. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 106–113. Springer, Heidelberg (2001)

Case-Based Argumentation Infrastructure for Agent Societies

Jaume Jordán, Stella Heras, and Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València, València Spain
{jjordan,sheras,vinglada}@dsic.upv.es

Abstract. Despite the important advances in the argumentation theory, it is difficult to find infrastructures of argumentation offering support for agent societies and their social context. Offering support for agent societies allows to represent more realistic environments to have argumentation dialogues. We propose an infrastructure to develop and execute argumentative agents in an open Multi-Agent System. This infrastructure offers the tools to develop agents with argumentation capabilities, including the communication skills and the argumentation protocol. It also offers support for agent societies and their social context.

Keywords: Argumentation, Multi-Agent Systems, Case-Based Reasoning.

1 Introduction

Argumentation theory has produced important benefits on many AI research areas, from its first uses as an alternative to formal logic for reasoning with incomplete and uncertain information to its applications in Multi-Agent Systems (MAS) [15]. Currently, the study of argumentation in this area has gained a growing interest. The reason behind is that having argumentation skills increases the agents' autonomy and provides them with a more intelligent behaviour.

An autonomous agent should be able to act and reason as an individual entity on the basis of its mental state (beliefs, desires, intentions, goals, etc.). As member of a MAS, an agent interacts with other agents whose goals could come into conflict with those of the agent. In addition, agents can have a social context that imposes dependency relations between them and preference orders among a set of potential values to promote/demote. For instance, an agent representing the manager of a company could prefer to promote the value of *wealth* (to increase the economic benefits of the company) over the value of *fairness* (to preserve the salaries of his employees). Therefore, agents must have the ability of reaching agreements that harmonise their mental states and that solve their conflicts with other agents by taking into account their social context. Argumentation is a natural way of reaching agreements between several parties. The argumentation techniques, hence, can be used to facilitate the agents' autonomous reasoning and to specify interaction protocols between them.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 13–24, 2012.

Currently, the ASPIC project[1] [3] made an effort to consolidate the work done in argumentation languages and protocols, argument visualisation and editing tools and, generally, in argumentation frameworks for MAS. Nevertheless, we do not know about any infrastructure of argumentation in open MAS offering support for agent societies and their social context (dependencies and values) to generate, select and evaluate arguments.

In this work, we propose a case-based infrastructure to develop and execute argumentative agents in a MAS. This work is an extension of those presented in [12,13]. We use the argumentation framework as the base to build a case-based infrastructure to deal with different kind of problems. A Case-Based Reasoning (CBR) system tries to solve a problem by means of reusing the solution of an old similar case [14]. This solution is previously stored in a memory of cases (case-base) and it can either be retrieved and applied directly to the current problem, or revised and adapted to fit the new problem. This infrastructure offers the necessary tools to develop agents with argumentation capabilities, including the communication skills and the argumentation protocol, and it offers support for agent societies and their social context. The main advantage of having this infrastructure is that it is possible to create agents with argumentation capabilities to solve a specified problem. Also, the infrastructure allows agents to store in the form of cases the information about the argumentation dialogues that they hold. Therefore, agents can use their argumentation experience to enhance current argumentation dialogues. This approach can obtain better results than other distributed approaches due to the argumentation process between agents and their reasoning skills. In the argumentation dialogue the agents try to reach an agreement about the best solution to apply for each proposed problem. Our approach is a hybrid system [7,1] that integrates CBR methodology, argumentation and MAS.

This paper is structured as follows. In Section 2 the argumentation framework implemented in the infrastructure for argumentative agents is presented. In Section 3 the complete infrastructure developed in this work is explained. Section 4 shows an example of the use of the presented infrastructure in a customer support application, and also an evaluation of the performance of the developed application. Finally, Section 5 presents the conclusions extracted from this work.

2 Argumentation Framework

In this section, we explain the computational framework, proposed in [11], for the design of MAS in which the participating software agents are able to manage and exchange arguments between themselves, taking into account the agents' social context. First, we define precisely our notion for an agent society. After that, we introduce the knowledge resources that agents can use to generate, select and propose their positions (solutions proposals) and arguments to support them. Furthermore, we present the argument types and their support set, that is a set

[1] European Union's 6th Framework ASPIC Project (IST-002307), http://www.fri.uni-lj.si/en/laboratories/ailab/136/project.html

of elements that support the argument. Finally, the argumentation protocol that agents follow is shown. This protocol is the mechanism to manage arguments and defines the argumentation dialogue that agents follow.

An *agent society* is defined in terms of a set of *agents* that play a set of *roles*, observe a set of *norms* and a set of *dependency relations* between roles and use a *communication language* to collaborate and reach the global objectives of the *group*. This definition, based on the approach of [8] and [5], can be applied to any open MAS where there are norms that regulate the behaviour of agents, roles that agents play, a common language that allow agents to interact defining a set of locutions and a formal semantics for each of these elements.

Furthermore, we consider that the values that individual agents or groups want to promote or demote and preference orders over them have also a crucial importance in the definition of an argumentation framework for agent societies. These values represent the motivation of agents to act in a specific way. Also, dependency relations between roles could imply that an agent must change or violate its value preference order. For instance, a manager could impose their values to an expert or a base operator could have to adopt a certain preference order over values to be accepted in a group. Therefore, we endorse the view of [6], who stress the importance of the audience in determining whether an argument is persuasive or not for accepting or rejecting someone else's proposals.

In open multi-agent argumentation systems the arguments that an agent generates to support its position can conflict with arguments of other agents and these conflicts are solved by means of argumentation dialogues between them. The argumentation framework includes a domain-cases case-base, with cases that represent previous problems and their solutions. The domain-cases are used to generate positions (solutions) to solve a problem and arguments to support them or attack other positions and arguments. The structure of these cases is domain-dependent and consist of a set of features that describe the problem to solve and the solution applied.

Arguments that agents interchange are defined as tuples: $\text{Arg} = \{\phi, v, < S >\}$, where ϕ is the conclusion of the argument, v is the value (e.g. economy, quality) that the agent wants to promote with it and $< S >$ is a set of elements that support the argument (*support set*). A support set (S) is defined as a tuple: S=$< \{P\}, \{DC\}, \{AC\}, \{DP\}, \{CE\} >$; with the following elements: *Premises (P)* are features that describe the problem to solve. These are the features that characterise the problem and that the agent has used to retrieve similar domain-cases from its case-base. Note that the premises used might be all features of the problem description or a sub-set. *Domain cases (DC)* are cases that represent previous problems and their solutions whose features match with some or all features of the problem description. *Argument-cases (AC)* are cases that represent past argumentation experiences with their final outcome. These cases are used to select the best position and argument to propose in view of the current context and the argumentation experience of the agent. *Distinguishing premises (DP)* are premises that can invalidate the application of a knowledge resource to generate a valid conclusion for an argument. These premises are extracted from

a domain-case that propose a different solution for the argument to attack. They consist of features of the problem description that where not considered to draw the conclusion of the argument to attack. *Counter-examples (CE)* are cases that match the problem description of a case but have different conclusions.

Agents generate *support arguments* when they are asked to provide evidence to support a position since, by default, agents are not committed to show evidences to justify their positions. Therefore, an opponent has to ask a proponent for an argument that justifies its position before attacking it. Then, if the proponent is willing to offer support evidences, it can generate a support argument which support set is the set of features (premises) that describe the problem and match the knowledge resources that it has used to generate and select its position. Note that the set of premises could be a subset of the features that describe the problem (e.g. when a position has been generated from a domain-case that has a subset of features of the problem in addition to other different features).

When the proponent of a position generates an argument to justify it and an opponent wants to attack the position or the argument, it generates an *attack argument*. Arguments can be attacked by putting forward distinguishing premises and counter-examples. The attack arguments that the opponent can generate depend on the elements of the support set of the argument of the proponent:

- If the justification for the conclusion of the argument is a set of premises, the opponent can generate an attack argument with a distinguishing premise that it knows. It can do it, for instance, if it is in a privileged situation and knows extra information about the problem or if it is implicit in a case that it used to generate its own position, which matches the problem specification.
- If the justification is a domain-case or an argument-case, then the opponent can check its case-base of domain-cases and try to find counter-examples to generate an attack argument with them.

The agents of the framework need a mechanism to manage the arguments and perform the argumentation dialogue. Therefore, an *argumentation protocol* has been defined in [12]. This protocol is represented by a set of locutions that the agents use to communicate each other depending on their needs, and an state machine that defines the behaviour of an agent in the argumentation dialogue.

3 Infrastructure

In this section, the infrastructure created to support the argumentation framework is described. The components of the infrastructure and the interactions between them are represented in Figure 1. The main components of our infrastructure explained in this Section are the argumentative agents, the Commitment Store and the knowledge interchange mechanism. As we can see in Figure 1, there are different organizations or groups composed by some argumentative agents. Also, the Commitment Store interacts with all the argumentative agents to store the positions and the arguments generated in the argumentation dialogue. The knowledge interchange is performed by using concepts of a defined ontology that is used as a language representation of the cases.

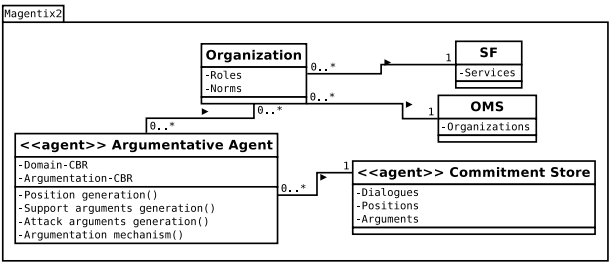


Fig. 1. Infrastructure diagram

The agent platform used in the implemented infrastructure is Magentix2[2]. This is a platform that provides services and tools that allow for the secure and optimized management of open MAS. There are two modules of the platform represented in Figure 1, the *Organization Manager Service* (OMS) and the *Service Facilitator* (SF). The OMS is in charge of managing the organizations, groups, roles and norms of the system. The SF registers the different services that can offer the agents and acts as yellow pages to find services. A more detailed description of the OMS and the SF can be found in [4]. The argumentative agents and the Commitment Store agent are extensions of the Magentix2 CAgent[3].

3.1 Argumentative Agents

Argumentative agents are the most important element of our infrastructure. These agents have all the tools needed to engage in an argumentation dialogue and reach an agreement with other agents about the best solution to apply for a problem. The solution applied to solve a problem in the past and the information about the problem-solving process can be reused to propose a solution to another similar problem. CBR systems have been widely applied to perform this task [2,16]. The argumentative agents have two CBR based modules: Domain CBR and Argumentation CBR, shown in Figure 1 and described in what follows.

Domain CBR. The argumentative agents have their own domain CBR. This CBR store cases that represent previous solved problems. A domain-case is composed basically by a set of features that describe the problem that the case solved and the final solution applied. The cases in the case-base are organised by their features. A case is equal to another if it has the same features with the same values in each feature. Thus, to retain cases in the case-base the features are used as indexes to organise the cases. Also the features are used to extract similar cases from the case-base in the retrieve phase. An algorithm based on the Euclidean distance is used to measure the similarity degree between different cases of the

[2] http://users.dsic.upv.es/grupos/ia/sma/tools/magentix2/index.php
[3] http://users.dsic.upv.es/grupos/ia/sma/tools/magentix2/archivos/javadoc/es/upv/dsic/gti_ia/cAgents/CAgent.html

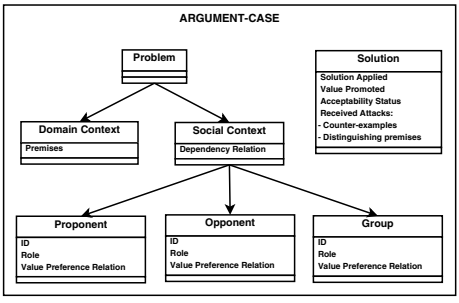


Fig. 2. Structure of the Argument-case

domain case-base. This algorithm is easy to implement and it works well with different domains. Other algorithms could be used, but it is not the objective of this work to evaluate the different alternatives. To retrieve domain-cases of the case-base, we perform queries with a set of features that must match the problem description of the cases retrieved. The list of similar domain-cases retrieved is limited by a similarity degree, which is a threshold that can be specified depending on the application domain.

Argumentation CBR. The Argumentation CBR consists of a CBR module with argumentation data. This CBR stores arguments that were used in previous argumentation dialogues as argument cases. The structure of an argument-case is shown in Figure 2. The argumentative agents have their own argumentation CBR. This knowledge is used to generate better arguments in the argumentation dialogues taking into account similar previous argumentation experiences where similar solutions were proposed. Thus, the best argument to propose in the current problem to solve will be selected in view of the acceptance that had a similar argument in the past. Therefore, argument-cases store information related to the domain and the social context where previous arguments (and their solution) were proposed. The problem description of an argument-case includes information about the domain context where the argument was put forward and information about the social context where the solution was applied (the agents that participated in the dialogue, their roles or their value preferences). The latter information can determine if certain arguments are more persuasive than others for particular social contexts (their acceptability status was set to *accepted* at the end of the dialogue where the argument was put forward).

3.2 Argument Management Process

The argument management process that the argumentative agents perform is described below. It includes: the position generation, the support arguments generation, the attack arguments generation and the argumentation mechanism.

Position Generation. A position is a solution that defends an agent as the correct one to apply to the problem to solve. The position generation is made in

two steps. First, the agent searches in its domain CBR the most similar domain-cases to the current problem. With them, the agent is able to propose a list of potential positions. Then, the agent evaluates the suitability of each position using its argumentation CBR. To do that, such argument-cases which features match with the domain-cases extracted are retrieved. Each position is evaluated in function of its chances of being well defended. As this evaluation is based on argument-cases, the best position to propose will be selected in view of the acceptance that a support argument for a similar position had [10, Chapter 3].

Support Arguments Generation. A support argument is an argument that justifies a position. The support set of this kind of argument can be formed by argument-cases, domain-cases and premises. These cases and premises are used as pieces of evidence to justify the solution defended by the position. To generate a support argument for a position, the argumentative agents search for similar argument-cases that can justify the current position. Also, they include the domain-case that is currently supporting the position in the support set. Finally, a list of possible support arguments is generated with different combinations of the available support elements in the support set. This list is ordered by a suitability degree [10, Chapter 3] in terms of the acceptance that a similar argument had in the past.

Attack Arguments Generation. An attack argument is an argument that attacks a support argument or another attack argument. The attack argument has a different solution than the argument attacked. To generate the attack argument, the premises that the argument to attack has and the social context (the relation with the other agent) are taken into account. The first type of attack that the agent will try to generate is a counter-example attack, and if it is not possible it will generate a distinguishing premises attack. An agent is able to generate a list of potential attack arguments depending on the knowledge resources used to generate them. Therefore, it has to query its argument-cases case-base and taking into account the acceptance that similar arguments had in the past, sort the list by suitability degree (as done for support arguments) [10, Chapter 3]. Finally, the most suitable attack argument is used first.

Argumentation Mechanism. The agents of the framework need a mechanism to manage the arguments and perform the argumentation dialogue. To deal with this functionality, an argumentation protocol has been defined in [10, Chapter 3]. This protocol is represented by a set of locutions that the agents use to communicate with other agents, and a state machine that defines the behaviour of an agent in the argumentation dialogue. In each state, the different locutions that can be received and generated are specified. Inside each state of the protocol, the corresponding actions of the argument management process are performed by using the necessary calls to the different functions implement them.

3.3 Commitment Store

The Commitment Store is a resource of the argumentation framework that stores all the information about the agents participating in the problem-solving process,

the argumentation dialogues between them, their positions and arguments. By making queries to this resource, every agent can read the information of the dialogues that it is involved in. In the infrastructure, it has been implemented as an agent to allow a good communication with the other agents. Concretely, it is an extension of the Magentix2 CAgent, as the argumentative agents are.

3.4 Knowledge Interchange Mechanism

The case-bases of the domain CBR and the argumentation CBR are stored as OWL 2[4] data of an ontology[5] that we have designed to act as language representation of the cases. In this way, heterogeneous agents can use it as common language to interchange solutions and arguments generated from the case-bases of the argumentation framework. The main advantage of using ontologies is that the structures and features of the cases are well specified and agents can easily understand them. The infrastructure includes ontology parsers to provide an API to read and write data in the case-bases. To implement them, we have used the OWL API[6], which is a Java API and reference implementation for creating, manipulating and serialising OWL Ontologies.

4 Call Centre Example

In this section, we validate our infrastructure by implementing a call centre application with it.

4.1 Customer Support Application

Nowadays, companies have to offer a good customer support to take an advantage over their competitors. A good customer support depends, in many cases, on the experience and skills of its operators. A quick and accurate response to the customers problems ensures their satisfaction and a good reputation for the company and, therefore, it can increase its profits.

A common customer support system settled in a company consists of a network of operators that must solve the incidences (also known as *tickets*) received in a Technology Management Centre (TMC). This help is commonly offered via a call centre, where the operators have computers provided with a helpdesk software and phone terminals connected to a telephone switchboard that balances the calls among operators. Commonly, the staff of a call centre is divided into three levels: 1) Base operators, who receive customer queries and answer those ones from which they have background training; 2) Expert operators, who are the technicians in charge of solving new problems; 3) Managers, who are in charge of organising working groups, of assigning problems to specific operators and of creating generic solutions.

[4] http://www.w3.org/TR/owl2-overview/
[5] http://users.dsic.upv.es/∼vinglada/docs/
[6] http://owlapi.sourceforge.net/

The solution applied to each problem and the information about the problem-solving process could provide suitable information to improve the customer support offered by the company. The suitability of CBR systems in helpdesk applications to manage call centres has been guaranteed for the success of some of these systems from the 90s to nowadays [2,16].

These approaches propose systems for human-machine interaction where the CBR functionality helps the operators to solve problems more efficiently by providing them with potential solutions via the helpdesk software. We have applied the infrastructure proposed in this paper to extend a previous work that presented a CBR module that acts as a solution recommender for customer support environments [9]. The CBR module is flexible and multi-domain. However, to integrate the knowledge of all experts in a unique CBR module can be complex and costly in terms of data mining (due to extra large case-bases with possible out-of-date cases). Moreover, to have a unique but distributed CBR could be a solution, but to assume that all operators are willing to share unselfishly their knowledge with other operators is not realistic, since in many companies they are under privacy contracts depending on the project that they work or they get incentives if their performance overpasses the one of their colleagues. In this case, the modelling of the system as a MAS will be adequate. Finally, several experts could provide different solutions and hence, they need a mechanism to negotiate and reach an agreement about the best solution to apply.

In our prototype, the operators and experts of a call centre are represented by agents that use an automated helpdesk and argue to solve an incidence. Every agent has individual CBR resources and preferences over values (e.g. economy in the resources used, quality of the solution applied, solving speed). A solution for a problem promotes one value. Thus, each agent has its own preferences to choose a solution to propose. Furthermore, agents can play two different roles: *operator* and *expert*. The main difference between an operator and an expert is that the second one has more specific domain knowledge to solve certain types of problems. Also, dependency relations between roles could imply that an agent must change or violate its value preference order. For instance, an expert could impose their values to an operator and the last could have to adopt a certain preference order over values. The data-flow of the argumentation process of the helpdesk application is explained in [13].

4.2 Evaluation

To validate the infrastructure, we make an evaluation of the prediction error of our prototype. The domain-cases case-bases of each agent are populated randomly by using some of the 48 cases of a case-base of real computer problems reported to the call centre, increasing the number of cases from 5 to 45 cases in each round. Each problem is described by a set of features (e.g. the type of problem, the log of the system, etc.) and the description of the solution applied. To diminish the influence of random noise, all results report the average of 48 simulation runs per round. In each round, an tester agent acts as initiator of the process. This agent has access to the whole case-base and in each run takes

the corresponding case to solve and sends it to the other agents. In this way, the initiator knows which was the real solution applied to the problem and can compare this value to the solution decided by the agreement process. To make the evaluation the tests have been performed with the following decision policies:

- CBR-Random (CBR-R): which consists in choosing randomly a solution of domain-cases case-base, without using argumentation.
- CBR-Majority (CBR-M): which consists in selecting the solution most frequently proposed by the agents, again using the domain-cases case-base, and also without any argumentation process.
- CBR-Argumentation (CBR-A): where agents are provided with the proposed case-based argumentation functionalities and perform an argumentation dialogue to select the best solution of those proposed by the group.

In all decision policies, agents propose solutions using their own CBR. So, an agent will be able to propose a solution if in its CBR there is a case that match with the ticket to solve. In the performed tests, we use two different configurations: a group of 7 operators; and a group of 6 operators and 1 expert. The main difference between these two configurations is that in the second configuration, an agent has been allowed to play the role of an *expert*, while the rest of agents play the role of *operators*. An expert is an agent that has specific knowledge to solve certain types (categories) of problems and has its case-base of domain-cases populated with cases that solve them. Thus, the expert domain-cases case-base has as much knowledge as possible about the solution of past problems of the same type. That is, if the expert is configured to have 5 domain-cases in its domain-cases case-base, and there are enough suitable information in the original tickets case-base, these cases represent instances of the same type of problems. In the case that the tickets case-base has less than 5 cases representing such category of problems, 3 for instance, the remaining two cases solve problems of the same category and so on.

In our case, the expert agent has an authorisation dependency relation over other technicians. This dependency relation means that when an agent has committed itself to other agent for a certain service, a request from the latter leads to an obligation when the conditions are met. Therefore, if the expert agent is able to propose a solution for the ticket requested, it can generate arguments that support its position and that will defeat other operators' arguments, due to its authorisation dependency relation over other technicians. However, in the CBR-R and the CBR-M policies there is not argumentation dialogue, so this dependency relation is not taken into account, but the proposals of the expert have the same influence than other operators proposals in the final solution selected.

For the tests shown in Table 1, we evaluate the average error in the prediction of the best solution to apply with regard to the size of the case-bases of domain-cases of the agents. As we can expect, the prediction error of the system decreases as the number of domain-cases grows. In addition, the prediction error of the CBR-Argumentation policy is always lower or equal than the other policies. Argumentation allows agents to argue and hence, the solution with more justification elements prevails. Thus, the best solution has more probability of

Table 1. Prediction error

	7 operators			**6 operators and 1 expert**		
cases	CBR-R	CBR-M	CBR-A	CBR-R	CBR-M	CBR-A
5	47,92%	39,58%	**37,50%**	39,58%	39,58%	**29,50%**
10	25,00%	16,67%	**14,58%**	18,75%	18,75%	**9,67%**
15	12,50%	6,25%	**2,08%**	6,25%	6,25%	**2,08%**
20	6,38%	6,38%	**2,13%**	6,38%	**2,13%**	2,13%
25	4,17%	2,08%	**0%**	4,17%	**0%**	0%
30	4,17%	0%	0%	4,17%	0%	0%
35	4,17%	0%	0%	4,17%	0%	0%
40	2,08%	0%	0%	2,08%	0%	0%
45	4,17%	0%	0%	4,17%	0%	0%

being proposed and the error decreases. Furthermore, the general prediction error of the group with an expert is lower than in the other group without any expert. The reason behind that improvement in the results is that the expert is providing the best solution that it knows and imposing its opinion about which is the best solution to apply and, since it has more specialized knowledge, the quality of the final solution agreed increases.

5 Conclusions

In this work, we have implemented an infrastructure to develop and execute argumentative agents in an open MAS based on the argumentation framework described in Section 2. This infrastructure offers the necessary tools to develop agents with argumentation capabilities, including the communication skills and the argumentation protocol. Also, it offers support for agent societies and takes into account the agents' social context. The infrastructure combines the CBR methodology, argumentation and MAS.

Furthermore, the proposed infrastructure has been validated with an example in a customer support application. In the performed tests, the best results are obtained using an argumentation policy that takes into account the social context of agents. In addition, having at least an expert involved in the group of agents that tries to solve a problem increases the quality of the final solution agreed.

As a future work, our infrastructure will be used in other domains to measure the performance and the differences having a largest database of solved problems.

Acknowledgement. This work is supported by the Spanish government grants CONSOLIDER INGENIO 2010 CSD2007-00022, TIN2008-04446 and TIN2009-13839-C03-01 and by the GVA project PROMETEO 2008/051.

References

1. Abraham, A., Corchado, E., Corchado, J.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Acorn, T.L., Walden, S.H.: Smart: support management automated reasoning technology for compaq customer service. In: Proceedings of the 4th Conference on Innovative Applications of Artificial Intelligence, IAAI 1992, pp. 3–18. AAAI Press (1992)

3. Amgoud, L., Bodensta, L., Caminada, M., McBurney, P., Parsons, S., Prakken, H., van Veenen, J., Vreeswijk, G.: Project N 002307 ASPIC, Argumentation Service Platform with Integrated Components. Deliverable D2.6. Tech. rep., ASPIC Consortium (February 15, 2006)
4. Argente, E., Botti, V., Carrascosa, C., Giret, A., Julián, V., Rebollo, M.: An Abstract Architecture for Virtual Organizations: The THOMAS approach. Knowledge and Information Systems, 1–35 (2011)
5. Artikis, A., Sergot, M., Pitt, J.: Specifying norm-governed computational societies. ACM Transactions on Computational Logic 10(1) (2009)
6. Bench-Capon, T., Atkinson, K.: Abstract argumentation and values. In: Argumentation in Artificial Intelligence, pp. 45–64. Springer, Heidelberg (2009)
7. Corchado, E., Abraham, A., Carvalho, A.C.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
8. Dignum, V.: PhD Dissertation: A model for organizational interaction: based on agents, founded in logic. Ph.D. thesis, Proefschrift Universiteit Utrecht (2003)
9. Heras, S., García-Pardo, J.A., Ramos-Garijo, R., Palomares, A., Botti, V., Rebollo, M., Julián, V.: Multi-domain case-based module for customer support. Expert Systems with Applications 36(3), 6866–6873 (2009)
10. Heras, S.: Case-Based Argumentation in Agent Societies. Ph.D. thesis, Universitat Politècnica de Valencia (2011)
11. Heras, S., Botti, V., Julián, V.: An Abstract Argumentation Framework for Supporting Agreements in Agent Societies. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS(LNAI), vol. 6077, pp. 177–184. Springer, Heidelberg (2010)
12. Jordán, J., Heras, S., Julián, V.: A customer support application using argumentation in multi-agent systems. In: Fusion 2011, pp. 772–778 (2011)
13. Jordán, J., Heras, S., Valero, S., Julián, V.: An Argumentation Framework for Supporting Agreements in Agent Societies Applied to Customer Support. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS(LNAI), vol. 6678, pp. 396–403. Springer, Heidelberg (2011)
14. Kolodner, J.: Case-based Reasoning (1993)
15. Rahwan, I., Simari, G. (eds.): Argumentation in AI. Springer, Heidelberg (2009)
16. Roth-Berghofer, T.R.: Learning from HOMER, a Case-Based Help Desk Support System. In: Melnik, G., Holz, H. (eds.) LSO 2004. LNCS, vol. 3096, pp. 88–97. Springer, Heidelberg (2004)

The Application of Multi-Agent System in Monitoring and Control of Nonlinear Bioprocesses

Piotr Skupin and Mieczyslaw Metzger

Faculty of Automatic Control, Electronics and Computer Science
Silesian University of Technology,
ul. Akademicka 16, 44-100 Gliwice, Poland
{piotr.skupin,mieczyslaw.metzger}@polsl.pl

Abstract. Most of the continuous processes (e.g. chemical processes) are monitored and controlled in the classical way, i.e. entirely by the process operator based on the measurement data. However, in many cases, due to the nonlinear nature of some continuous processes, the use of this approach may not always be sufficiently efficient. In particular, it concerns a large class of biological processes (bioprocesses) for which more complex data analysis is required. Hence, this paper presents the possibility of application of a Multi Agent System (MAS) as a support for the process operator. The proposed solution being a combination of the classical approach and the MAS, called Hybrid Intelligent Operations and Measurements (HIOM), is tested based on the simulation runs of a mathematical model of the bioprocess.

Keywords: Multi Agent System, Hybrid Intelligent Operations and Measurements, agent-based process control, CSTB, self-sustained oscillations, biomass productivity.

1 Introduction

The present concept of agents, as independent and intelligent applications has been rapidly developed since the beginning of the 1990's. According to the first authors [1-3] agent applications were defined as applications operating both independently and collaboratively in a group to exchange information between each other and to attain a common goal. Currently, agent-based solutions can be effectively applied in modern industrial control systems [4-7] and the latest review paper in this area is [8]. Especially, it concerns the manufacturing systems, for which the agent applications provide a new way for more flexible control of such systems [4]. On the other hand, one can distinguish a large group of continuous processes [9] (e.g. chemical or sedimentation processes), for which PID or predictive controllers still provide the primary way to control the key process variables and are the main support for operating control (Fig.1a). In this case, the process operator monitors and supervises the course of the process based on the measurement data. He can also exert an influence on the process by changing the set point values for controllers or even by changing the control strategy. This approach is most common in practice and is referred to as the classical

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 25–36, 2012.

approach. However, for some continuous processes the use of the basic (classical) approach may not always be sufficiently efficient. In particular, it concerns a large class of biological processes (bioprocesses), which take place in continuous stirred tank bioreactors (CSTBs).

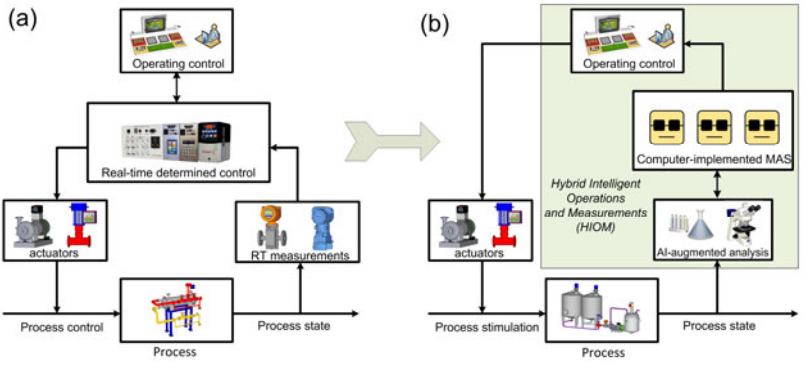


Fig. 1. Two possible approaches for continuous processes: (a) classical approach (process operator is entirely responsible for operating control) and (b) MAS-based control

Such CSTB systems allow for continuous culture of microorganisms, which increase their concentration (biomass concentration), and concentration of a secreted product, by consumption of available substrates. Due to the nonlinear dynamics of such processes, self-sustained oscillations (SSO) of microorganism concentration (biomass concentration) are often observed during laboratory experiments [10]. It means that the CSTB can operate in the range of SSO or in the range of steady states. However, in the case of improving bioreactor productivity, the choice of the mode of operation (induction or elimination of SSO) becomes an important factor to be taken into consideration [11-13].

Hence, in the classical approach, the process operator must detect oscillations by himself and to determine if these oscillations result from non-linear dynamics of the process or other reasons (e.g. inaccuracy of control systems equipment [14]). Then, if the SSO are present, the process operator must calculate the average values of biomass concentration (or biomass productivity) and take appropriate decisions to maximize these values and to choose the desired mode of operation. The situation becomes more complicated, if there are more bioreactors to be monitored and controlled. Hence, it seems interesting to apply a Multi Agent System (MAS) consisting of a sufficient number of agent applications performing simple actions and supporting the process operator [15]. The proposed combination of the classical approach and the MAS for the nonlinear biochemical process will be referred to as the Hybrid Intelligent Operations and Measurements (HIOM) (Fig.1b). And as has been shown in literature (see e.g. [16-18]), a hybrid approach is usually more efficient.

In this paper, based on the simulation runs of the mathematical model of the CSTB, the MAS as a support for the process operator has been presented. The MAS is consisted of three independent and cooperating with each other agent applications (two

monitoring agents and one control agent). Of course, in the case of a greater number of CSTBs or when taking into consideration more parameters to be calculated, the MAS will be composed of a greater number of agent applications.

In order to assign the tasks to individual agents, it is necessary to look a bit deeper into the process under consideration. Knowing the dynamical properties of the process, the choice of number of agents and the assignment of their functions will be an easier task. This, in turn, requires some more detailed discussion, which will be given in the next section.

2 The Control of the Oscillatory Mode of Operation

Further studies on the effects of the oscillatory behavior of bioprocesses revealed some advantages and disadvantages resulting from the mode of operation of the CSTB. It turns out that the operation in the range of SSO may lead to higher or lower average values of biomass concentration in comparison to the results obtained in steady-state regime [11]. Hence, the main task of the control agent is to choose a desired mode of operation of the CSTB (operation in steady-state regime or in the range of SSO). This, in turn, requires to find an effective influence on the bioprocess, which will allow for induction or elimination of the SSO.

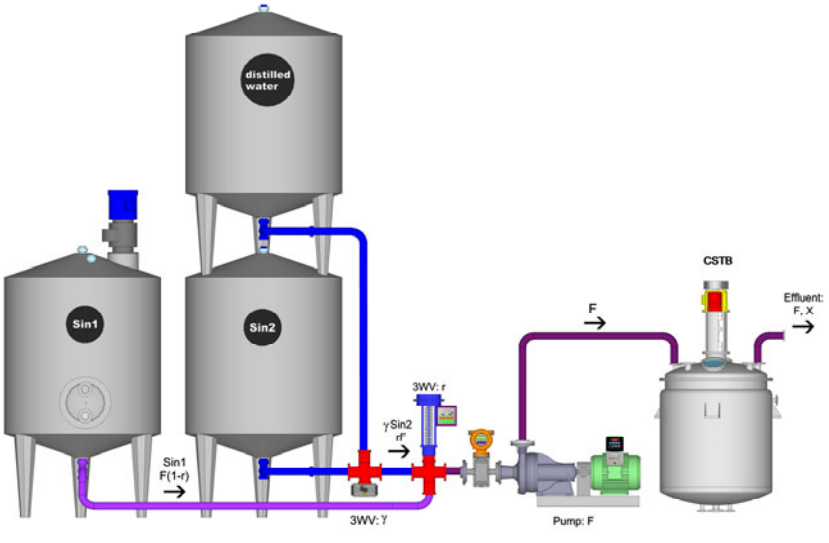


Fig. 2. The general scheme of the well-mixed CSTB fed with the mixture of two different substrates of similar properties. The second substrate is diluted with distilled water.

Reviewing literature in this area, several different approaches can be distinguished and the most common methods are those based on the changes in: flow rates F of the medium, pH levels or dissolved oxygen concentration (by changing the agitation speed of the reactor content) [14], [19]. Because, the SSO of biomass concentration take place for some range of the flow rates F (or dilution rates D=F/V, V – volume of

the medium in the CSTB, V=const), hence, based on the results obtained in [13], it is applied a method involving a mixture of two different substrates of similar properties fed into the CSTB at the flow rate F. The scheme of the CSTB fed with the mixture of two substrates of concentrations S_{in1} and S_{in2}, respectively, has been shown in Figure 2. The important components of the system are two three-port valves: 3WVr and 3WVγ. The three-port valve 3WVγ is responsible for diluting the substrate of concentration S_{in2} by mixing it with distilled water. The desired level of dilution is obtained by setting an appropriate degree of valve opening, which is represented by the parameter $\gamma \in [0,1]$. As a result, assuming a very small volume inside the valve (neglecting the valve's dynamics), the output substrate concentration is γS_{in2}.

In turn, the valve 3WVr is responsible for setting an appropriate contribution of individual substrates to the mixture, which is represented by the parameter $r \in [0,1]$. It means that the substrate of inlet concentration S_{in1} is fed into the CSTB at the flow rate (1-r)F, and the diluted substrate of inlet concentration γS_{in2} – at the flow rate rF. In other words, the diluted substrate is fed into the CSTB at the expense of the original substrate to minimize the cost of such an approach.

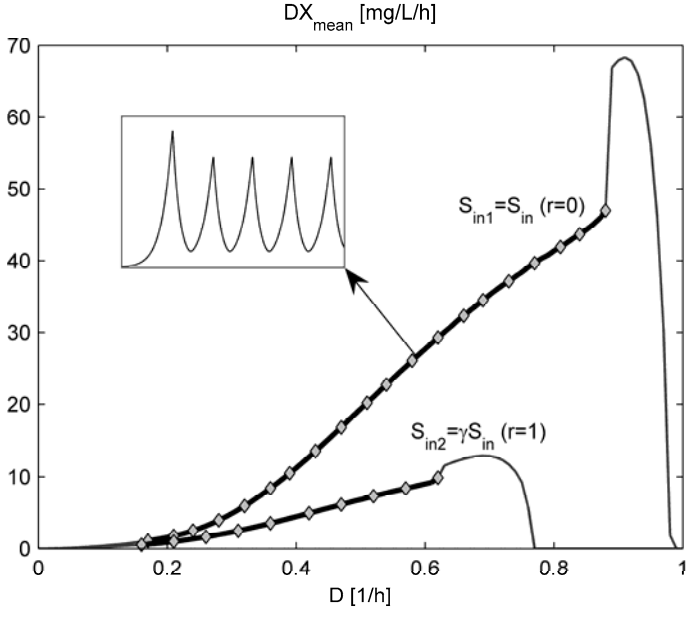


Fig. 3. The average biomass productivity as a function of dilution rate D for two extreme cases, when r=0 or r=1. The black thick lines with diamond symbols show the range of SSO. The additional window presents an exemplary time course of the biomass productivity in the range of SSO.

From the bioreactor productivity point of view, an important variable taken into account is the biomass productivity, which is equal to the mass of biomass produced per unit time, per unit volume of the medium in the CSTB [12]. Figure 3 presents an example of the dependency of the average biomass productivity on the dilution rate D

for two extreme cases, i.e. for r=0 (the CSTB is fed only with the substrate of concentration S_{in1}) and for r=1 (the CSTB is fed only with the diluted substrate of concentration γS_{in2}). It can be clearly seen that the range of the occurrence of oscillatory behavior depends on both the dilution rate D and the inlet substrate concentration. Moreover, for dilution rates D higher than a critical value Dc, the washout phenomenon occurs, i.e. for D≥Dc, the whole bioreactor content is washout and the biomass concentration drops to zero. We further assume that the manipulated variables are the dilution rate D and the degree of the 3WVr valve opening (parameter r). The changes in dilution rate D will allow us to choose an operating point and the changes in r will allow us to choose a desired mode of operation of the CSTB. However, it is convenient to consider a constant value of γ parameter (degree of the 3WVγ valve opening). The choice of the γ parameter will be described in the next section and will be based on the simulation runs of the mathematical model of the bioprocess.

3 Mathematical Model

For the CSTB, as shown in Figure 2, a well-known mathematical model with the specific growth rates described by the Monod relations [20] and with the weight coefficients proposed in [21] is as follows:

$$\frac{dS_1}{dt} = D\left((1-r)S_{in1} - S_1\right) - \frac{w_1}{\alpha+\beta S_1} \cdot \frac{\mu_{m1} S_1}{S_1 + K_{s1}} X \tag{1}$$

$$\frac{dS_2}{dt} = D\left(\gamma r S_{in2} - S_2\right) - \frac{w_2}{\alpha+\beta S_2} \cdot \frac{\mu_{m2} S_2}{S_2 + K_{s2}} X \tag{2}$$

$$\frac{dX}{dt} = -DX + w_1 \frac{\mu_{m1} S_1}{S_1 + K_{s1}} X + w_2 \frac{\mu_{m2} S_2}{S_2 + K_{s2}} X \tag{3}$$

where: S_1, S_2 – are the outlet concentrations of substrates (the CSTB is well-mixed) [mg/L], S_{in1}, γS_{in2} – are the inlet concentrations for S_1 and S_2, respectively [mg/L], X – outlet biomass concentration [mg/L], D – dilution rate (D=F/V, F – flow rate, V – volume of the bioreactor medium and V=const) [1/h], μ_{mi} – maximum specific growth rate (i=1,2) [1/h], K_{si} – half saturation constant (i=1,2) [mg/L], $Y_i=\alpha+\beta S_i$ – dimensionless variable yield coefficient (i=1,2 and α,β>0 – constant parameters). w_1, w_2 – dimensionless weight coefficients described by the following equations:

$$w_1 = 1, \; w_2 = \frac{1}{1 + e^{b(D - D_{trans})}} \tag{4}$$

$$D_{trans} = D_{C1} + (D_{C2} - D_{C1})r \tag{5}$$

$$D_{C1} = \frac{\mu_{m1}(1-r)S_{in1}}{(1-r)S_{in1} + K_{s1}},\ D_{C2} = \frac{\mu_{m2}\gamma r S_{in2}}{\gamma r S_{in2} + K_{s2}} \tag{6}$$

where: b – appropriately chosen constant parameter (in our case b=200), D_{C1}, D_{C2} – critical values of dilutions rates for the first and second (diluted) substrate, respectively, D_{trans} – transition dilution rate.

All the parameter values in equations (1)-(6) have been taken from [11] and [22], and the inlet concentrations of both substrates are $S_{in1}=S_{in2}=100$ [mg/L]. The mathematical model described by equations (1)-(3) with the weight coefficients (4) assumes that for $D<D_{trans}$ both substrates of concentrations S_1 and S_2 are consumed for growth of microorganisms (biomass concentration). If $D \geq D_{trans}$, then the growth of microorganisms results from the consumption of only one substrate – the substrate, which provides a faster growth of microorganisms. This phenomenon is known as the diauxic growth and it is well-described in the literature [22]. Other typical phenomena, i.e. washout and SSO of biomass concentration, which are observed during laboratory experiments, are also taken into account by the model (1)-(6) and shown in Figure 3.

3.1 The Choice of the γ-Parameter

In order to have the possibility to choose a desired mode of operation by mixing two substrates of similar properties (e.g. two different sugars), it is necessary to find an appropriate value of γ (a degree of dilution of the second substrate).

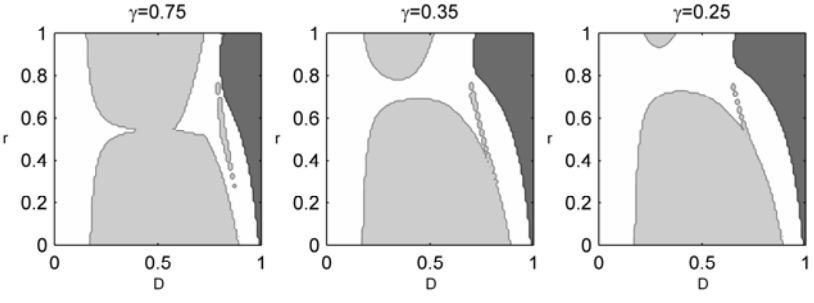


Fig. 4. Regions of steady states (white), SSO (light grey) and washout (dark grey) on parameter plane (D,r) when mixing two different substrates of similar properties for three different degrees of dilution of the second substrate

Then, by setting an appropriate contribution of both substrates to the mixture (parameter r), it is possible to completely eliminate the oscillatory behavior for any value of D. The problem of choice of the γ-parameter explains Figure 4. The smaller the value of γ, the greater is the dilution of the second substrate of initial concentration S_{in2}, which is equivalent to the fact that the regions of SSO are getting smaller (Fig.4). This, in turn, allows for choosing a desired mode of operation of the CSTB.

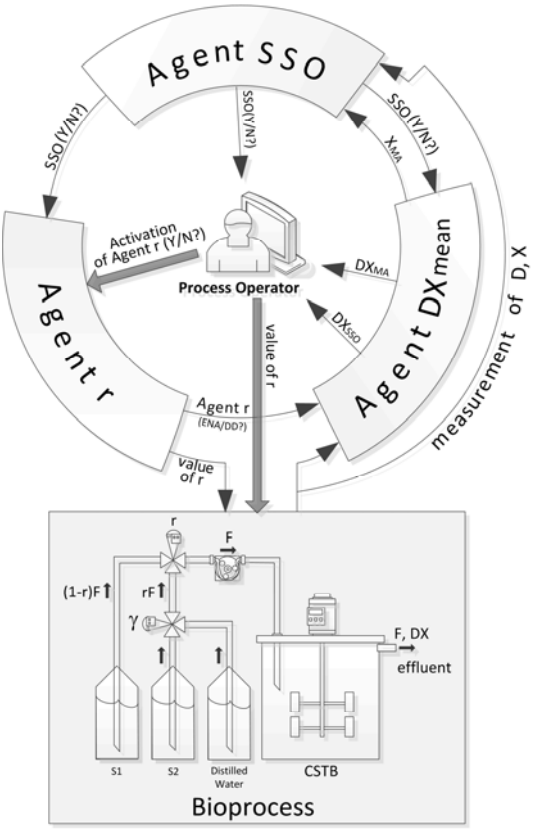


Fig. 5. The structure of the proposed MAS showing dependencies between the agent applications and the process operator

Taking into account the obtained results (Fig.4), for further calculations, we assume γ=0.35, which corresponds to the inlet concentration γS_{in2}=35 [mg/L] for the diluted substrate. It should also be emphasized that the higher contribution of the diluted substrate to the mixture, the lower are values of the biomass concentration X. However, in industrial applications, most of the CSTBs operate in the steady-state regime for small values of dilution rate D. Hence, by adding the diluted substrate to the bioreactor, we obtain the smaller values of biomass concentration X, but comparable values of the biomass productivity DX, because D can be changed in a wide range.

For the assumed mathematical model of the system, three agent applications have been proposed and their detailed description is given in the next section.

4 The Description of the Agent Applications

In the case of the occurrence of the SSO, the measurement of the biomass productivity on-line is not sufficient and some additional algorithms, for calculation of the average values, have to be applied. This, in turn, requires the prior detection of the oscillatory

behavior. Hence, in the presented solution, the application of two monitoring agents (Agent SSO and Agent DXmean) and one control agent (Agent r), which is responsible for the choice of a desired mode of operation, has been proposed. Moreover, it is assumed that the agent applications can cooperate with each other and with the process operator. The proposed distribution of agents results from functions (monitoring and control) performed by them and provides a better transparency of the MAS for user (process operator). The diagram of the MAS has been shown in Figure 5.

Agent SSO.
The Agent SSO is one of the monitoring agents, which is responsible for detection of the SSO of biomass concentration. The operation of this application is based on the measurements of X(t) and the average values of biomass concentration $X_{MA}(t)$, which are provided by the second monitoring agent (Agent DXmean) and calculated on-line by means of the moving average (MA). Hence, the detection of the oscillatory behavior is possible owing to the observation of a variable $X^*(t)=X(t)-X_{MA}(t)$, which in the case of the occurrence of SSO, oscillates around zero. The information about the detection of the SSO is sent to the control agent (Agent r) and to the second monitoring agent – Agent DXmean. This information can also be sent to the process operator, which has been presented in Figure 5.

Agent DXmean.
The second monitoring agent is responsible for the calculations of the average biomass concentrations $X_{MA}(t)$ and the average biomass productivity $DX_{MA}(t)$ on-line by means of MA for a sufficiently long time window. Then, the calculated values of $DX_{MA}(t)$ and $X_{MA}(t)$ are sent to the Agent SSO and to the process operator. However, in the range of SSO the calculated values of $DX_{MA}(t)$ oscillate, as well. Hence, in order to attenuate these oscillations, it is necessary to lengthen the time window for calculating the MA, but this will cause a slower response of the agent application to the changes in amplitude and frequency of X(t). Furthermore, it would be necessary to know the lowest oscillation frequency of X(t) a-priori to assume an appropriate length of the time window. Therefore, after receiving information about the occurrence of the SSO, the Agent DXmean also calculates the average value of the biomass productivity ($DX_{SSO}(t)$) as an average value for the period. The calculated values of $DX_{SSO}(t)$ are then sent to the process operator to evaluate the efficiency of the bioprocess.

Agent r.
The Agent r is the control agent, which is responsible for the choice of the most suitable mode of operation of the CSTB, i.e. for attenuation of the SSO of biomass concentration. Depending on the requirements, the Agent r is either active and eliminates oscillations or is blocked by the process operator and the CSTB can operate in the rage of the SSO. The damping of oscillations is possible by increasing the contribution of the diluted substrate to the mixture – the parameter $r \in [0,1]$ is increased by Δr every Δt interval. If the contribution of the diluted substrate is too high (too large value of r), then the process operator can change it manually.

The described agent applications perform their functions individually and decisions taken by them can be realized instantaneously without interference of the process operator. Based on the simulation runs of the mathematical model, the effectiveness of the MAS will be presented in the next section. It should also be emphasized that it is not easy to make a detailed comparison between the classical and agent-based solutions. Monitoring and analysis of the oscillatory behavior, but also its consequences on the bioreactor productivity, are not typical tasks performed by the classical controllers. In such case, the process operator is entirely responsible for analysis of the oscillatory behavior. Hence, the support provided by the proposed agent-based solution will be beneficial when making difficult decisions (e.g. determining the most suitable mode of operation for the CSTB).

5 Results

Because, the differential equations (1)-(3) are non-stiff, they are solved by the second-order Runge-Kutta method (RK2) with an integration step h=Dt/NIS, where Dt is an observation step and NIS is a number of integration steps in Dt interval [23]. It means that the equations (1)-(6) are numerically integrated with the step h<<Dt, and the calculated variables (e.g. biomass concentration X) are "sampled" every observation step Dt. All the obtained results refer to the case where the operation of the CSTB in the range of SSO is forbidden. Hence, the main task of the Agent r is elimination of the oscillatory behavior by increasing the amount of the diluted substrate in the mixture (an increase of r).

Figure 6 presents the contour line plot of the biomass productivity DX as a function of the dilution rate D and the parameter r. Hence, three different regions can be distinguished: the region of SSO (light-grey), the region of washout, DX=0 (dark-grey) and the region of steady states for DX>0 (white). The black broken line presents the changes in the dilution rate D made by the process operator and the changes in r made by the Agent r. For example, the change in dilution rate D from 0.22 to 0.55 [1/h] for r=0.65, triggers the SSO of biomass concentration instantaneously. Once the oscillations are detected by the Agent SSO, the Agent r starts increasing contribution of the diluted substrate to the mixture (an increase of r). As a result, the operating point of the bioprocess is moved to the region of steady states (Fig.6), while, at the same time, the higher values of biomass productivity DX are obtained.

From Figure 7 it can be clearly seen that the Agent DXmean calculates the average values $DX_{SSO}(t)$ only in the time intervals, in which the oscillatory behavior is detected by the Agent SSO. In the same time intervals, the Agent r increases the value of r by Δr until when the oscillations are no longer detected by the Agent SSO. It should also be emphasized that, the proposed MAS is not a model-dependent solution and will work properly for other mathematical models of bioprocesses having similar dynamical properties.

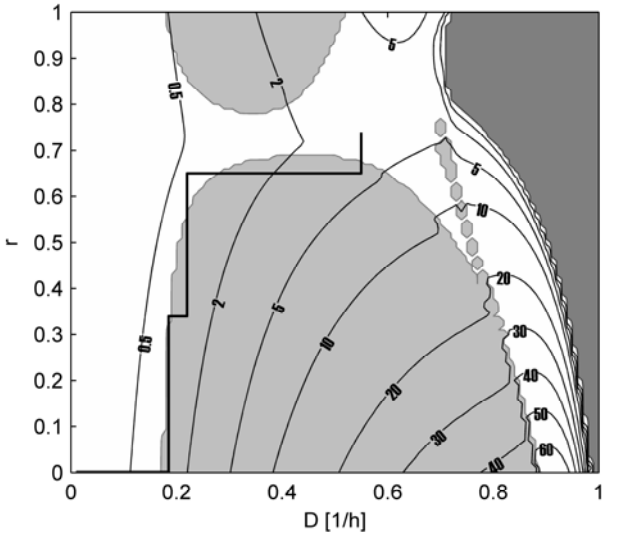


Fig. 6. The response of the Agent r (which is responsible for elimination of the SSO) on changes in the dilution rate D (black broken line) on the parameter plane (D,r). The isolines present the average values of biomass productivity.

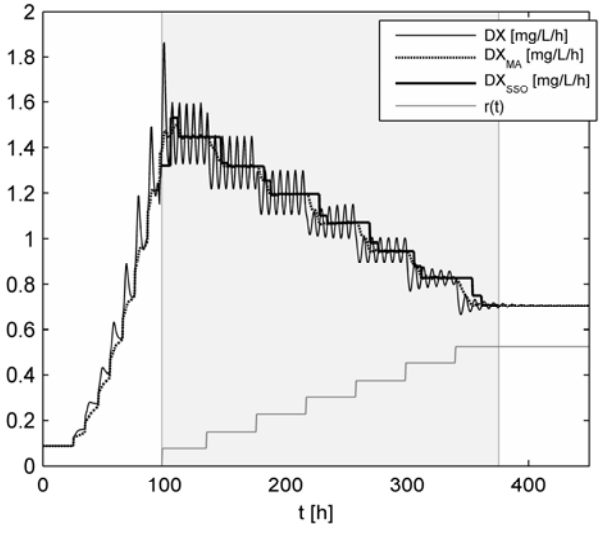


Fig. 7. The operation of the all agent applications in steady-state regime and in the range of SSO of biomass concentration. The light-grey region indicates the range of SSO detected by the Agent SSO. The ordinate axis for r(t) overlaps with ordinate axis for DX(t).

6 Concluding Remarks

Despite the high prevalence of agent applications in the manufacturing industry, this work presents the possibility of application of the hybrid system (HIOM) being a combination of the classical approach and the MAS for the continuous industrial process. Based on the example of a bioprocess, the MAS supports the process operator (e.g., when making decisions regarding the choice of operating regime for the CSTB). Using the mathematical model of the bioprocess, the operation of the MAS consisted of two monitoring agents and one control agent has been presented. Whenever the SSO took place, the Agent SSO detected these oscillations and the Agent DXmean could calculate the average values of biomass productivity as an average value for the period. In turn, the control agent (Agent r) could eliminate these oscillations by exerting an effective influence on the bioprocess, i.e. by mixing two different substrates of similar properties. Then, the average values of the biomass productivity and the possibility of control of the oscillatory behavior allow for the choice of most suitable operating conditions for the CSTB. Of course, the presented solution does not have to be limited to the three agent applications and it can be easily extended for systems composed of a greater number of CSTBs or when taking into account other aspects of the process, e.g. its productivity. Since the function DX(D) is characterized by a maximum (Fig.3), therefore there exist an optimal value of dilution rate D maximizing the biomass productivity DX. Hence, it is possible to introduce some extra agent applications, as a support for the process operator, which will be responsible for seeking the extremum of DX.

However, future work requires further investigation in order to test the effectiveness of the proposed system on a real industrial or a laboratory pilot plant. For instance, the system can be useful in laboratory experiments, where a few agent applications perform simple actions to support the analysis of the measurement data.

Acknowledgments. This work was supported by the Polish Ministry of Science and Higher Education under Grant N N514 471539 and Grant BK-UiUA.

References

1. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. 10, 115–152 (1995)
2. Jennings, N.R., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. Auton. Agent. Multi-Ag. 1, 7–38 (1998)
3. Weiss, G. (ed.): Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge (1999)
4. Marik, V., McFarlane, D.: Industrial Adoption of Agent-Based Technologies. IEEE Intel. Syst. 20, 27–35 (2005)
5. Pechoucek, M., Marik, V.: Industrial deployment of multi-agent technologies: review and selected case studies. Auton. Agent. Multi-Ag. 17, 397–431 (2008)
6. Van Dyke Parunak, H.: A practitioners' review of industrial agent applications. Auton. Agent. Multi-Ag. 3, 389–407 (2000)

7. Choinski, D., Metzger, M., Nocon, W., Polakow, G.: Cooperative Validation in Distributed Control Systems Design. In: Luo, Y. (ed.) CDVE 2007. LNCS, vol. 4674, pp. 280–289. Springer, Heidelberg (2007)
8. Metzger, M., Polakow, G.: A Survey on Applications of Agent Technology in Industrial Process Control. IEEE T. Ind. Inform. 7, 570–581 (2011)
9. Nocon, W., Metzger, M.: Predictive Control of Decantation in Batch Sedimentation Process. AICHE J. 56, 3279–3283 (2010)
10. Chen, C.I., McDonald, K.A., Bisson, L.: Oscillatory behavior of Saccharomyces cerevisiae in continuous culture: I. Effects of pH and nitrogen levels. Biotechnol. Bioeng. 36, 19–27 (1990)
11. Balakrishnan, A., Yang, R.Y.K.: Self-forcing of a chemostat with self-sustained oscillations for productivity enhancement. Chem. Eng. Commun. 189, 1569–1585 (2002)
12. Dunn, I.J., Heinzle, E., Ingham, J., Prenosil, J.E.: Biological Reaction Engineering. In: Dynamic Modelling Fundamentals with Simulation Examples. Wiley-VCH Verlag (2003)
13. Skupin, P., Metzger, M.: Cooperative Operating Control for Induction or Elimination of Self-sustained Oscillations in CSTB. In: Luo, Y. (ed.) CDVE 2011. LNCS, vol. 6874, pp. 66–73. Springer, Heidelberg (2011)
14. Harrison, D.E.F., Topiwala, H.H.: Transient and oscillatory states of continuous culture. Adv. Biochem. Eng. 3, 167–219 (1974)
15. Metzger, M.: Fast-mode real-time simulator for the wastewater treatment process. Water Science and Technology 30, 191–197 (1994)
16. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Inform. Sciences 180, 2633–2634 (2010)
17. Corchado, E., Grana, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75, 61–63 (2012)
18. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72, 2729–2730 (2009)
19. Parulekar, S.J., Semones, G.B., Rolf, M.J., Lievense, J.C., Lim, H.C.: Induction and elimination of oscillations in continuous cultures of Saccharomyces Cerevisiae. Biotechnol. Bioeng. 28, 700–710 (1986)
20. Bader, F.G.: Analysis of double-substrate limited growth. Biotechnol. Bioeng. 20, 183–202 (1978)
21. Skupin, P.: Mathematical model for continuous culture under double substrate limitation with the specific growth rates proportional to the ratio of flow rates and exponential weights. Internal Report of the Institute of Automatic Control, 1–10 (2009)
22. Kompala, D.S., Ramkrishna, D., Jansen, N.B., Tsao, G.T.: Investigation of bacterial growth on mixed substrates. Experimental evaluation of cybernetic models. Biotechnol. Bioeng. 28, 1044–1056 (1986)
23. Skupin, P.: Simulation approach for detection of the self-sustained oscillations in continuous culture. In: Proceedings of the 11th WSEAS International Conference on Mathematics and Computers in Biology and Chemistry, pp. 80–85. Iasi (2010)

Agent Capability Taxonomy for Dynamic Environments

Jorge Agüero, Miguel Rebollo, Carlos Carrascosa, and Vicente Julián

Departamento de sistemas informáticos y computación
Universitat Politècnica de València
Camino de Vera S/N 46022 Valencia, Spain
{jaguero,mrebollo,carrasco,vinglada}@dsic.upv.es

Abstract. Currently there are many intelligent agent models in different design methodologies, from simple models of embedded agents to powerful and complex models for virtual organizations. However, although many of these agent models provide a number of components to solve different types of problems, few of them provides abstractions or concepts that allow to consider how to deal with dynamic environments at the design level, (an environment that changes itself in terms of resources available, global behavioural rules, etc.) In this work, we propose two abstractions that provide the developer with a new way of modeling reactive agent capabilities in dynamic environments. The first abstraction focuses on how to process the environmental stimuli as events, and the second abstraction specifies how to launch tasks in response to events, an approach that is based on event-condition-action rules.

Keywords: Capability Taxonomy, Agent Model, Model-Driven.

1 Introduction

The design of systems to solve problems in dynamic environments is a complex task. This is most evident when addressing real problems where the dynamic nature of the environment makes solutions that are appropriate at a given time and inappropriate for another instant in time. These systems must deal with an environment that changes itself in terms of its available resources, global behavioural rules, norms. The environment includes multiple actors and artifacts which may enter or leave frequently of the scene unexpectedly.

An example of such systems are *Pervasive Systems* or systems in *Ambient Intelligence* (AmI). *Pervasive Systems* is a paradigm with multiples scenarios, actors, devices and different state environments; where it is difficult to find a compact view of all components. The requirements of this kind of systems are very different[11]. Software engineering based on Multi-Agent Systems (MAS), has the capability to fulfill these requirements[5,8]. This approach supports the integration of highly heterogeneous components, where agents work together to support complex actions, in a collaborative and dynamic way[9,1], where the agent cognitive capabilities or intelligent capabilities (pro-activity, planning, inference, sociality) are used by solved the dynamic nature of the environment[8].

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 37–48, 2012.

However, most of the agents design methodologies do not include concepts or abstractions that deal with aspects of dynamic environments. In these methodologies, the environment usually is an abstraction where agents are located and interact with it. Our proposed offers tools to encapsulate the cognitive capabilities of the agent to face a changing environment, providing the developer (at design time) with abstractions to model interactions with dynamic environments. We propose to extend the generic agent-π [2] by adding a tasks and events models in order to increase the expressiveness of the model.

The first abstraction allows the agent to know how to process repetitive stimuli coming from the environment (how to handle events that cause changes in the environment). The taxonomy allows that the agent identify all the events, and the order in which events will be processed. The second abstraction allows the agent to decide how to launch actions or tasks in response to changes in the environment (in an event-condition-action model). With this proposal we want to provide the developer abstractions in order to have specialized agent responsiveness in dynamic environments. Model-Driven Development (MDD)[13] approach is used by developed these abstractions[12], so our agent model have an abstraction high level. The high-level abstractions provided are platform-independent, and the corresponding rules to transform them into the concepts related with different execution platforms (platform dependent) are provided.

The rest of the document is structured as follows. Section 2 describes the Agent Model to be extended with our proposal. Section 3 explains the different Task and Event taxonomies. These taxonomies generate the new Agent Capability model that is presented in section 4. Finally, the conclusions of this work are presented in section 5.

2 Agent-π: An Agent Model

MDD approach is based in the definition of a set of meta-models and the translations among them[13]. One fundamental challenge in the meta-model definition is to select which concepts or components will be included in order to model the system. To achieve this objective, some of the most well-known approaches in the area of MAS were studied: TROPOS[6], GAIA [15], OperA[10], INGENIAS[14] and AML[7]. The purpose of this analysis was to extract the common features from the methodologies studied and adapt them to the current proposal, specifying a generic platform-independent meta-model. This set of meta-models is called πVOM (*Platform-Independent Virtual Organization Model*)[5].

This set of meta-models is created by the detection of common concepts in an iterative cycle consisting of a bottom-up analysis. Common elements in existing MAS methodologies, have been identified and incorporated to Platform Independent Model (PIM). After that, it is possible to transform PIM models into Platform Specific Models (PSM). The main views/models of πVOM are the *Structure*, *Functionality*, *Normative*, *Agent*, and its *Environment*.

The *Agent* Model it is shown in Figure 1 and it provides an abstract view of its main components, concepts and the existing relationships. This model is

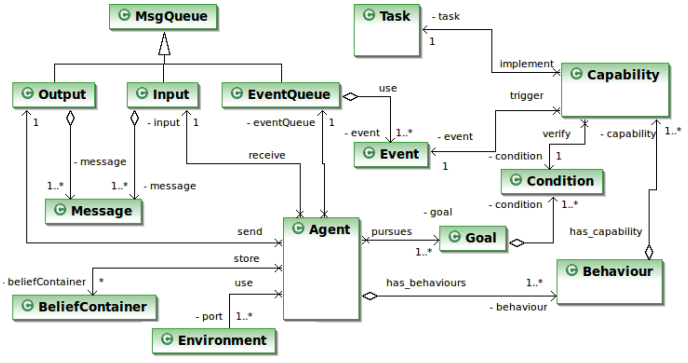


Fig. 1. Concepts used in the *agent-π* model

called *agent-π* (agent Platform-Independent). Furthermore, an automatic transformation rules have been obtained according to the MDD process. These rules allow the transform of the agent-π meta-model into specific models for different agent platforms, such as JADE-Leap and ANDROMEDA agent platforms (PSMs). The complete process has been applied the design of *Pervasive Systems* [3,4].

The Agent model is a set of interrelated components, each of which serves a specific function for the agent definition. The main components are: *Behaviours*, *Capabilities*, and *Tasks*. The highest-level entity to be considered is the agent. The organizations, group rules, or behaviour norms (other πVOM models) are not taken into account in this paper, due to space limitations.

Tasks represent the *know-how* of the *Agent* and they are the components where actions are implemented. *Capabilities* represent the different situations of the agent and control where *Tasks* are applied. *Capabilities* follow a pattern of *event-condition-action*. *Behaviours* are roles that group these capabilities.

The main reason for splitting the whole problem-solving method is to provide an abstraction that organizes the problem-solving knowledge in a modular and gradual way. The *Task* concept is the concept that incorporates the needed know-how that allows the agent to try to solve a problem. This concept is encapsulated in the meta-model in a *Capability*, which is an event-oriented component to express the circumstances under which a *Task* must be launched to execution.

A set of *Capabilities* can be encapsulated into a *Behaviour* that models the response of the agent to different situations. An agent state defines a situation (which is represented by the current *Beliefs* and *Goals*) that activates a *Behaviour* or allows it to go on being activated. Table 1 summarizes the main components and concepts employed in the Agent model.

3 Extending Task and Event Models

As stated above, we propose two abstractions that allows modeling a highly dynamic environments. These abstractions are implemented on the *agent-π* model (on the *Task* and *Event* meta-models), allowing that the *agent-π* architecture

Table 1. The main concepts used in the Agent model

agent-π concepts	Description
Agent	The *agent* is a rational and autonomous entity. The entity that usually is represented in MAS methodology.
Behaviour	It encapsulates a set of capabilities activated in specific circumstances; it represents the abstract concept of role.
Capability	It represents an event-driven approach to solve a specific problem.
Task	The know-how related to a specific problem.
Event	It is employed to activate capabilities inside the agent. Occurrence of something that changes the environment and/or agents.
BeliefContainer	An abstraction employed to represent the agent knowledge.
Goal	A specification of a state that the agents are trying to achieve.
Condition	A specification of a set of constraints.
MsgQueue	A specification of a collection of different messages.
Message	A mechanism employed for intercommunication among agents.
Environment	It is the way to model the external world

to be extended. This new architecture is focused on proposing a novel way to model the agent reaction capability. This new *agent-π* provides of a set of different skills (*capabilities*). This set of skills give the developer the ability to decide how to try to resolve a specific problem.

3.1 Task Taxonomy

A fundamental component of the agent model is the *Task*. *Tasks* are the elements that contain the code associated to the agent's Capabilities. A *Task* in execution belongs to only one *Capability*, and it may be seen as the agent's answer to a problem. However, it is the designer who must determine whether that problem must be solved only once or it must be solved as many times as it occurs. In accordance with to the way different instances can be activated, *Tasks* can be classified into the following types:

1. **Multiples:** Different instances of the same *Task* can be activated. For instance, if a *Task* has to answer a specific message by means of an ACK, the designer can decide that this answer be done in parallel to different messages. Thus, the designer would define the *task* in charge of answering these messages as multiple.
2. **Exclusive:** There exists only one instance of the *Task* at the same time.This kind of *Task* can be divided into two sub-kinds according to the way new activations of instances are dealt with when the event and the suitable condition are given:
   (a) **Non-interrupting:** The first instance is continued until it finishes, thus delaying the possible execution of new instances of the same *Task*.
   (b) **Interrupting:** The new instance eliminates the old one. For instance, if the capacity is to calculate a solution to a problem and the generation of a new instance indicates that the data being used in the old instance calculation are outdated, then the calculation that is being done is no longer useful.

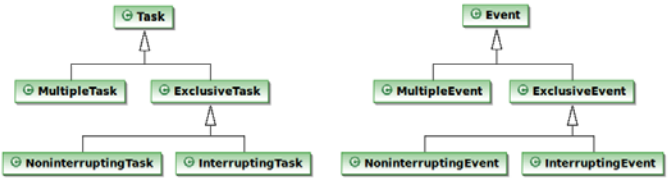


Fig. 2. New abstractions for tasks and events in the *agent-π* model

3.2 Event Taxonomy

An *Event* is any notification that received by the agent informing it that something that may be of interest has happened in the environment or inside the agent. This may cause the activation of a new *Capability*.

In a way similar to *Tasks*, one possible classification of events comes up when the subject is the management of new instances of the same event:

1. **Multiple:** There can be different instances of the same event in the queue, and all of them have to be managed.
2. **Exclusive:** There exists only one event instance waiting to be attended to. Depending on the way new event instances will be managed, events of this kind can be classified into two sub-types:
   (a) **Non-interrupting:** The new event instance is eliminated, only the first instance is generated.
   (b) **Interrupting:** The new event instance eliminates the old one, only the last generated event instance.

Now, events may be generated according to different subjects, such as: (i) Changes in beliefs: These changes may be a *new* belief or the *update* or *deletion* of an existing belief. (ii) Changes in goals: An event may be related to the creation of a *new* goal or to the *update* or *deletion* of an existing goal. (iii) New ACL message arrival. (iv) Changes in the resources.

Finally, our proposal extends the *agent-π* architecture. These changes allow the agent to better address changing environments. This is achieved through new abstractions: agent tasks and events, which are summarized in Figure 2.

4 Capability Taxonomy

We can now analyze in detail the different types of capabilities (resulting from the combination of different types of events and tasks) that are available to the developer to solve different problems. However, before describing the Capability Taxonomy, we will use an example to illustrate this taxonomy.

The example consists of a Multi-Agent System that manages and controls the processes of The *Smart Transport System*. The *Smart Transport System* is an application that facilitates the interconnection between passengers (citizens,

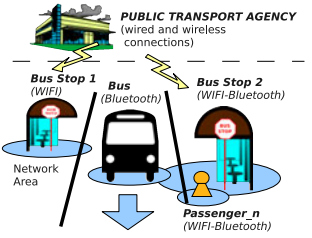


Fig. 3. (a) Summary structure of the Smart Transport System

tourists), Bus Stops and transport vehicles (Bus, Subways, Trains, Trams); delimiting services that each one can request or offer. The system controls which services must be provided by each agent. The system proposed provides wireless data services, which allow mobile devices (by example PDAs), cellular phones to communicate with different elements of the system that provide services.

The intelligent agent installed in the embedded device allows the providers to mutually discover each other. This interaction is illustrated (physical viewpoint) in Figure 3(a). The system can provide various types of services to facilitate and improve the quality of the public transport system. The *Smart Transport System* offers mobile services (by passengers and vehicles), for example:

- The passengers can receive/request information for possible tourist routes, personalized news services, transport ETA(Estimate Time Arrival), shorter routes to the destination, etc.
- The basic services of the Bus Stop Agent could be: (i) Display Information; (ii) Route Search; (iii) Transport Notification and ETA; and (iv) POI'S (Point Of Interest). The *Display Information* adjusts the information in the stop screens according to user profiles. Passengers can also download their favorite news sources to the mobile terminal (depending of their user profile).

To analyze the different features provided by our proposal, we assume that an agent A_i has various *Tasks*, *Capabilities*, and *Behaviours* and we define the following:

1. Let $C = \{C_1, C_2, \ldots, C_I\}$ be the set of all the agent capabilities, such that the i-nth capability is $C_i \in C | i = \{1, \ldots, I\} \wedge I \in \mathbb{N}$.
2. Let E be the set of all the events that handle the agent. These events are grouped into queues $E = \{E_1, E_2, \ldots, E_K\}$, where E_K is the event queue of the capability C_k, such that $E_K \in E | K = \{1, \ldots, I\} \wedge I \in \mathbb{N}$.
3. Let $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ be an event queue at the instant of time t, containing a series of events $e_k[t_i]$ produced in the time interval $[t_0, t_n]$, such that $e_k[t_i] \in E_K | (0 \leq i \leq n) \wedge (t_n \leq t_i) \wedge (t_n > t_0 > 0)$
4. Let $T = \{T_1, T_2, \ldots, T_J\}$ be the set of all the agent tasks, such that $T_j \in T | j = \{1, \ldots, I\} \wedge I \in \mathbb{N}$.
5. Let $T[t]$ be the set of agent active tasks at the instant t (with $T[t] \in T$).

6. Let $T_k[t]$ be an instance of the k-nth task that is launched at the instant t by the capability C_k in response at the event $e_k[t]$ of the queue $E_K[t]$ (with $T_k[t] \in T[t] \in T$).

Finally, as stated above, our agent model is a process of *Event-Condition-Action* (definition 6), that is managed by the *Capabilities*. However, this process *Event-Condition-Action* can be interpreted as a functional relationship, i.e., a *Capability* can be interpreted as a function $y = C_k(x)$ (a function that activates or initiates tasks). A function that takes the event $e_k[t]$ as input (argument x), and produces the task $T_k[t]$ (value y) as output. Thus, an active task force is launched as $T_k[t] = C_k(e_k[t])$, if the trigger condition is correct. Summarizing, by interpreting *Capability* as a function, this can be described as:

$$T_k[t] = C_k(e_k[t]) = \begin{cases} \emptyset & \text{if } event\ condition = false \\ T_k & \text{if } event\ condition = true \end{cases}$$

With these definitions, it is possible to analyze different combinations of the proposed abstractions that allow the developer to solve different problems in dynamic environments. The implementation of this proposal assumes that the agent life cycle is determined by a set of events with a frequency f_e, which are stored in its queue. These events are processed by the agent with a frequency or rate given by the scheduler (f_s) and a task is subsequently launched in response to this event (which has a duration τ). To appreciate the usefulness of this proposal assumes that $f_e \gg f_s \gg 1/\tau$, i.e., the speed with which the events arrived is very high (additional tasks may be atomic or may not). Therefore, the above description of the events and tasks taxonomy generates nine types of *Capabilities* that we described in the following subsections.

4.1 mmCapability: Multiple Events and Multiple Tasks

In this case, there is an event queue at the instant t, such that $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and take the first event in the queue (which is pending) $e_k[t] = first(E_K[t]) = e_k[t_0]$, which is verified by the capability C_k to activate the task $T_k[t] = C_k(e_k[t_0])$ for its launch. The task and event queues (in the agent) can be described as:

$$E_K[t+1] = E_K[t] - first(E_K[t])$$
$$T[t+1] = T[t] \cup \{T_k[t]\}$$

This scenario assumes that the agent reacts to all of the stimuli in the environment and, therefore, has all the events in memory (queued). In response to this monitoring the agent reacts by launching as many tasks as events processed. An illustration of this scenario implies that the agent has a complete monitoring of the environment and responds to all the dynamics or changes. In our example, the *Smart Transport System*, this capability allows the traffic status service (of the Central Agency) to be modelled. Each transport unit may request the traffic state and the information must be sent to each transport unit.

4.2 imCapability: Interrupting Events and Multiple Tasks

In this case, it is assumed that there is an event queue $E_K[t] = \{e_k[t_n]\}$ that maintains only one significant event -the last one-. However, we can assume that there is a real event queue at the instant t, such as $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and to model interrupting events, it only needs to process the last event (and only one event) $e_k[t] = last(E_K[t]) = e_k[t_n]$, and then proceed to empty the queue of events that remain or are stored $E_K[t] = \emptyset$. Thus, once the relevant event is received, it is verified by capacity C_k activate the task $T_k[t] = C_k(e_k[t_n])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - last(E_K[t]) \Rightarrow E_K[t+1] = \emptyset$$
$$T[t+1] = T[t] \cup \{T_k[t]\}$$

In this scenario, the agent has good reactivity, and launch new tasks at relevant events. However, it has limited monitoring capability (the last event is considers as relevant). In our example, the *Smart Transport System*, this capability allows the Breaking News service (of the Central Agency) to be modelled. This service sends the most relevant news of the transport system and the city to the user.

4.3 nmCapability: Non-interrupting Events and Multiple Tasks

In this case, it is assumed that there is an event queue $E_K[t] = \{e_k[t_0]\}$ that maintains only one significant event -the first one-. However, we can assume that there is a real event queue at the instant t, such as $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and to model non-interrupting events, it must only take the first event (and only one event) $e_k[t] = first(E_K[t]) = e_k[t_0]$, and then proceed to empty the queue of events that remain or are stored $E_K[t] = \emptyset$. Thus, once the relevant event is received, it is verified by capacity C_k to activate the task $T_k[t] = C_k(e_k[t_0])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - first(E_K[t]) \Rightarrow E_K[t+1] = \emptyset$$
$$T[t+1] = T[t] \cup \{T_k[t]\}$$

In this scenario, the agent has good reactivity and launch new tasks in response to relevant events. However, it has limited monitoring capability (the first event is considers as relevant). In our example, the *Smart Transport System*, this capability allows the Checking service in Bus Stop or Transport Unit to be modelled. This service checks the user presence in the Transport Unit or Bus stop and only processes the first request for all the users in the domain.

4.4 miCapability: Multiple Events and Interrupting Tasks

In this case, there is an event queue at the instant t, such that $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and take the first event in the queue

(which is pending) $e_k[t] = first(E_K[t]) = e_k[t_0]$, which is verified by the capability C_k to activate the task $T_k[t] = C_k(e_k[t_0])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - first(E_K[t])$$
$$T[t+1] = \begin{cases} (T[t] - \{T_k\}) \cup \{T_k[t]\} & \text{if } T[t] \cap \{T_k[t]\} \neq \emptyset \\ T[t] \cup \{T_k[t]\} & \text{otherwise} \end{cases}$$

This scenario assumes that the agent monitors all of the events, but in response keeps only one task (of a specific type) running. This scenario is quite reactive, but it is costly since the agent must safely stop when the task is interrupted. In our example, the *Smart Transport System*, this capability allows the temperature service in Bus unit to be modelled. The service adjusts the temperature of the air conditioner at every stop, looking for the average temperature from the user profiles.

4.5 iiCapability: Interrupting Events and Interrupting Tasks

In this case, it is assumed that there is an event queue $E_K[t] = \{e_k[t_n]\}$ that maintains only one significant event -the last one-. However, we can assume that there is a real event queue at the instant t, such as $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and to modeling interrupting events, it must only process the last event (and only one event) $e_k[t] = last(E_K[t]) = e_k[t_n]$ and then proceed to empty the queue of events that remain or are stored $E_K[t] = \emptyset$. Thus, once the relevant event is received, it is verified by capacity C_k to activate the task $T_k[t] = C_k(e_k[t_n])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - last(E_K[t]) \Rightarrow E_K[t+1] = \emptyset$$
$$T[t+1] = \begin{cases} (T[t] - \{T_k\}) \cup \{T_k[t]\} & \text{if } T[t] \cap \{T_k[t]\} \neq \emptyset \\ T[t] \cup \{T_k[t]\} & \text{otherwise} \end{cases}$$

In this scenario, the agent has good reactivity, although it has limited memory capability (of events), only store one event (the last one). The new event arrival, stops the task for this type of event, and launches a new task. A new task that best matches or adjusts the new event received. In our example, the *Smart Transport System*, this capability allows the POI'S service to be modelled. This is a service that informs the user of the points of interest (e.g. tourist sites, restaurants) near each stop or station. This service tells to the mobile device the direction to follow (path) to reach the point of interest using the last events sent by the bus stop.

4.6 niCapability: Non-interrupting Events and Interrupting Tasks

In this case, it is assumed that there is an event queue $E_K[t] = \{e_k[t_0]\}$ that maintains only one significant event -the first one-. However, we can assume that

there is a real event queue at the instant t, such as $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and, to model non-interrupting events, it must only process the first event (and only one event) $e_k[t] = first(E_K[t]) = e_k[t_0]$ and then proceed to empty the queue of events that remain or are stored $E_K[t] = \emptyset$. Thus, once the relevant event is received, it is verified by capacity C_k to activate the task $T_k[t] = C_k(e_k[t_0])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - first(E_K[t]) \Rightarrow E_K[t+1] = \emptyset$$
$$T[t+1] = \begin{cases} (T[t] - \{T_k\}) \cup \{T_k[t]\} & \text{if } T[t] \cap \{T_k[t]\} \neq \emptyset \\ T[t] \cup \{T_k[t]\} & \text{otherwise} \end{cases}$$

In this scenario, the agent has good reactivity, although it has limited memory capability (the first event is relevant). This capability allows to stop the previous task (that no longer responds to the event), and launched a new task.

4.7 mnCapability: Multiple Events and Non-interrupting Tasks

In this case, there is an event queue at the instant t, such that $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and take the first event in the queue (which is pending) $e_k[t] = first(E_K[t]) = e_k[t_0]$, which is verified by the capability C_k to activate the task $T_k[t] = C_k(e_k[t_0])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - first(E_K[t])$$
$$T[t+1] = T[t] \cup \begin{cases} \emptyset & \text{if } T[t] \cap \{T_k[t]\} \neq \emptyset \\ \{T_k[t]\} & \text{otherwise} \end{cases}$$

In this scenario, the agent has limited resources, there is only one task running (of a specific task type) and the agent needs to complete monitoring capability. Therefore, it has all the events stored in the queue. In our example, the *Smart Transport System*, this capability allows the ETA service at the bus stops to be modelled. The transport units to send a event of location to bus stop, and bus stop shows on their screens the ETA information of each unit (as a queue of transport units).

4.8 inCapability: Interrupting Events and Non-interrupting Tasks

In this case, it is assumed that there is an event queue $E_K[t] = \{e_k[t_n]\}$ that maintains only one significant event -the last one-. However, we can assume that there is a real event queue at the instant t, such as $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and to modeling interrupting events, it must only process the last event (and only one event) $e_k[t] = last(E_K[t]) = e_k[t_n]$ and then proceed to empty the queue of events that remain or are stored $E_K[t] = \emptyset$. Thus, once the relevant event is received, it is verified by capacity C_k to activate the task $T_k[t] = C_k(e_k[t_n])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - last(E_K[t]) \Rightarrow E_K[t+1] = \emptyset$$
$$T[t+1] = T[t] \cup \begin{cases} \emptyset & \text{if } T[t] \cap \{T_k[t]\} \neq \emptyset \\ \{T_k[t]\} & \text{otherwise} \end{cases}$$

In this scenario, the agent has limited resources there is only one task running (of a specific task type), and also a limited monitoring capability (the last event is considers as relevant). In our example, the *Smart Transport System*, this capability allows the Location Service in Transport Unit to be modelled. This service shows the location in the route map on the unit screen (this Capability is computationally expensive and only uses the latest data of the beacon).

4.9 nnCapability: Non-interrupting Events and Non-interrupting Tasks

In this case, it is assumed that there is an event queue $E_K[t] = \{e_k[t_0]\}$ that maintains only one significant event -the first one-. However, we can assume that there is a real event queue at the instant t, such as $E_K[t] = \{e_k[t_0], \ldots, e_k[t_n]\}$ (with $t_n > t_0$), and to modeling non-interrupting events, it must only process the first event (and only one event) $e_k[t] = first(E_K[t]) = e_k[t_0]$ and then proceed to empty the queue of events that remain or are stored $E_K[t] = \emptyset$. Thus, once the relevant event is received, it is verified by capacity C_k to activate the task $T_k[t] = C_k(e_k[t_0])$ for its launch. The task and event queues can be described as:

$$E_K[t+1] = E_K[t] - first(E_K[t]) \Rightarrow E_K[t+1] = \emptyset$$
$$T[t+1] = T[t] \cup \begin{cases} \emptyset & \text{if } T[t] \cap \{T_k[t]\} \neq \emptyset \\ \{T_k[t]\} & \text{otherwise} \end{cases}$$

In this scenario, the agent has limited resources there is only one task running (of a specific task type), and also a limited monitoring capability (the last event is considers as relevant). In our example, the *Smart Transport System*, this capability allows the Route Search service in Bus Stop to be modelled. This service calculates a transportation route for the user since the stop is the starting point to a destination, indicating the possible transfer that the user needs.

5 Conclusions

This work presents a *Capability* Taxonomy that provides the developer with a new way of modeling agent reactive capabilities in dynamic environments.

Thus, with this *Capability* Taxonomy, the changes in the environment can not only be solved using the cognitive abilities of the agent, but also using the two abstractions (events and tasks) that provide agents with different capabilities of reaction. The agent knows how to process repetitive stimuli coming from the environment and can decide how to launch actions to the environment in response to changes in the environment (in an event-condition-action model). This process was illustrated with *Smart Transport System* managed by a MAS.

Future work in this research area will focus on developing explicit support for Context-Awareness specification, using ontologies to describe the contextual information, allowing that new knowledge of the environment can to be inferred.

Acknowledgments. This work was partially supported by TIN2009-13839-C03-01 and PROMETEO/2008/051 projects of the Spanish government and CONSOLIDER-INGENIO 2010 under grant CSD2007-00022.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Does Android Dream with Intelligent Agents? In: Proceedings of DCAI 2008, vol. 50, pp. 194–204 (2008)
3. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Agent design using Model Driven Development. In: Proceedings of PAAMS 2009, vol. 55, pp. 60–69 (2009)
4. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: MDD-based agent-oriented software engineering for ubiquitous deployment. In: Proceedings of MobiQuitous 2009, pp. 1–2. IEEE press (2009)
5. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Developing Pervasive Systems as Service-oriented Multi-Agent Systems. In: Proceedings of MobiQuitous 2010, pp. 1–12. CD press (2010)
6. Castro, J., Kolp, M., Mylopoulos, J.: A Requirements-Driven Development Methodology. In: Dittrich, K.R., Geppert, A., Norrie, M. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 108–123. Springer, Heidelberg (2001)
7. Cervenka, R., Trencansky, I.: The Agent Modeling Language – AML. Whitestein Series in Software Agent Technologies and Autonomic Computing (2007)
8. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Inf. Sci. 180(14), 2633–2634 (2010)
9. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
10. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. Phd dissertation, Utrecht University (2003)
11. Endres, C., Butz, A., MacWilliams, A.: A survey of software infrastructures and frameworks for ubiquitous computing. Mobile Infor. Syst. 1(1), 41–80 (2005)
12. Hahn, C., Madrigal-Mora, C., Fischer, K.: A platform-independent metamodel for multiagent systems. In: AAMAS 2008, vol. 18(2), pp. 239–266 (2008)
13. (OMG): Object management group. MDA guide version 1.0.1 (June 2003), http://www.omg.org/docs/omg/03-06-01.pdf
14. Pavón, J., Gómez-Sanz, J.: Agent Oriented Software Engineering with INGENIAS. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS (LNAI), vol. 2691, pp. 394–403. Springer, Heidelberg (2003)
15. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multiagent systems: The GAIA methodology. ACM Trans. Softw. Eng. Methodol. 12(3), 317–370 (2003)

Modeling Internet as a User-Adapted Speech Service

David Griol, Javier Carbó, and José Manuel Molina

Applied Artificial Intelligence Group
Computer Science Department
Universidad Carlos III de Madrid
28911 - Leganés, Spain
{david.griol,javier.carbo,josemanuel.molina}@uc3m.es

Abstract. The web has become the largest repository of multimedia information and its convergence with telecommunications is now bringing the benefits of web technology and hybrid artificial intelligence systems to hand-held devices. However, maximizing accessibility is not always the main objective in the design of web applications, specially if it is concerned with facilitating access for disabled people. This way, natural spoken conversation and multimodal conversational agents have been proposed as a solution to facilitate a more natural interaction with these kind of devices. In this paper, we describe a proposal to provide spoken access to Internet information that is valid not only to generate basic applications (e.g., web search engines), but also to develop dialog-based speech interfaces that facilitate a user-adapted access that enhances web services. We describe our proposal and detail several applications developed to provide evidences about the benefits of introducing speech to make the enormous web content accessible to all mobile phone users.

Keywords: Conversational Agents, Multimodality, Internet Modeling, VoiceXML, XHTML+Voice, Speech Interaction, Neural Networks.

1 Introduction

Continuous advances in the development of information technologies and the miniaturization of devices have made it possible to access information, web services, and artificial intelligence systems from anywhere, at anytime and almost instantaneously through wireless connections [2]. Although devices such as PDAs and smartphones are widely used today to access the web, contents are usually accessible only through web browsers, which are operated by means of traditional graphical user interfaces (GUIs). The reduced size of the screen and keyboards makes the use of these devices very difficult and also avoids the use of these applications by motor-handicapped and visually impaired users. The major drawback of the existing web infrastructure is that, the present web content was originally designed for traditional desktop browsers. This way, although mobile phones are designed to provide anytime and anywhere access to users, the challenge that is

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 49–60, 2012.

presented to the present Internet world is to make the enormous web content accessible to all mobile phone users and by means of a more natural communication with the user.

Multimodal interfaces go a step beyond GUIs by adding the possibility to communicate with these devices through other interaction modes such as speech. Multimodal conversational agents [8,7] can be defined as computer programs designed to emulate communication capabilities of a human being including several communication modalities. The use of these agents provides three main benefits. Firstly, they facilitate a more natural human-computer interaction, as it is carried out by means of a conversation in natural language. Secondly, multimodal interfaces make possible the use of these applications in environments in which the use of GUI interfaces is not effective, for example, in-car environments. Finally, these systems provides the objective of facilitating the access to the web for people with visual or motor disabilities, allowing their integration and the elimination of barriers to Internet access [1].

Most of multimodal conversational agents to access web contents and services applications are currently developed using the VoiceXML language[1], given that it has been defined as the World Wide Web Consortium (W3C) standard to access Internet contents by means of speech. VoiceXML audio dialogs feature synthesized speech, digitized audio, recognition of spoken and DTMF key input (Dual-tone multi-frequency signaling), recording of spoken input, telephony, and mixed initiative conversations. The standard also enables the integration of voice services with data services using the client-server paradigm. Therefore, the VoiceXML standard facilitates the access to the net in new devices and environments by providing all these functionalities in combination with well-defined semantics (thus making XML documents universally accessible).

However, this programming language only allows the definition of a dialog strategy based on scripted Finite State Machines. This way, VoiceXML systems usually emphasize on the search of web documents in specific tasks and not on the interaction with the user. With the aim of creating dynamic and adapted dialogs, as an alternative of the previously described rule-based approaches, the application of soft computing models and statistical approaches to dialog management makes it possible to consider a wider space of dialog strategies [11,5]. The main reason is that these models can be trained from real dialogs, modeling the variability in user behaviors.

In this paper we describe a proposal to model the web as an speech-based service by means of the combination of the VoiceXML standard and statistical methodologies for dialog management. This makes possible to generate not only general-purpose applications (e.g., speech-based access to web search engines or extended use applications like the Wikipedia), but also to develop enhanced speech-based interfaces that provide personalized access to web services in which dialog is required to iteratively exchange information and achieve the objectives (e.g., ask the user about different information items in order to provide them specific information related to travel planning, hotel booking, etc).

[1] http://www.w3.org/TR/voicexml20/

For the former applications we propose the use of the XHTML+Voice language[2] in combination with a specific strategy to dynamically create the different grammars in the application. This language combines the visual modality offered by the XHTML language and the functionalities offered by the VoiceXML language for the interaction by means of speech. For the latter applications we propose the use of a statistical dialog management methodology in combination with a user simulation technique. This combination makes possible to automatically learn a statistical dialog model to select systems responses adapted to each user and also include grammars which facilitate the interaction in natural language.

We have applied our proposal to develop several conversational agents which interact with different web-based applications, providing a sophisticated interface which merges voice with traditional GUIs. All these applications are easily interoperable so that they are very useful to evaluate the potential of voice interaction in general and specific domains, through a variety of resources and with different users. This way, one of the main objectives of our work is to adequately convey to users the logical structure and semantics of content in web documents.

The remainder of the paper is as follows. Section 2 describes our proposal to include speech to facilitate the information and services on the Internet. Section 3 describes the application of our proposal to develop several multimodal conversational agents. This section also summarizes the results of a preliminary evaluation of these agents. Finally, Section 4 provides some conclusions and future research work.

2 Our Proposal to Provide a Speech-Based Access to the Web

HTML is the language popularly used for marking up information on the web so that it can be displayed using commercially available browsers. However, the eXtensible Markup Language (XML) was developed as a solution to correct the limitation of the use of information that is available on the web by marking-up information in the web pages and also allowing developers to define their own tags with well-defined meanings that others can understand. The use of an XML-based language significantly improves the portability and interoperability of the programmable parts (including data and programs) of the service. One of the main objectives of XML and ontology-based languages is to adequately convey to users the logical structure and semantics of content in web documents.

Several proposals have been developed to translate HTML pages to speech [3]. These systems captured the text present in the web page and employed speech synthesis technologies to speak this information to the user, also introducing different sounds and tones to visually impaired can make-out the structure of the document. However, these systems capture only specific parts of the HTML code and fails to address the interactive features provided by the

[2] http://www.w3.org/TR/xhtml+voice/

HTML pages. Although additional proposals have been developed to translate from HTML or XML web pages to VoiceXML [10,4], they require a previous preprocessing of the code by means of the user or they only handle a subset of HTML tags. In addition, the XML language by definition and the extended use of Cascading Style Sheets (CSS) to describe the presentation semantics make a general definition of these translators almost impossible. For these reasons, we propose a specific translation from HTML pages to XHTML+Voice only to develop general-purpose multimodal applications (Section 2.1) and the use of a user-adapted statistical dialog management methodology to develop dialog-based applications which are focused on interactive dialog with users (Section 2.2). A number of currently extended speech-based applications with a general purpose (like the API proposal from google) are based on additional languages and technologies mainly focused on the development of general-purpose systems.

2.1 Developing General-Purpose Web Applications

As stated in the introduction, our proposal to generate speech-based interfaces for general-purpose web applications is based on the use of the XHTML+Voice language, thus combining speech access with visual interaction. Figure 1 shows the translation between a HTML document and its equivalent XHTML+Voice file. As it can be observed, the development of oral interfaces implemented by means of XHTML+Voice implies the definition of grammars, which delimit the speech communication with the system. The <*grammar*> element is used to provide a speech or DTMF grammar that specifies a set of utterances that a user may speak to perform an action or supply information, and for a matching utterance, returns a corresponding semantic interpretation.

We have defined a specific strategy to cover the widest range of search criteria by means of the definition of speech recognition grammars in the different applications. This strategy is based on different aspects such as the dynamic generation of the grammars built from the results generated by the interaction with a specific application (e.g., to include the results of the search of a topic using a speech-based search engine), the definition of grammars that includes complete sentences to support the naturalness of the interaction with the system (e.g., to facilitate a more natural communication and cover more functionalities), and the use of the ICAO phonetic alphabet[3] in the cases in which spelling of the words is required in order not to restrict the contents of the search or in situations in which repetitive recognition errors are detected (e.g., in order not to delimit the topics to search using a search engine).

2.2 From General-Purpose to More Natural Mixed-Initiative Dialogs

When designing dialog-based conversational agents, developers need to specify the actions a system should take in response to user speech input and the state

[3] International Civil Aviation Organization (ICAO) phonetic alphabet: http://www.icao.int/icao/en/trivia/alphabet.htm

```
% HTML document
<html>
<head>
<title>VoiceApp-Voice Browser</title>
</head>
<body>

<li>LINK 1:
<a href="http://www.beatles.com/">
<b>The Beatles</b> Find out all about
The Beatles...</li>

...

<li> LINK 10:
<a href="http://www.rarebeatles.com/">
<b>Songs, Pictures, and Stories of
The Beatles</b>
Beatles website for collectors
and fans ...</li>

</body>
</html>
```

```
% XHTML+Voice file
<?xml version="1.0" encoding="ISO-8859-1"?>
<html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:vxml="http://www.w3.org/2001/vxml"
 xmlns:ev="http://www.w3.org/2001/xml-events"
 xmlns:xv="http://www.voicexml.org/2002/xhtml+voice">

<head>
<title>VoiceApp - Voice Browser</title>
<vxml:form id="nav">
    <vxml:block>
    To visit the links, you have to say
    "LINK" and thecorresponding number.
    </vxml:block>
    <vxml:field xv:id="app" name="app">
      <vxml:grammar src="inig.jsgf"/>
      <vxml:nomatch>
        <vxml:prompt>
        Please repeat again, I can not understand you.
        </vxml:prompt>
      </vxml:nomatch>
    </vxml:field>
    <vxml:filled mode="all">
      <vxml:prompt> Ok got them. </vxml:prompt>
      <vxml:elseif cond="app == 'home'"/>
        <assign name="window.location" expr="index"/>
      <vxml:elseif cond="app == 'link 1'"/>
        <assign name="window.location" expr="x1x"/>
     ...
      <vxml:elseif cond="app == 'link 10'"/>
        <assign name="window.location" expr="x10x"/>
      </vxml:if>
    </vxml:filled>
  </vxml:form>
<script src="java.js" type="text/javascript"></script>
</head>

<body id="docBody" ev:event="load" ev:handler="#nav">
<div id="cont" ev:event="click" ev:handler="#nav">
<h1>Results for: The Beatles</h1>

<li>LINK 1:<a href="http://www.beatles.com/">
<b>The Beatles</b> Find out all about
The Beatles...</li>

...

<li> LINK 10: <a href="http://www.rarebeatles.com/">
<b>Songs, Pictures, and Stories of The Beatles</b>
Beatles website for collectors and fans ...</li>

</body></html>
```

Fig. 1. Translation of a HTML document into an equivalent XHTML+Voice file

of the environment based on observed or inferred events, states, and beliefs. In addition, the conversational agent requires a dialog strategy that defines the conversational behavior of the system. Thus, a great effort is employed to empirically design dialog strategies, as the design of a good strategy is far from trivial since there is no clear definition of what constitutes a good strategy [11]. Additionally, speech recognition grammars for conversational agents have been usually build on the basis of handcrafted rules which are tested recursively,

which in complex applications constitutes a costly process [8]. As an alternative of the previously described rule-based approaches, the application of statistical approaches to dialog management makes it possible to consider a wider space of dialog strategies. The main reason is that statistical models can be trained from real dialogs, modeling the variability in user behaviors [11,5].

We propose to merge statistical approaches with VoiceXML in order to benefit from the flexibility of statistical dialog management and the facilities that VoiceXML offers. Our technique employs a statistical dialog manager that takes into account the history of the dialog until the current moment in order to decide the next system prompt, whereas the system prompts and the grammars which indicate the possible user responses to them are implemented in VoiceXML [5]. This technique is based on the definition of a data structure that we call Dialog Register (DR), and contains the information provided by the user throughout the previous history of the dialog. For each time i, the selection of the next system prompt A_i is carried out by means of the following maximization:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | DR_{i-1}, S_{i-1})$$

where set $\mathcal{A}$ contains all the possible system answers and S_{i-1} is the state of the dialog sequence (*system-turn, user-turn*) at time i.

Each user turn supplies the system with information about the task; that is, he/she asks for a specific concept and/or provides specific values for certain attributes. However, a user turn could also provide other kinds of information, such as task-independent information. This is the case of turns corresponding to *Affirmation*, *Negation* and *Not-Understood* dialog acts. This kind of information implies some decisions which are different from simply updating the DR_{i-1}. For that reason, for the selection of the best system answer A_i, we take into account the DR that results from turn 1 to turn $i-1$, and we explicitly consider the last state S_{i-1}.

The selection of the system answer is then carried out through a classification process, for which a soft-computing methodology based on multilayer perceptrons (MLP) is proposed. The input layer receives the codification of the pair (DR_{i-1}, S_{i-1}). The output generated by the MLP can be seen as the probability of selecting each of the different system answers defined for a specific task.

A corpus of dialogs for the specific task is required to learn the dialog model. Our approach for automatically acquiring a dialog corpus is based on the interaction of a user agent simulator and a conversational agent simulator [6]. In our dialog simulation technique, both agents use a random selection of one of the possible responses defined for the semantics of the task (expressed in terms of user and system dialog acts). At the beginning of the simulation, the set of system responses is defined as equiprobable. When a successful dialog is simulated, the probabilities of the answers selected by the the conversational agent simulator during that dialog are incremented before beginning a new simulation.

Errors and confidence scores are simulated by a specific module in the simulator using a model for introducing errors based on the method presented in [9]. The generation of confidence scores is carried out separately from the model employed for error generation. This model is represented as a communication

channel by means of a generative probabilistic model $P(c, a_u|\tilde{a}_u)$, where a_u is the true incoming user dialog act $\tilde{a}_u$ is the recognized hypothesis, and c is the confidence score associated with this hypothesis. The probability $P(\tilde{a}_u|a_u)$ is obtained by Maximum-Likelihood using the initial labeled corpus acquired with real users and considers the recognized sequence of words w_u and the actual sequence uttered by the user $\tilde{w}_u$.

$$P(\tilde{a}_u|a_u) = \sum_{\tilde{w}_u} P(a_u|\tilde{w}_u) \sum_{w_u} P(\tilde{w}_u|w_u)P(w_u|a_u)$$

Confidence score generation is carried out by approximating $P(c|\tilde{a}_u, a_u)$ assuming that there are two distributions for c.

$$P(c|a_w, \tilde{a}_u) = \begin{cases} P_{corr}(c) & if \quad \tilde{a}_u = a_u \\ P_{incorr}(c) & if \quad \tilde{a}_u \neq a_u \end{cases}$$

3 Practical Applications

To test our proposal, we have developed several applications corresponding to both general-purpose speech interfaces and dialog-based interactive conversational agents. In order to provide web-content using speech, an interactive voice response (IVR) platform is required. We have selected the Prophecy IVR Platform[4]. This IVR can interpret the VoiceXML language and act like a client to the web-servers. This way, it can translate an incoming request to a URL, fetch the document, interpret it and return the output to a mobile client. The Prophecy Media Resource Control Protocol (MRCP) media server has been used for prompting, recording, automatic speech recognition, text-to-speech generation, and conferencing functionalities.

3.1 General-Purpose Applications

Regarding general-purpose applications we have developed *Voice Dictionary* and *Voice Browser*. Both applications consists of a set of X+V documents. Some of them are stored from the beginning in the server of the application, while others are dynamically generated using PHP and JavaScript. This dynamic generation takes into account the information extracted from different web servers and MySQL databases in the system, and a set of users preferences and characteristics (e.g., sex, preferred language for the interaction, number of previous interactions with the system, and preferred application).

The *Voice Browser* application has been developed with the main objective of allowing speech access to both the search and presentation of the results in the interaction with the Google search engine. The application interface receives the contents provided by the user and displays the results both visually and using synthesized speech. This application also allows the multimodal selection of any of the links included in the result of the search by numbering them and allowing

[4] http://www.voxeo.com/products/voicexml-ivr-platform.jsp

using their titles as voice commands. Detailed instructions, help messages and menus have been also incorporated to facilitate the interaction.

The *Voice Dictionary* application offers a single environment where users can search contents in the Wikipedia encyclopedia with the main feature that the access to the application and the results provided by the search are entirely facilitated to the user either through visual modalities or by means of speech. Once the result of an initial search is displayed on the screen and communicated to the user by means of speech, they can easily access any of the links included in the result of the search or visit the rest of applications in the system with the possibility of interrupting the system's speech in any case. This functionality is achieved by means of the dynamic generation of the corresponding grammars, in which the different links that are present in the result of a specific search are included in the dynamic XHTML+Voice page automatically generated by means of a PHP script that captures the different information sources to inform the user about them (headings, text, contents, formulas, links, etc.).

A number of tests and verifications have been carried out to maximize the functionalities and accessibility of these two applications. These tests have been very important to detect and correct programming errors and accessibility problems. In addition, we have completed a preliminary assessment by means of a questionnaire to measure users subjective opinion about the system. The questionnaire contained five questions: i) Q1: *Did the system correctly understand you during the interaction?*; ii) Q2: *Did you understand correctly the messages of the system?*; iii) Q3: *Was it simple to obtain the requested information? / Was it simple to play the game?*; iv) Q4: *Do you think that the interaction rate was adequate?*, v) Q5: *Was it easy to correct mistakes made by the system?*; vi) Q6: *In general terms, are you satisfied with the performance of the system?* The possible answers to the complete set questions were the same: *Never*, *Rarely*, *Sometimes Usually* and *Always*. A numerical value between one and five was assigned for each answer (in the same order as they are shown in the questionnaire). Table 1 shows the average, maximum and minimum values obtained from the results provided by a total of 35 students and professors of our University using the different modules of the system without predefined scenarios.

The results of the preliminary evaluation of both applications show that the users who participated in the assessment positively evaluate the facility of obtaining the requested information by interacting with the system, the appropriate interaction rate during the dialog, and overall operation of the different applications in the system. The main problems mentioned by the users include the need

Table 1. Results of the preliminary evaluation of the *Voice Dictionary* and *Voice Search Engine* applications (1=minimal value, 5=maximum value)

	Q1	Q2	Q3	Q4	Q5	Q6
Average value	3.6	3.8	3.2	3.7	3.2	4.3
Maximum value	4	5	5	4	4	5
Minimal value	2	3	2	3	2	3

of improving the word error rate and achieve a better clarification of the action expected by the system at each moment of interaction. In addition, the 97% of the interactions finished achieving the objective(s) expected by the user, only the 4% of the systems turns corresponded to reprompts and the 12% to system confirmations. The error correction rate (computer as the average number of corrected errors per dialog divided by the number of corrected and uncorrected errors) was 91%.

3.2 Web Applications Based on Interactive Dialog

To test our proposal with a conversational agent focused on an interactive dialog with users, we have used the definitions taken to develop the EDECAN dialog system, which was developed in a previous study to provide information about train services, schedules and fares in Spanish [5]. The developed conversational agent generates a total of 51 different prompts.

A total of 100,000 dialogs was simulated using our user simulation technique and a set of scenarios covering the different queries for the system [6]. Then, the acquired dialogs were employed to automatically generate VoiceXML code for each system prompt and the grammar needed to correctly recognize the possible user responses. The 51 different system prompts have been automatically generated in VoiceXML using the proposed technique. For example, Figure 2 shows the VXML document to prompt the user for the origin city and the obtained grammar for ASR.

The *DR* defined for the system is a sequence of 15 fields, corresponding to the five possible queries that users can carry out to the system (*Hour, Price, Train-Type, Trip-Time, Services*) and the ten attributes that they can provide to complete these queries (*Origin, Destination, Departure-Date, Arrival-Date, Departure-Hour, Arrival-Hour, Class, Train-Type, Order-Number, Services*). This way, every dialog begins with a dialog register in which every value is equal to 0 and the greeting turn of the system, as it is showed following.

. . . . . .
S_1: *Welcome to the railway information system. How can I help you?*
A_1: (*Opening*:*Nil*:*Nil*)
DR_0: 00000-1000001000
. . . . . .

Each time the user provides information, this is used to update the previous *DR* and to obtain the new one. For instance, given a user turn providing the origin city, the destination city and the date, the new dialog register could be as follows.

. . . . . .
U_1: *I want to know timetables from Valencia to Madrid.*
Task Dependent Information: (*Hour*) [0.7] *Origin*:Valencia [0.2] *Destination*:Madrid [0.9]
Task Independent Information: None
DR_1: 10000-2100000000
. . . . . .

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
   xmlns:xsi="http://www.w3.org/2001/
   XMLSchema-instance"
   xsi:schemaLocation="www.w3.org/2001/vxml
   http://www.w3.org/TR/voicexml20/vxml.xsd"
   version="2.0" application="trains.vxml">
 <form id="origin_form">
  <field name="origin">
     <grammar type="application/srgs+xml"
           src="/grammars/origin.grxml"/>
     <prompt>Tell me the origin city.</prompt>
     <filled>
       <return namelist="origin"/>
     </filled>
  </field>
 </form>
</vxml>
```

```
#JSGF V1.0;
grammar origin;
public <origin> = [<desire>]
[<travel> <city> {this.destination=$city}]
[<proceed> <city> {this.origin=$city}];
<desire> = I want [to know] | I would like
[to know] | I would like | I want | I need
| I have to;
<travel> = go to | travel to | to go to
| to travel to;
<city> = Murcia | Vigo | Sevilla | Huelva |
Cuenca | Lugo | Granada | Salamanca |
Valencia | Alicante | Albacete | Barcelona
| Madrid;
<proceed> = from | going from | go from;
```

Fig. 2. VoiceXML document to require the origin city (left) and grammar to capture the associated value (right)

In this case, the confidence score assigned to the attribute *Origin* (showed between brackets in the previous example) is very low. Then, a "2" value is added in the corresponding position of the DR_1. The concept (*Hour*) and the attribute *Destination* are recognized with a high confidence score, adding a "1" value in the corresponding positions of the DR_1. Then, the input of the MLP is generated using DR_1, the codification of the labeling of the last system turn (A_1), and the task-independent information provided in the last user turn (none in this case). The output selected for the MLP would consist in the case of requiring the departure date. This process is repeated to predict the next system response afterwards each user turn.

A total of 25 dialogs was recorded from interactions of six students and professors of our University employing the conversational agent developed for the task following our proposal. We considered the following measures for the evaluation: i) Dialog success rate (*%success*). This is the percentage of successfully completed tasks; ii) Average number of turns per dialog (nT); iii) Confirmation rate (*%confirm*). It was computed as the ratio between the number of explicit confirmations turns (nCT) and the number of turns in the dialog (nCT/nT); iv) Average number of corrected errors per dialog (nCE). This is the average of errors detected and corrected by the dialog manager; v) Average number of uncorrected errors per dialog ($nNCE$). This is the average of errors not corrected by the dialog manager; vi) Error correction rate ($\%ECR$). The percentage of corrected errors, computed as nCE/ (nCE + nNCE).

The results presented in Table 2 show that the developed conversational can interact correctly with the users in most cases, achieving a success rate of 94%. The dialog success depends on whether the system provides the correct data for every objective defined in the scenario. The analysis of the main problems

Table 2. Results of the evaluation of the railway information conversational agent

	%success	nT	%confirm	%ECR	nCE	nNCE
Conversational Agent	94%	10.6	37%	93%	0.89	0.06

detected in the acquired dialogs shows that, in some cases, the system did not detect that the user wanted to finish the dialog given that the the system was developed following a mixed dialog initiative, which allow users to control the dialog flow without requiring the use of submit commands. A second problem was related to the introduction of data in the DR with a high confidence value due to errors generated by the automatic speech recognizer that were not detected by the dialog manager. However, the evaluation confirms a good operation of the approach since the information is correctly given to the user in the majority of cases, as it is also shown in the value of the error correction rate.

4 Conclusions

In this paper, we have described a technique for providing speech access to Internet by means of conversational agents. Our proposal works on the benefits of statistical methods for dialog management and XHTML+Voice. The former provide an efficient means to exploring a wider range of dialog strategies, whereas the latter makes it possible to benefit from the advantages of using the different tools and platforms that are already available to simplify system development.

Two applications have been developed to study the XHTML+Voice to develop multimodal conversational agents that improve the accessibility to information on the Internet. These conversational agents respectively facilitate the multimodal access for the search of contents in the Wikipedia encyclopedia, and the complete implementation of a speech-based interface to an Internet search engine. We have also applied our technique to develop a conversational agent that provides railway information, which integrates our statistical dialog management technique for creating automatically VoiceXML documents to prompt the user for data, as well as the necessary grammars for ASR. This conversational agent has been enhanced by means of a user simulation technique in order to facilitate the automatic learning of the dialog model and the provision of system responses that are adapted to the specific evolution of the dialog.

The results of the subjective and objective evaluations of these agents show an appropriate interaction rate during the dialog and overall operation of the different applications in the system, thus providing a solution to both general-purpose speech-based interfaces to access web contents and user-adapted conversational agents oriented to slot-filling dialog tasks. Current research lines include the adaptation of the systems for its interaction using additional languages, more complex domains, and also considering information about users' preferences.

Acknowledgements. Research funded by projects CICYT TIN2011-28620-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485), and DPS2008-07029-C02-02.

References

1. Beskow, J., Edlund, J., Granström, B., Gustafson, J., Skantze, G., Tobiasson, H.: The MonAMI Reminder: a spoken dialogue system for face-to-face interaction. In: Proc. of Interspeech/ICSLP, pp. 296–299 (2009)
2. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
3. Danielsen, P.J.: The Promise of a Voice-Enabled Web. Computer 33(8), 104–106 (2000)
4. González Ferreras, C., Escudero Mancebo, D., Cardeñoso Payo, V.: From HTML to VoiceXML: A First Approach. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2002. LNCS (LNAI), vol. 2448, pp. 266–279. Springer, Heidelberg (2002)
5. Griol, D., Hurtado, L.F., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. Speech Communication 50(8-9), 666–682 (2008)
6. Griol, D., Sánchez-Pi, N., Carbó, J., Molina, J.M.: An Agent-Based Dialog Simulation Technique to Develop and Evaluate Conversational Agents. In: Proc. of PAAMS 2011. AISC 2011, vol. 88, pp. 255–264 (2011)
7. López-Cózar, R., Araki, M.: Spoken, Multilingual and Multimodal Dialogue Systems. John Wiley & Sons Publishers (2005)
8. McTear, M.F.: Spoken Dialogue Technology: Towards the Conversational User Interface. Springer, Heidelberg (2004)
9. Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., Young, S.: Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In: Proc. of HLT/NAACL 2007, pp. 149–152 (2007)
10. Shao, Z., Capra, R.G., Pérez-Quiñones, M.A.: Transcoding HTML to VoiceXML Using Annotation. In: Proc. of ICTAI 2003, pp. 249–258 (2003)
11. Young, S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Tech. rep., Cambridge University Engineering Department, UK (2002)

Unsupervised Classification of Audio Signals by Self-Organizing Maps and Bayesian Labeling

Ricardo Cruz, Andrés Ortiz, Ana M. Barbancho, and Isabel Barbancho

Departamento de Ingeniería de Comunicaciones
E.T.S. Ingeniería de Telecomunicación. Universidad de Málaga
Campus de Teatinos s/n. 29071 Málaga, Spain.

Abstract. Audio signal classification consists of extracting some descriptive features from a sound and use them as input in a classifier. Then, the classifier will assign a different label to any different sound class. The classification of the features can be performed in a supervised or unsupervised way. However, unsupervised classification usually supposes a challenge against supervised classification as it has to be performed without any *a priori* knowledge. In this paper, unsupervised classification of audio signals is accomplished by using a Probabilistic Self-Organizing Map (PSOM) with probabilistic labeling. The hybrid unsupervised classifier presented in this work can achieve higher detection rates than the reached by the unsupervised traditional SOM. Moreover, real audio recordings from clarinet music are used to show the performance of our proposal.

Keywords: Self-Organizing maps, SOM labeling, PSOM, audio classification.

1 Introduction

Audio signal classification is a research topic which has been developed in different areas over the time. For instance, speech recognition is one of the classic problems addressed by a large number of authors [1,2,3]. Thus, many efficient algorithms have been developed for this purpose [5,6]. Other area in the audio signal classification consist on music signals processing for recognition or retrieval, which includes musical instrument classification, music genre recognition, source identification or speaker recognition. Sound source separation (segmentation) as well as speaker recognition can be performed through blind source separation techniques such as Independent Component Analysis (ICA) [7]. On the other hand, signal classification can be performed in a supervised or unsupervised way. Unsupervised classification organizes the training samples into groups according to their classes without using any *a priori* information. In other words, the samples to be classified are not labeled. On the other hand, supervised classification uses a priori information of at least some of the samples. It means that the training samples have labels according to their corresponding classes. This way, techniques which use neural networks has been developed for

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 61–70, 2012.

unsupervised and supervised classification [22,23]. However, the current trend in artificial intelligence systems is to develop hybrid systems trying to take advantage of several techniques at the same time [11,13,17,18,19,20,21]. In this work we propose an unsupervised hybrid classifier based on SOM and Gaussian Mixture Models (GMM) which aims to improve the classification performance achieved by classical SOM through a probabilistic model of the SOM units activation. This scheme aims to define the clusters on the SOM layer assigning a label [11] to unlabeled units. The GMM model built over the SOM maximizes the *a posteriori* probability when labeling an unlabeled unit. The scheme presented on this work is similar to [11] but it has been improved using *bayesian inference* to determine the label. The rest of the paper is organized as follows. Section 2 shows an introduction to the Probabilistic Self-Organizing Maps (PSOM) and its mathematical basis. Section 3 describes how the GMM is built over the SOM and the way it is used to determine the probability of a unit activation. Section 4 shows the probabilistic labeling method used on this work, based on the GMM model and *Bayesian inference*. Section 5 presents describes the audio signals used on this work as well as the preprocessing performed. Section 6 shows the experimental results and finally, Section 7 concludes this paper.

2 Probabilistic Self-Organizing Maps (PSOM)

SOM [12] is one of the most used artificial neural network models for unsupervised learning. The main purpose of SOM is to group the similar data instances close in into a two or three dimensional lattice (output map). On the other hand, different data instances will be apart in the output map. SOM consist of a number or neurons also called units which are arranged following a previously determined lattice. During the training phase, the distance between an input vector and the weights associated to the units on the output map are calculated. Usually, the Euclidean distance is used as shown in Equation 1. Then, the unit closer to the input vector is referred as winning unit and the associated weight is updated. Moreover, the weights of the units in the neighbour of the winning unit are also updated as in Equation 2. The neighbour function defines the shape of the neighbourhood and usually, a Gaussian function which shrinks in each iteration is used as shown in Equation 3. This deals with a competitive process in which the winning neuron on each iteration is called Best Matching Unit (BMU). At map convergence, the weights of the map units will not change significantly with iterations.

$$U_\omega(t) = argmin_i \|x(t) - \omega_i(t)\| \tag{1}$$

$$\omega_i(t+1) = \omega_i(t) + \alpha_i(t) h_{U_i}(t) \Big(x(t) - \omega_i(t) \Big) \tag{2}$$

$$h_{U_i}(t) = e^{-\frac{\|r_U - r_i\|}{2\sigma(t)^2}} \tag{3}$$

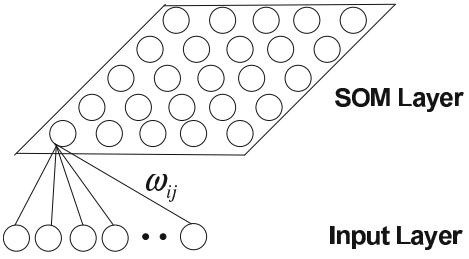


Fig. 1. Architecture of the SOM

In Equation 3, r_i represents the position on the output space (2D or 3D) and $\|r_U - r_i\|$ is the distance between the winning unit and the unit i on the output space. On the other hand, $\sigma(t)$ controls the reduction of the Gaussian neighborhood on each iteration. $\sigma(t)$ Usually takes the form of exponential decay function as in Equation 4.

$$\sigma(t) = \sigma_0 e^{\left(\frac{-t}{\tau_1}\right)} \tag{4}$$

In the same way, the learning factor $\alpha(t)$ in Equation 2, also diminishes in time. However, α may decay in a linear or exponential fashion.

Unsupervised SOM are frequently used for classification. Nevertheless, it does not use class information in the training process. As a result, the performance with high-dimensional input data highly depends on the specific features and the calculation of the clusters borders may be not optimally defined. Therefore, we used a supervised version of the SOM, by adding an output layer composed by four neurons (one per class). This architecture is shown in Fig. 1. In this structure, each weight vector ω_{ij} on the SOM is connected to each neuron on the output layer y_k.

After all the input vectors have been presented to the SOM layer, some of the units remain unlabeled. The number of unlabeled units usually depends on the map's size. Thus, the bigger the map, the greater the number of units which remain unlabeled after the training phase. A variant of the SOM consist on measuring the response of the map units instead of calculating the BMU as the unit which is closest to the input data. This deals with a probabilistic view of the output layer. In order to provide this probabilistic behaviour to the SOM, a GMM model is built over the output layer [12]. Thus, the BMU is determined not only computing the minimum distance from an input vector but also taking into account the likelihood of an unit to be the BMU. This way,the responses of the units surrounding the BMU can be taken into account. In our experiments, the prior probability of each map unit i is computed in a similar way than in [13], as shown in equation 5.

$$p(i) = \frac{\#\widetilde{X_i}}{\#\widetilde{X}} \tag{5}$$

where $\#\widetilde{X}$ is the total number of input vectors and $\#\widetilde{X_i}$ is the number of vectors whose closest prototype is ω_i is the number of sample vectors as defined on equation 6.

$$\widetilde{X_i} = \{x \in V \ / \ \|x - m_i\| \leq \|x - m_k\| \ k = 1, ...N\} \tag{6}$$

Thus, $\#\widetilde{X_i}$ can be defined as the set of data samples whose first BMU is the unit i (Voronoi set of unit i).

The GMM is built according to the equation 7 where the weights p_i for each gaussian component corresponds to the prior probabilities computed in equation 5.

$$P(x_1...x_n) = \sum_N p_i P_i(x_1..x_n) \tag{7}$$

in 7, each individual gaussian component P_i corresponds to the $n - dimensional$ weights associated to each unit (prototype vectors) [13,14]. The mean of each individual gaussian component (kernel center) is the weight vector of the corresponding unit itself, while the covariance matrix for the component i is given by the dispersion of the data samples around the prototype i. Once the GMM model has been built, the response of the unit i can be computed as the posterior probability by using the Bayes theorem.

$$p(\omega_k|x_i) = \frac{p(x_i|\omega_k)P(\omega_k)}{p(x_i)} \tag{8}$$

In equation 8, $p(\omega_k|x_i)$ represents the probability that a sample vector x_i belongs to class ω_k, while $p(x_i|\omega_k)$ is the probability density function of the prototype ω_k computed from the GMM and $p(x_i)$ is a normalization constant. This way, this posterior probability can be used to classify new samples. Nevertheless, in this work the posterior probabilities has been used to relabel the units which remain unlabeled during the SOM training process.

3 SOM Labeling Method

As commented in Section 2, the response of the SOM units computed from the posterior probabilities are used to label those units which remain unlabeled during the training process. This way, when a new sample is presented to the SOM, the BMU with the maximum *a posteriori* probability is selected. However, this BMU could be unlabeled if the map is big enough.

$$\mathcal{L} = \mathcal{L}\{argmax_i\{p(\omega_i|x) \ \forall i \ \in \mathcal{B}_N\}\} \tag{9}$$

In that case, the label of this unit is computed taking into account the response of the units in the neighborhood of the BMU. Hence, the label assigned to the BMU will be the label of the unit in the neighborhood which provides the stronger response at the unlabeled BMU as shown in equation 9.

4 Feature Extraction for Clarinet Music Classification

The classification method described in Sections 2 and 3 has been tested using audio signals from clarinet music. The goal of this work is to detect four different playing techniques from clarinet music. For this purpose, it is necessary to extract some features from the audio signals which will compose the feature space. These vectors are the data samples used as inputs to the SOM. Regarding the features extracted from the audio signal, we use features which characterize the signal in both, time and frequency domains. The features used in this work are similar than the features proposed in [16] where their discriminant capabilities are shown.

4.1 Time Domain Characterization

In this Section we describe the features used to characterize the audio signal in time domain. The duration and the shape of the envelope of the clarinet audio signals contain information about the technique used to play the notes. This way, the envelope model proposed in [15]. Thus, as in [15,16], the attack time (T_a) is considered from the first sample that reaches a 10% until it reaches the 90% of the amplitude. The release time (T_r) is considered from the last sample that reaches 70% of the amplitude to the last one over the 10% of the amplitude. On the other hand, the time between the attack time and the release time is the sustain time (T_s). T_a, T_r and T_s depend on the playing technique. The signal envelope is obtained by filtering the signal with a 5th order Butterworth filter and a cut-off frequency of 66 Hz. Then after filtering the signal, it is normalized so that the amplitude is 1. On the other hand the samples with amplitude under 2.5% of the maximum are removed. Moreover, we add another time domain T_f feature based on the presence of signal fading (i.e.: if the signal envelope is fading, $T_f = 1$, otherwise $T_s = 0$).

4.2 Frequency Domain Characterization

In order to characterize the clarinet in the frequency domain, we use the *Fast Fourier Trasnsform* (FFT). This way, the frequency axis is converted into MIDI numbers according to the equation 10.

The features described above are normalized in order to avoid one feature to have more influence than other during the training process due to different scaling among features.

$$MIDI = 69 + 12\log_2(f/440) \tag{10}$$

Taking into account that the frequency range of the clarinet is 146.83 Hz to 1975 Hz, the MIDI range of interest is 50 to 94. However, it is necessary to remove redundant information from the FFT, simplifying the signal spectrum. Thus, for a certain MIDI number n_{MIDI} the spectrum between n_{MIDI} and $n_{MIDI} + 0.5$ is considered and the maximum spectrum value of that interval is assigned to

n_{MIDI}. This way, the MIDI number spectrum (or MIDI simplified spectrum) of each note will have 45 samples. From this simplified spectrum, the pitch and the spectrum width around the pitch will be calculated.

At this point we have three time domain features (T_a, T_r, T_s and T_f) and two frequency domain features, F_p and T_w. Thus, the feature space is composed by six-dimensional vectors in the form (T_a, T_r, T_s, T_f, F_p, T_w). These six features are be discriminant enough to characterize the playing technique [15,16].

5 Experimental Results

This section presents the experimental results obtained when trying to classify audio signals from clarinet music. The dataset used in this work comes from a database containing 1433 real clarinet music recordings, playing different notes with different playing techniques. We use 6 features as described in [16] where the use of these time and frequency domain features is shown to be discriminant enough to classify notes and playing techniques. These playing techniques we classify with our classification approach are:

1. *Normal*. This corresponds to the normal playing technique. It is noted as *NO*.
2. *Staccato*. In this technique, the duration of the note is reduced to a half, that is, the main characteristic of a note played in staccato will be its short duration. It is noted as *ST*.
3. *Vibrato*. Vibrato stands for a slight fluctuation of the amplitude and the pitch notes. It is noted as *VI*.
4. *Trill*. Trill is a quavering or vibratory sound, specially a rapid alternation of sung or played noted. It is noted as *TL*.

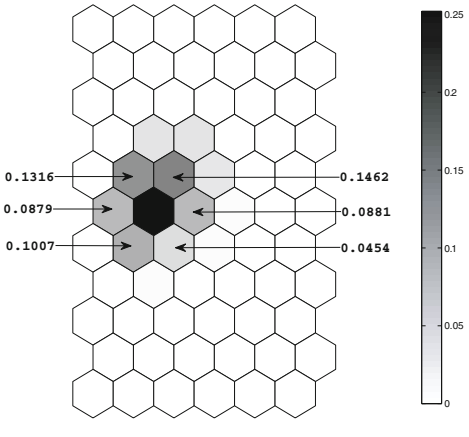
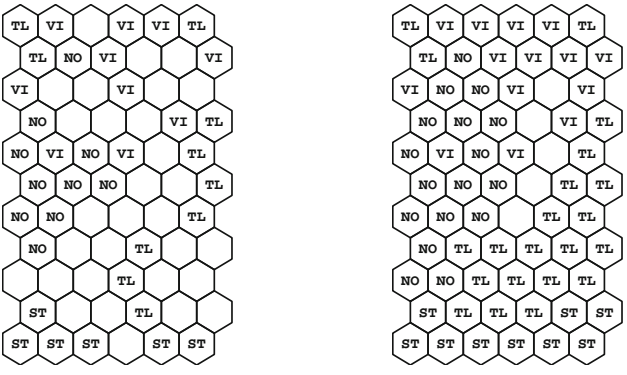


Fig. 2. Response of the units in the BMU neighboorhod. The numbers on the figure indicate the *a posteriori* activation probability of each unit.



(a) SOM **(b)** SOM+GMM

Fig. 3. SOM labeling with (a) SOM and (b) SOM+GMM Bayesian labeling

In the experiments performed, we used the 15% of the available data samples for training the SOM and for initial labeling. The 15% of the data samples for training the SOM are taken randomly from the data manifold. Then, the simulations have been repeated 50 times each in order to show the average classification rates. The initial labeling is performed through a *mayority voting scheme*. This labels the units taking into account the hits on the neighboorhod of the BMU. Thus, the label assigned to the BMU corresponds to the most frequent label of the units with more hits in the neighborhood. In order to show the responses of the SOM by using the GMM model, Figure 2 presents an example of the activation of the neighbour units of the BMU. In this figure, the black unit represents the BMU, while different gray levels show the activation level of the neighbour units. Thus, the darker the gray level, the higher the activation level of each unit, according to the scale on the right side of the figure. This activation level is the *a posteriori* probability of the unit to be activated with an input data sample.

The SOM size has been determined by experimentation. This way we used a 11x6 map size (66 map units) since it presented the best performance with the classical SOM.

Regarding the labeling process, Figure 3 shows the improvement provided by our bayesian labeling method. Figure 3a shows the labeling of the SOM units when they are labeled using a *mayority voting* scheme. On the other hand, Figure 3b shows the labels assigned to the SOM units when they are labeled using our method described on Section 3. As can be seen on this figure, our method assigns labels to the unlabeled units left when the *majority voting* scheme is used.

Figure 4 shows the classification results provided by the SOM classifier (black bar) and by the hybrid SOM+GMM classifier with bayesian labeling (hollow bar). As shown in Figure 4, our hybrid classifier outperforms the SOM classifier providing an average gain of 5% in relation to the SOM method. It is important to note that the number of VI samples we have in our database for training and testing is considerably less than the number of samples available for the other

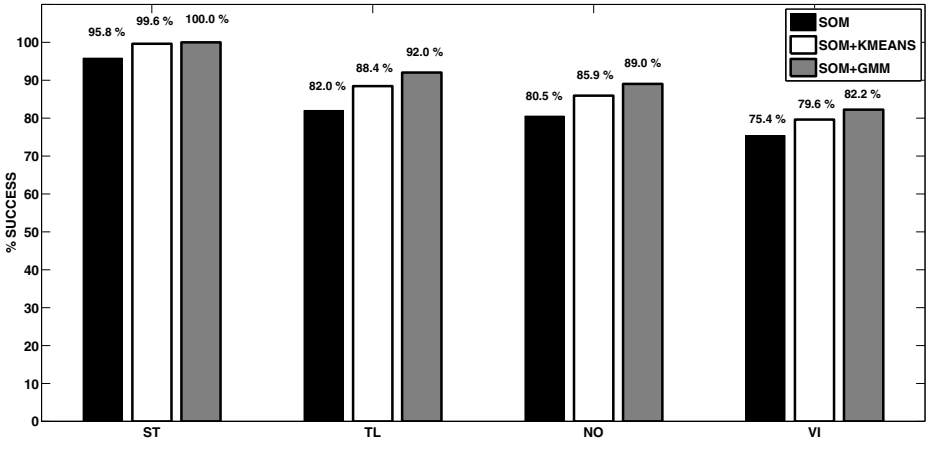


Fig. 4. Classification success for each playing style

playing styles. Hence, the classification performance for VI samples is lower than the performance obtained for NO, ST and TL playing styles.

The results obtained by the k-means labeling have been also included in figure 4, indicated as SOM+KMEANS. In this method, the k-means algorithm is used to cluster the trained SOM. Thus, all the units which remain unlabeled after the training process are labeled using the k-means clustering. This procedure is included to show the improvement that SOM+GMM clustering introduces in relation to another labeling methods.

6 Conclusions

In this paper we presented an unsupervised classification method based on a probabilistic version of the Self-Organizing Map which allow to measure the responses of each unit. Thus, opposite the classical SOM implementation, the activation level for each map unit is measured when a new data sample arrives. Moreover, we make the map more flexible leaving some unit unlabeled during the training process. Then, these unlabeled units are relabeled by bayesian inference taking into account the labels of the units on the BMU neighborhood. Although, there are other types of SOMs which grow according to the input space, the aim of our work is to provide a system trained in an unsupervised way, which is able to classify input data instances using the GMM model built over the SOM. This deals with a faster way to classify new inputs since new training is not necessary.There are other types of SOMs which grow according to the input space. However, the aim of our work is to provide a system trained once in an unsupervised way, which is able to classify input data instances using the GMM model built over the SOM. This deals with a faster way to classify new inputs since new training is not necessary. The presented approach has been tested for classifying audio signals from clarinet music. This way, four different

playing styles can be recognized with a detection rate above 80% in all the styles. Moreover, the results obtained with our approach has been compared to the results obtained by the classical supervised and unsupervised versions of the SOM. The experimental results show that our approach provides a higher detection rate than the unsupervised version of the SOM. As future work, we plan to develop an improved labeling mechanism which dynamically applies the bayesian labeling method described on this paper not only to the unlabeled units but also to the labeled units.

Acknowledgments. This work has been funded by the Ministerio de Ciencia e Innovación of the Spanish Government under Project No. TIN2010-21089-C03-02.

References

1. Holmes, W.J., Huckvale, M.: Why have HMMs been so successful for automatic speech recognition and how might they be improved? In: Speech, Hearing and Language, UCL Work in Progress, vol. 8, pp. 207–219 (1994)
2. Juang, B.H., Rabiner, L.R.: Automatic Speech Recognition A Brief History of the Technology. In: Elsevier Encyclopedia of Language and Linguistics, 2nd edn. (2005)
3. Kimura, S.: Advances in Speech Recognition Technologies. Fujitsu Sci. Tech. J. 35(2), 202–211 (1999)
4. Zils, A., Pachet, F.: Automatic Extraction of Music Descriptors from Acoustic Signals using EDS. In: Proc. of the 116th AES Convention, Berlin, Germany (2004)
5. Farahani, G., Ahadi, S.M.: Robust Features for Noisy Speech Recognition Based on Filtering and Spectral Peaks in Autocorrelation Domain. In: Proc. of the European Signal Processing Conference, Antalya, Turkey (2005)
6. Minematsu, N., Nishimura, T., Murakami, T., Hirose, K.: Speech recognition only with suprasegmental features - hearing speech as music. In: Proc. of the International Conference on Speech Prosody, Dresden, Germany, pp. 589–594 (2006)
7. Lee, J.-H., Jung, H.-J., Lee, T.-W., Lee, S.-Y.: Speech Feature Extraction Using Independent Component Analysis. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. III, pp. 1631–1634 (2000)
8. Lee, T.-W., Lewicki, M.S., Sejnowski, J.: ICA Mixture Models for Unsupervised Classification of Non-Gaussian Sources and Automatic Context Switching in Blind Signal Separation. IEEE Transactions on Pattern Recognition and Machine Intelligence 22(10), 1–12 (2000)
9. Martin, K.D.: Sound-Source Recognition: A Theory and Computational Model. PhD thesis, Massachusets Institue of Technology (June 1999)
10. Moon, T.K.: The expectation-maximization algorithm. IEEE Signal Processing Magazine, 47–60 (November 1996)
11. Ortiz, A., Gorriz, J.M., Ramirez, J., Salas-Gonzalez, D.: MR brain image segmentation by growing hierarchical SOM and probability clustering. Electronic Letters 47(10), 585–586 (2011)
12. Kohonen, T.: Self-Organizing Maps, 2nd edn. Springer series in information sciences, Berlin (1997)
13. Alhoniemi, E., Himberg, J., Vesanto, J.: Probabilistic measures for responses of Self-Organizing Map units. In: Proceedings of the International ICSC Congress on Computational Intelligence Methods and Applications, CIMA 1999 (1999)

14. Riveiro, M., Johansson, F., Falkman, G., Ziemke, T.: Supporting Maritime Situation Awareness Using Self Organizing Maps and Gaussian Mixture Models
15. Jenses, J.: Envelope model of isolated musical sounds. In: Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects, Trondheim, Norway (1999)
16. Barbancho, I., Bandera, C., Barbancho, A.M., Tardon, L.J.: Transcription and Expressiveness Detection System for Violin Music. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009), Taipei, Taiwan (2009)
17. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
18. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
19. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
20. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
21. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
22. Murtagh, F.: Multilayer perceptrons for classification and regression. Neurocomputing 2(5-6), 183–197 (1991)
23. Prasad, B., Prasanna, S.R.M. (eds.): Speech, Audio, Image and Biomedical Signal Processing using Neural Networks. SCI, vol. 83. Springer, Heidelberg (2008)

Robust Speaker Identification Using Ensembles of Kernel Principal Component Analysis

IL-Ho Yang[1] , Min-Seok Kim[2], Byung-Min So[1], Myung-Jae Kim[1], and Ha-Jin Yu[1]

[1] University of Seoul, School of Computer Science,
Seoulsiripdaero 163, Dongdaemun-gu, Seoul, Korea
[2] Advanced Research Institute, LG Electronics Inc., Seoul, Korea
heisco@hanmail.net, minseok3.kim@lge.com, sbm1210@naver.com,
arthmody@naver.com, hjyu@uos.ac.kr

Abstract. In this paper, we propose a new approach to robust speaker identification using KPCA (kernel principal component analysis). This approach uses ensembles of classifiers (speaker identifiers) to reduce KPCA computation. KPCA enhances the features for each classifier. To reduce the processing time and memory requirements, we select a subset of limited number of samples randomly which is used as estimation set for each KPCA basis. The experimental result shows that the proposed approach shows better accuracy than PCA and GKPCA (greedy KPCA).

Keywords: classifier ensemble, greedy kernel PCA, speaker identification.

1 Introduction

The accuracy of the speech and speaker recognition systems can be degraded according to the environmental condition including the channel, noise, etc. Various feature enhancement methods which transform conventional speaker features (such as MFCCs) are used in order to alleviate the problem.

PCA (principal component analysis) [1] is one of the feature enhancement methods which are widely used, but it cannot represent nonlinearly distributed data properly. KPCA (kernel PCA) [2][3] can handle nonlinearly distributed data. But the computational complexity and memory requirement are increased proportionally to the square of the number of samples which are used to estimate the KPCA basis. Therefore, we cannot apply KPCA to speaker identification directly because a large number of features can be extracted from a short utterance.

GKPCA (greedy KPCA) [4] can reduce the computational complexity and memory requirement by using greedy filtering. Greedy filtering selects a subset of the whole feature vectors with minimal representation error. However, in general, the number of subset feature vectors is very smaller than the number of whole features. Therefore, accuracy improvement of GKPCA is limited. We try to overcome this limitation by applying the concept of classifier ensemble.

In [5], a hybrid system for robust speaker identification in adverse environments is proposed. It combines two kinds of classifiers which are trained by different feature-

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 71–78, 2012.

sets. One is based on popular MFCCs and the other on the new parametric feature-set (PFS). In this research, like [5], we combine multiple classifiers which trained by different feature-sets. However, unlike [5], we extract feature-sets using KPCA transforms from original MFCCs feature-set. As in GKPCA, we apply KPCA to subsets of the whole features. At this time, the subsets are selected randomly unlike GKPCA. This process is repeated many times in order to obtain several subsets. Each subset is used to estimate the KPCA basis. Multiple classifiers (speaker identifiers) are trained with these features. Finally, the speaker identification results are combined using majority voting [6].

The remainder of this paper is organized as follows. Section 2 describes the proposed ensemble system. Section 3 and 4 show the experimental results and conclusion respectively.

2 Related Works

2.1 Speaker Identification Using GMM-UBM [7]

GMM (Gaussian mixture model) [8] is well-known modeling method in speaker recognition domain. It represents a speaker model with weighted combination of several Gaussian components (probability density functions). The equation of likelihood function is

$$p\left(\vec{x} \mid \lambda\right) = \sum_{i=1}^{M} w_i p_i\left(\vec{x}\right). \tag{1}$$

Where $\vec{x}$ and λ are input feature vector and model parameter (weights, means and covariances). w_i and p_i represent weight and probability density function of i^{th} Gaussian component respectively. In D-dimensional space, p_i is

$$p\left(\vec{x}\right) = \frac{1}{\left(2\pi\right)^{D/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\vec{x}-\vec{\mu}_i\right)^{\mathrm{T}}\left(\Sigma_i\right)^{-1}\left(\vec{x}-\vec{\mu}_i\right)\right\}. \tag{2}$$

Where $\vec{\mu}_i$ and Σ_i are mean vector and covariance matrix of i^{th} Gaussian component.

In GMM-UBM method, each speaker dependent model is adapted from UBM (universal background model) which is a speaker independent model to represent general human voice. UBM is represented by a GMM which is trained by EM (Expectation-Maximization) algorithm [8]. Also, each speaker model is represented by a GMM which is adopted from UBM by MAP adaptation [7].

In identification phase, log likelihoods of input feature vector sequence $\vec{X} = \left[\vec{x}_1, \vec{x}_2 ..., \vec{x}_t\right]^{\mathrm{T}}$ are calculated for each speaker model (equation (3)). And then, speaker identification system chooses an identified speaker which has the highest log likelihood.

$$\log p\left(\vec{X} \mid \lambda\right) = \sum_{i=1}^{t} \log p\left(\vec{x}_i \mid \lambda\right). \tag{3}$$

2.2 Feature Enhancement Using GKPCA [4]

PCA (Principal Component Analysis) [1] is used as popular feature enhancement method. It transforms original features to new basis which maximize the variance of the whole features. However, PCA may be ineffective when the feature vectors have nonlinear structure.

KPCA (Kernel Principal Component Analysis) [2][3] is the nonlinear version of PCA. It is more appropriate than PCA when the feature vectors have nonlinear structure. In KPCA, the D-dimensional feature vectors $\vec{X} = \left[\vec{x}_1, \vec{x}_2, ..., \vec{x}_t\right]^T$, $\vec{x}_i \in \Re^D$ in input space are mapped to a very high dimensional space $\Re^\infty$ (feature space) via a mapping function ϕ, and the PCA is performed in feature space.

$$\phi : \Re^D \to \Re^\infty. \tag{4}$$

The new coordinates for KPCA with centered features $\vec{X}_\phi = \left[\phi\left(\vec{x}_1\right), \phi\left(\vec{x}_2\right), ..., \phi\left(\vec{x}_i\right)\right]^T$ can be calculated by the eigenvalue decomposition problem of kernel matrix K which is a t t matrix whose elements $K_{i,j}$ are defined as

$$K_{i,j} \equiv \phi\left(\vec{x}_i\right) \cdot \phi\left(\vec{x}_j\right). \tag{5}$$

We can employ mercer kernel function to compute dot product in feature space.

$$k\left(\vec{x}_i, \vec{x}_j\right) = \phi\left(\vec{x}_i\right) \cdot \phi\left(\vec{x}_j\right). \tag{6}$$

In researches with toy examples or with a small number of training set, KPCA outperforms PCA. However, the memory requirement and the computational complexity increase quadratically with the number of training data. For this reason, it is difficult to apply KPCA-based feature extraction method to large-scale problems such as speaker or speech recognition systems.

GKPCA (Greedy Kernel Principal Component Analysis) [4] can compute KPCA using reduced training set. In GKPCA, the subset $\vec{S}_\phi = \left[\phi\left(\vec{s}_1\right), \phi\left(\vec{s}_2\right), ..., \phi\left(\vec{s}_n\right)\right]^{\mathrm{T}}$ ($\vec{S}_\phi \subset \vec{X}_\phi$) is selected from the total training set which minimizes the cost function over the total training set and the size n of the subset should be as small as possible to compute kernel matrix K and its eigenvectors. Fig. 1 shows the process of GKPCA briefly.

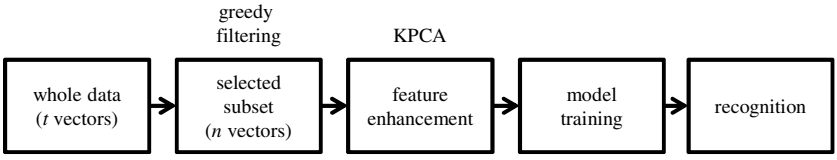


Fig. 1. Brief process of GKPCA.

3 Proposed Method

GKPCA estimates KPCA basis with n samples (speaker feature vectors) which are selected as a subset of the whole t samples ($n << t$). That is, GKPCA uses a small number of n samples. It may not be sufficient because n is very small number in comparison with t.

In this research, we propose an ensemble system using KPCA. The proposed method selects several subsets of the whole data randomly and train multiple classifiers with the enhanced subset features which are transformed using KPCA. Fig. 2 shows the proposed method in case of combining m classifiers.

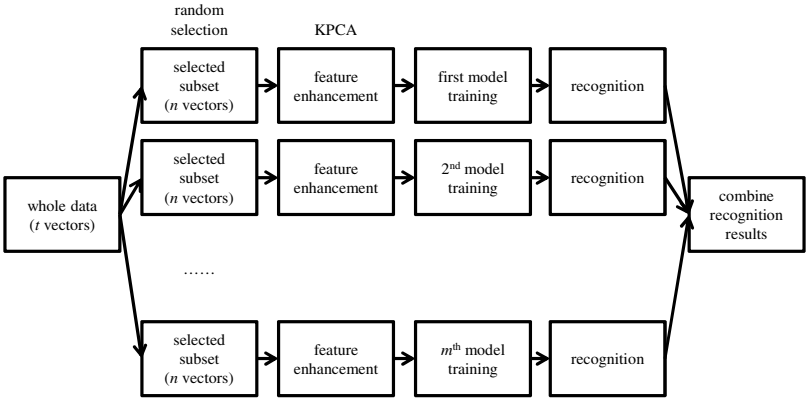


Fig. 2. Combination of m classifiers (proposed method)

The proposed method uses m times as much features than conventional GKPCA without increasing the time and memory complexity in proportion to the square of the number of training features. Therefore, we expect that it will increase the accuracy of the speaker identification.

We use majority voting scheme to combine the classifiers. This scheme chooses the class which receives the highest number of votes from the classifiers.

$$J = \arg\max_{j=1,C} \sum_{i=1}^{m} d_{i,j} \,. \tag{7}$$

Where $d_{i,j}$ means the result of i^{th} classifier for j^{th} class (0 or 1) and C is number of classes (number of speakers). If the result of i^{th} classifier is j^{th} class, $d_{i,j}$ is 1 and otherwise 0. J^{th} class is the combined result.

4 Experiments and the Results

4.1 Database and Speaker Model Training

To evaluate the proposed system in various environments, two types of speech corpora are used: ETRI PC DB and ETRI cellular phone DB. These are consist of Korean speeches which are collected by ETRI (Electronics and Telecommunications Research Institute) speech information research center at Korea. Types of utterances are 2-digit numbers, 4-digit numbers and sentences. Each recording session is repeated 4 times and each session includes 5 recording trials. In ETRI PC DB, whole recording files are saved 16kHz / 16bits mono wave format (headerless linear PCM). The wave files in cellular phone DB are compressed by mu-law (8kHz / 16bits). For the experimental equality, whole recording files of the two DBs are uncompressed and downsampled to 8kHz. The speakers of the corpora are divided into three groups according to the session terms: 'WEEK', 'MONTH' and 'SEASON'. To build a UBM, 'MONTH' speakers' 10 utterances are used as training set (1^{st} month – 1^{st} session). For training speaker models and testing, 'WEEK' speakers' 10 utterances are used (training: 1^{st} week – 1^{st} session, test: 3^{rd} week – 1^{st} session). The number of test speakers for PC DB is one hundred and that for cellular phone DB is 104. The UBM training sets consist of 101 speakers.

The GMM-UBM [7] is used for speaker model training. To build a UBM, GMMs with 256 Gaussian components are trained by EM (expectation-maximization) algorithm [8]. The number of Gaussian components is started from 1 and is doubled, e.g. 1, 2, 4, 8, and 16. The model parameters (weights, means and covariances of Gaussian components) are trained once whenever the number of components is doubled. In the final stage (256 mix), model parameters are trained ten times. The speaker models are adapted once from UBM using MAP adaptation [7] with each speaker's training set (τ =1).

To evaluate the proposed system in noisy environment, CAR, SUBWAY and RESTAURANT noise in Aurora2 DB is added at SNR 20dB and 10dB. FaNT is used for adding noise.

4.2 Feature Extraction

15 MFCCs (mel-frequency cepstral coefficients) and energy and their derivatives are derived from each utterance (window size = 25ms, shift = 10ms). Silences are removed by energy based method in feature level. The CMVN (cepstral mean and variance normalization) was applied for each utterance.

4.3 4.3 Feature Enhancement

Each basis of PCA, GKPCA and proposed method is derived from the UBM training set. Then, the whole features (UBM training set, speaker model train set, test set) are transformed using the methods.

In KPCA, Gaussian kernel function (equation (7)) is used (σ=32).

$$k\left(\vec{x}_i, \vec{x}_j\right) = \exp\left(-\frac{\left\|\vec{x}_i - \vec{x}_j\right\|^2}{2\sigma^2}\right). \tag{8}$$

In greedy filtering and random selection, we select one hundred feature vectors from the UBM training set. This size is maximum number which can process KPCA without out of memory error in our PC (2GB RAM). For the proposed method, we use an ensemble of one hundred classifiers.

4.4 Experimental Results

Table 1 shows the speaker identification rate of the overall experiments which include two types of corpora (PC and cellular phone DB). 'NOISE' represents type and SNR of noises which are added to the original wave data using FaNT. 'CLEAN' means original wave data and others mean noise added data, e.g. CAR20 (CAR noise added at SNR 20dB), RESTAURANT10 (RESTAURANT noise added at SNR 10dB). 'Baseline', 'PCA', 'GKPCA', 'Proposed' mean MFCC features and their enhanced features using PCA, GKPCA and proposed method, respectively.

Table 1. Experimental results

DB	NOISE	Base line	PCA	GK PCA	Pro posed	MAX
PC	CLEAN	**96.50**	96.30	96.30	96.20	96.50
	CAR20	82.90	80.90	85.10	**86.10**	86.10
	SUBWAY20	**70.30**	67.40	67.10	67.80	70.30
	RESTAURANT20	80.20	74.80	82.50	**83.00**	83.00
	CAR10	57.40	40.10	55.00	**58.50**	58.50
	SUBWAY10	35.60	33.10	**35.70**	35.40	35.70
	RESTAURANT10	**63.80**	61.10	60.70	63.40	63.80
CELL	CLEAN	**79.90**	78.17	78.37	79.62	79.90
PHONE	CAR20	45.19	46.63	**47.02**	46.35	47.02
	SUBWAY20	57.69	57.21	59.71	**60.38**	60.38
	RESTAURANT20	53.37	58.56	58.46	**59.42**	59.42
	CAR10	22.21	26.35	28.17	**29.33**	29.33
	SUBWAY10	32.21	38.37	38.27	**42.12**	42.12
	RESTAURANT10	32.02	38.56	39.81	**40.96**	40.96
AVERAGE		57.81	56.97	59.44	**60.61**	60.61

Our proposed method shows better average accuracy in the overall environments over the other methods. Fig. 3 shows the average speaker identification accuracy according to the number of classifiers.

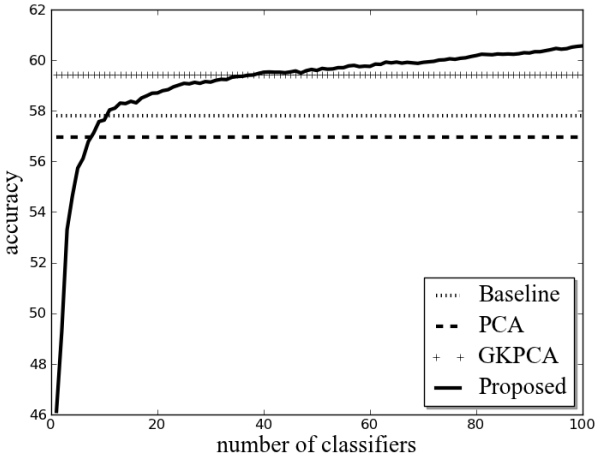


Fig. 3. Average speaker identification accuracy according to the number of classifiers

If we don't construct classifier ensemble, proposed method shows the lowest identification rate (46.12%). It is a natural result because our method is based on random selection. But, when we combining 11 classifiers, the accuracy of the proposed method is the same as the baseline (58.03%). And, when we combining up to 39 classifiers, proposed method shows the highest identification rate (59.47% at 39 classifiers and 60.56% at 100 classifiers). These results mean that the limitation of GKPCA can be overcome by the proposed method.

5 Conclusions

We have proposed an ensemble system for speaker identification using kernel PCA. In this research, like GKPCA, a small subset of the whole data is used to estimate each KPCA basis. Unlike GKPCA, each subset for a classifier is selected randomly and multiple classifiers for the ensemble system are trained. As the results, the proposed method shows better average accuracy in various environments.

Acknowledgement. This research was supported by Basic Science Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0024047).

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons (2001)
2. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)

3. Scholkopf, B., Smola, A., Muller, K.-R.: Kernel Principal Component Analysis. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 583–588. Springer, Heidelberg (1997)
4. Kim, M.-S., Yang, I.-H., Yu, H.-J.: Robust Speaker Identification using Greedy Kernel PCA. In: Proceedings of 20th IEEE International Conference on Tools with Artificial Intelligence, pp. 143–146 (2008)
5. Mashao, D.J., Skosan, M.: Combining classifier decisions for robust speaker identification. Pattern Recognition 39, 147–155 (2006)
6. Polikar, R.: Ensemble based Systems in Decision Making. Circuits and Systems Magazine (2006)
7. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing 10, 19–41 (2000)
8. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Transactions on Speech Audio Processing 3(1), 72–83 (1995)

Application of Genetic Algorithms to Optimize a Truncated Mean k-Nearest Neighbours Regressor for Hotel Reservation Forecasting

Andrés Sanz-García, Julio Fernández-Ceniceros, Fernando Antoñanzas-Torres, and F. Javier Martínez-de-Pisón-Ascacibar

EDMANS Research Group, University of La Rioja, Logroño, Spain
http://www.mineriadatos.com

Abstract. Progress in information technologies and their broad social expansion have led to the appearance of websites specialized in online hotel booking. These sites offer new approaches for customers that have demonstrated a strong tendency towards making last minute reservations. This scenario has dramatically affected the task of predicting hotel bookings, making these estimations is now much more complex using the traditional forecasting models. Given the importance of this estimation, it is crucial to find more accurate prediction models that take into account these new situations. This work aims to develop an application to predict hotel room reservations that tackles the consequences of last-minute reservations. Our proposal combines genetic algorithms optimization and truncated mean k-nearest neighbours regressor. After the analysis, we conclude that our method shows a significant improvement regarding working with historical booking information when compared with classical models. Therefore, the results of the study illustrate that our application will enable the development of useful prediction demand calendars.

Keywords: Genetic Algorithm, Optimization, k-Nearest Neighbours, Hotel Booking Forecasting, Decision Support System.

1 Introduction

In recent years, hotel guest-booking behaviour has evolved into a more complex pattern, primarily caused by an amplified range of new online booking services. These services has made clear hotel guests tend more towards last minute reservations [17]. Moreover, potential guests usually compare room prices within the same online services from different hotels to reduce costs. This novelty behaviour generates a significant impact in hotel pricing strategies, as well as in customer's choices that need to be considered for the forecasting process [2]. For this reason, hotel managers need to be able to forecast hotel arrivals more accurately in order to adequately estimate room charges. Accurate arrivals forecasting is necessary since hotel managers maximize the profits by adjusting the room prices according to the predicted demand. Nowadays, a popular trend in the hotel industry

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 79–90, 2012.

is to implement a data management system, called revenue management (RM) system, for supporting decisions in room pricing policy. RM is a growing area within the information technology development, which focuses on how companies should adjust prices to maximize their profitability. The usage of RM systems is increasing amongst hotel managers, who have a great interest in pricing and revenue optimization. It should be noted that a prediction model lays at the heart of the RM model, and the accuracy of this model is crucial to the forecast´s success [8]. The availability of historical databases allows a RM system to be implemented using data management systems for improving hotel booking forecasting.

Only a few studies focus on hotel reservation forecasting analysis. Previously, this methodology has been successfully applied to solve similar problems in other fields like the airline industry [13] Regarding hotel booking forecasting, Weatherford and Kimes [18] presented a detailed description using traditional methods for trying to solve this problem in 2003. The first attempt at forecasting consisted of clustering raw data into disjointed customer groups with the assumption of mutual independence. Then a prediction model was used to analyze each group. Other proposals for booking estimations employed Holt-Winters method [15] and Monte Carlo simulations [21]. In addition, other studies in forecasting prove that prediction accuracy can also be improved by dynamically updating the model as soon as new information is available [19]. In particular, Haensel et al. [8] adopted a method consisting of a dimension reduction and a penalizing least squares procedure for hotel booking prediction in order to reduce computation time.

In this article we propose the use of a different methodology related to soft computing (SC) instead of classical forecasting methods. SC consists of the use of computational techniques and intelligent systems for solving inexact and complex problems [16]. It involves different computational techniques such as neural networks (NN) and fuzzy sets or genetic algorithms (GAs). These approaches are all stochastic and therefore suited to investigate many real world problems [5]. In particular, GAs, a promising SC technique that has emerged in recent years can provide an efficient multivariable optimization compared to classical exhaustive search method. GAs are inspired by the law of nature [3] and can solve optimization problems through the principles of biological evolution [4]. These processes are also known as survival of the 'fittest', sexual reproduction and mutation among others [11].

Our suggested method was mainly formed by an instance based learning (IBL) algorithm like k-nearest neighbours (k-NN). The optimization of the model´s parameters and the input selection were based on genetic algorithms (GAs). We have avidly studied the optimization of a variable selection with GAs in combination with k-nearest neighbours (k-NN), an IBL algorithm, for the hotel booking prediction within a complete year. To improve results, k-NN algorithm uses a truncated mean into the regression analysis. This is one of the simplest machine learning (ML) algorithms for regression, whose core is assigning the truncated average of the values of its k nearest neighbours for each new

instance. Moreover, neighbours could be weighted in such a way so that the nearest elements contributes more to the weighted average. In our case, a specific prediction model for a hotel has been developed taking into account historical booking data and lists of local, regional and national festivities. We were able to work with real data from a Spanish hotel provided by Hoteloptimizer (http://www.hoteloptimizer.com), a new and dynamic company developing hotel RM systems. The experimental database was generated from the historical booking information of a Spanish hotel, taking into account relevant annual dates as well as other important information for a period of three years. Data were collected on a daily basis, providing useful information regarding the following characteristics: day of the week, day of the month, season, celebrations or festivals and so on. Furthermore, experiments were conducted to evaluate our proposed model comparing with four other classical methods. We have presented our results are presented by creating in a demand calendar for six months.

2 Algorithms and Methods for Regression Tasks

2.1 Truncated Mean k-NN Method

K-NN is an IBL algorithm, and for this reason it does not induce rules, decision trees, or other types of abstractions. K-NN incrementally derives its concept descriptions from a sequence of training instances without training time [1]. Furthermore, this method has been described as the most suitable to solve highly dimensional problems. The distance usually used to determine the similarity metric between the objects is the Euclidean. However the method also permits the use of other methods like Chebyshev, Manhattan, and Mahalanobis [7]. In classification tasks, k-NN uses majority vote among the classification of the K objects and it does not take into account assumptions on the distribution of predicting variables during the learning process [6]. Similiarly, k-NN uses the weighted mean of the outcomes of its k nearest neighbours in regression problems. In this work, a truncated mean is used to obtain robust estimations. The truncated mean consists of obtaining the mean of the nearest values inside the inter-quantile range defined by q and $1 - q$. Below are found the steps to predict variables using the proposed k-NN are listed:

1. Calculate the distance between training samples and the new instance.
2. Sort the k-th minimum distances to find the k nearest neighbours and determine their outputs y_k.
3. The output of the new instance is found computing the truncated mean of the k output values y_k.

The main advantage of k-NN is the effectiveness in situations where the training dataset is large and biased.

2.2 Other Machine Learning Algorithms

Through a list of models we have collated several ML algorithms, which we compare with our proposal:

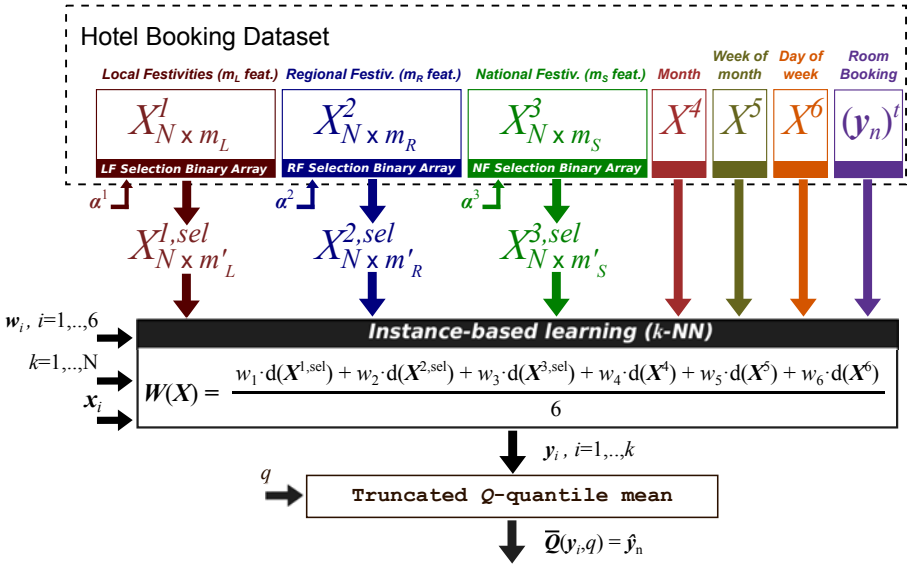


Fig. 1. Scheme combining variable selection, k-NN and trucated Q-quantile mean

- *M5P algorithm* [14]. This algorithm uses a so-called separate and conquer strategy to create a model tree in which each leaf is a linear regression model.
- *Multilayer perceptron* (MLP) [9]. MLP model can be considered as a highly adaptive nonlinear mathematical approximator. Its structure is an example of feedforward neural network where the activation of each hidden node (usually related to a sigmoid function) depends on the linear combination of its inputs. In case of numerical prediction, outer neurons are formed by linear units. The number of hidden neurons is not fixed and it defines the complexity of the model. Varying the number of neurons in the hidden layer or the input variables, the generalization performance of the MLP can be controlled.
- *Linear regression* (LINREG) [20]. Classical linear prediction model that uses a greedy method for variable selection based on Akaike information criterion.
- *Least median squared* (LMSQ) *linear regression* [12]. Based on linear regression, the method creates firstly a set of least squared regression functions using random subsamples of the training data. After that, the function with the lowest median squared error is selected as the resulting model.

3 Optimizing k-NN Regression Model with GAs

We introduce a method that mainly utilizes k-NN method for regression, with a previous variable selection to forecast hotel guest bookings for each day. It is well known the k-NN itself is a good approximator for time series. In this article, k-nearest days are selected for each day in such a way that the bias is minimized.

As mentioned before, in order to generate an accurate output, it is necessary to optimize the value of the number of the k nearest neighbours. We have to select a subset of relevant variables and choose the distance metric $d(p, q)$.

In this article, the optimization method based on GAs is selected as an effective technique, due to its ability in defining the best variables and their weighted array $\mathbf{w}_i$. Additionally, the aforementioned method will provide an optimal k and q which in turn will improve accuracy. As a result (see Figure 1), the optimization scheme obtains the optimal values of the k-NN regressor through the following steps:

1. First, variables are normalized between 0 and 1 so $\boldsymbol{X}^i \equiv \boldsymbol{X}^i_{j,\,k}\,\epsilon\,[0,\,1]$ where $j = 1, \ldots, m_L$, $k = 1, \ldots, N$, and $i = 1, \ldots, 6$.
2. Selection of the most important variables:
   (a) Matrix for local festivities in the city or in nearby cities $\boldsymbol{X}^1_{N\times m_L}$ where m_L is the number of initial local festivities and $N = 1095$ days.
   (b) Matrix for regional festivities (also in nearby regions) $\boldsymbol{X}^2_{N\times m_R}$ where m_R is the number of initial regional festivities.
   (c) Matrix for national festivities (only in Spain) $\boldsymbol{X}^3_{N\times m_S}$ where m_S is the number of initial national festivities.
   (d) Arrays of characteristics of one date in the calendar: Month of the year $\boldsymbol{X}^4_{N\times 1}$, week of the month $\boldsymbol{X}^5_{N\times 1}$, and day of the week $\boldsymbol{X}^6_{N\times 1}$.
3. Usage of binary arrays $\boldsymbol{X}^{i,sel} = \boldsymbol{X}^i \times (\boldsymbol{\alpha}_i)^t$ where $\boldsymbol{\alpha}^i$, $i = 1, \ldots, 3$ to reduce the initial selection in step 2 for the festivity matrix.
4. All matrices and column arrays are multiplied by their corresponding weights $\boldsymbol{w}^i$, $i = 1, \ldots, 6$.
5. A weighted Euclidean distance matrix $\boldsymbol{W}$ is created with the calculation of the Euclidean distances $d\,(p,\,q)$ between all dates.

$$W = \frac{\sum_{i=1}^{3} w_i \cdot d\left(\boldsymbol{X}^{i,sel}\right) + \sum_{j=4}^{6} w_j \cdot d\left(\boldsymbol{X}^j\right)}{6} \tag{1}$$

6. Determine the similar days and mean results with an inverse distance weighted average. The number of room bookings for k-nearest days are selected for each date from the Euclidean distance matrix. In order to obtain robust values, the average room booking is calculated using the truncated mean of the k-values that falls into the range limited by quantile q and $(1 - q)$.

3.1 Evolutionary Optimization Process Using a GA

A GA was used to optimize the parameters of the proposed model and to select the input variables of each model. Specifically, the parameters included into the genetic code are: number of nearest neighbours k, value of quantile q , weighted coefficients w_n, and binary arrays for input selection $\boldsymbol{\alpha}^i$ where $i = 1, \ldots, 3$ (see Figure 2). Before beginning this process, 50 individual algorithm's parameters are initialized with random values. For each individual parameter, training

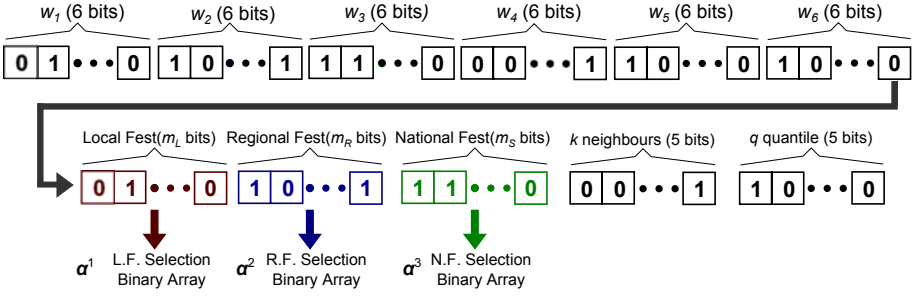


Fig. 2. Binary-coded chromosome for optimization process

data is randomly selected from the 70 % of the database. The other 30% of the database (validation data) is used to calculate the validated error between predictions and real room reservations.

The data showed a highly skewed dependent variable (Figure 3). Based on Liu and Chawla studies [10], we defined a quadratic mean which measures the magnitude of varyring quantities. This one corresponds with the square root of the arithmetic mean of the averaged squares of each group of elements. In particular, we redefine the following quadratic mean validation error function, named *RMSE10*, as:

$$RMSE10 = \sqrt{\frac{\frac{\sum_{i=1}^{n_1} e_1(i)^2}{n_1} + \frac{\sum_{i=1}^{n_2} e_2(i)^2}{n_2} + \ldots + \frac{\sum_{i=1}^{n_{10}} e_{10}(i)^2}{n_{10}}}{10}} \tag{2}$$

where $e_1(i)$ is the error for the n_1 normalized target values included within the range [0, 0.1], $e_2(i)$ is the errors which target values n_2 belong to the range (0.1, 0.2], ..., and so on.

Then, for each day of the validation database, k-nearest days are selected using the training matrix. In each case, the process is reproduced several times (10) with different training and validation data. Finally, the fitness function J to minimize is the following:

$$\min(J) = \min\left(\frac{\sum_{i=1}^{10} RMSE10_i}{10}\right) \tag{3}$$

where $RMSE10_i$ is the quadratic root validation mean squared error of the i-th process. Those individuals in generation 0 with lowest fitness function are selected to be the parents for the next generation (generation 1). So, the aforementioned generation is built as follows:

- 20 % comprising the best individuals from the previous generation. These are the parents for the next generation.
- 70 % comprising individuals obtained by crossovers from parents. The crossover process involves changing various digits in the chromosomes of the variables to be modified. These chromosomes are constructed through digits of the variables removing decimal points and creating a single set.

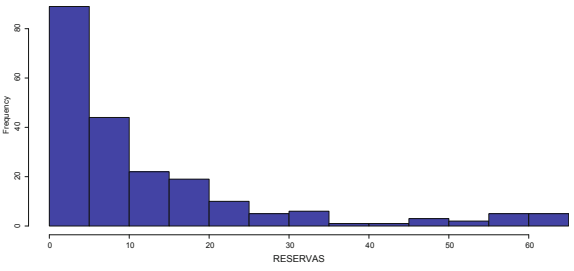


Fig. 3. Histogram of the target variable: room reservations demand

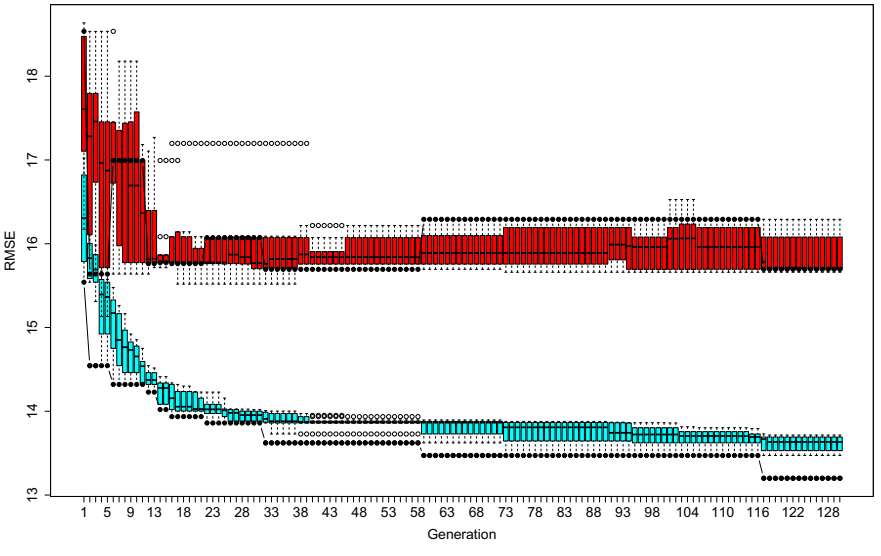


Fig. 4. Evolution of *RMSE* values for the 10 best individuals of generations 1 to 130: light grey boxes are validation RMSE values, dark grey boxes are testing RMSE values

– The remaining 10 % is obtained by the process of mutation. This mechanism follows into the creation of random chromosomes within the established ranges. The aim of such course of action is to find new solutions in an unexplored space.

This process is repeated over several generations until the fitness function of the best individual is observed to remain constant or suffers in absence of a significant variation from one generation to other.

4 Experiments and Results

The database was created from a historical reservation data from a Spanish hotel. The first step was analysing significant variables (macro-economic, social patterns, meteorological state, annual festivities). These concepts were

Table 1. Validation errors: RMSE and MAE

$Algorithm$	$RMSE_{mean}$	$RMSE_{sd}$	MAE_{mean}	MAE_{sd}	$time$
M5P	0.191	0.002	0.142	0.001	43.15
I$Bk(k = 8)$	0.202	0.001	0.152	0.001	0.022
L$INREG$	0.254	0.001	0.202	0.001	1.993
M$LP(n = 15)$	0.274	0.008	0.219	0.013	134.9
LMSQ	0.298	0.001	0.206	0.000	267.2

obtained from different sources such as: National Institute of Statistics of Spain (INE), meteorological databases and other data sources. The next step involved experts trained to select the most important variables. In this case of study, 119 attributes were selected from the following sources: historical room booking dataset, annual parameters database and set of indicators originated from one specific Spanish region.

Moreover, several scatterplots were used to identify the correlation between the aforementioned variables and the hotel overnight attributes from different Spanish regions. This mechanism was conducted in order to reduce the list of initially important variables according to the analysed correlation. After the selection of the most significant variables, we proceeded to engage in a thorough quest for a methodology which can develop useful models. These models in turn would allow us to achieve a calendar forecast for a specific hotel. Finally, the design of the prediction models was structured in 22 input attributes, which define the main characteristics of each day (month, day of the week, season, festivities in closely regions or cities, festivities in Spain, etc.). Additionally, these prediction models will generate an output variable with the number of reservations for each day. All models were created and validated using historical reservation data as a training-validating dataset from the years 2007 and 2009 inclusive. Data before 2007 was not considered in this study because of the different economic situation in Spain. Final testing dataset was created with the reservation data between January and July 2010.

4.1 Experiments Using Classical Machine Learning Techniques

The models listed in Section 2 were trained using 70 % of random sampled data, and remaining data (30 %) were employed to validate each model. The results of the training and validation process were measured along the 10 training/testing runs. Table 1 displayed the computation time, mean and standard deviation (SD) of validation root mean squared validation error ($RMSE$) and validation mean absolute validation error (MAE).

The summary of the validation errors is arranged by $RMSE_{mean}$. As observed, the best algorithm in terms of $RMSE$ was M5P tree with a validation $RMSE_{mean}$ value of 19.1 % and MAE_{mean} value of 14.2 %. By contrast,

Table 2. RMSE and MAE using data from January, 2010 to July, 2010

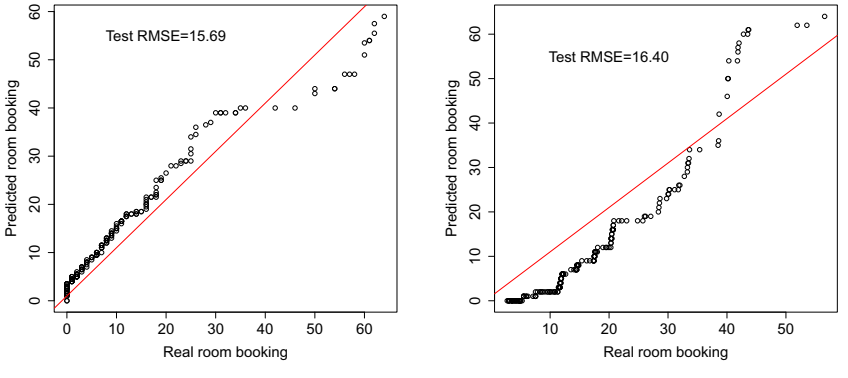
Algorithm	MAE	$RMSE$
$k-NN+GA$	0.100	0.157
M5P	0.131	0.164
I$Bk(k=8)$	0.141	0.180
I$Bk(k=10)$	0.143	0.180

Table 2 shows the results obtained after testing process. It can be said that the best algorithm was not the same as in validation process. The proposed model in this work achieved the minimum error with a testing $RMSE$ value of 15.7 % and MAE value of 10 %. This results are detailed in the next paragraph.

4.2 Experiments Using k-NN Method Optimized with GA

Taking 300 generations calculated during 50 days, the minimum J achieved with validation database was 15.20 and the minimum $RMSE$ 15.69 . In Figure 4 we observe the evolution of J and *testing* $RMSE$ value for 10 best individuals from generation 1 to 130.

Figure 6 shows evolution of test $RMSE$ of the best individuals with different fitness functions. These functions were MAE, $RMSE$, relative root squared error ($RRSE$), and quadratic mean error proposed ($RMSE10$). $RMSE10$ clearly shows superior performance. Moreover, Figure 5 compares the predicted values with real reservations for the test database (first seven months of the year 2010). In this respective case, the plot shows a good trend due to its a proximity towards the diagonal line. Furthermore, the testing RMSE obtained with the dataset of the proposed model is $RMSE = 15.69$ while the testing RMSE of M5P model is $RMSE = 16.40$.



(a) (b)

Fig. 5. Ordered plot between real and predicted room booking: a) Proposed k-NN model optimized with GA with variable selection; b) Quinlan's M5P model

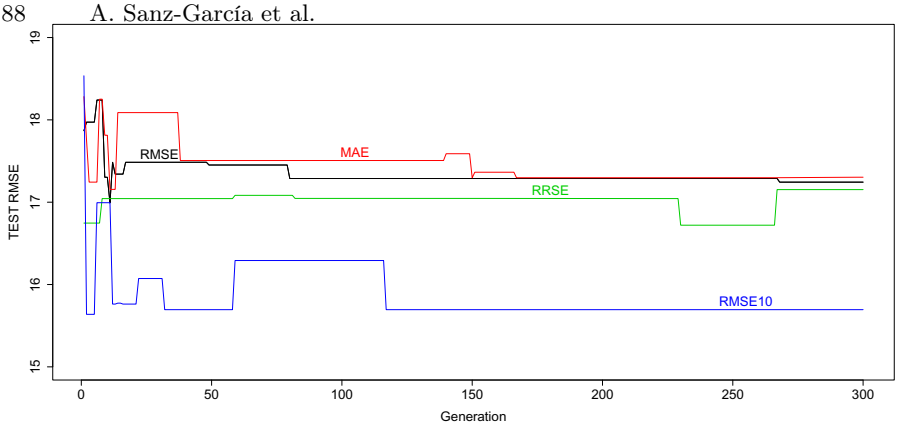


Fig. 6. Evolution of test errors using different measures as fitness function J

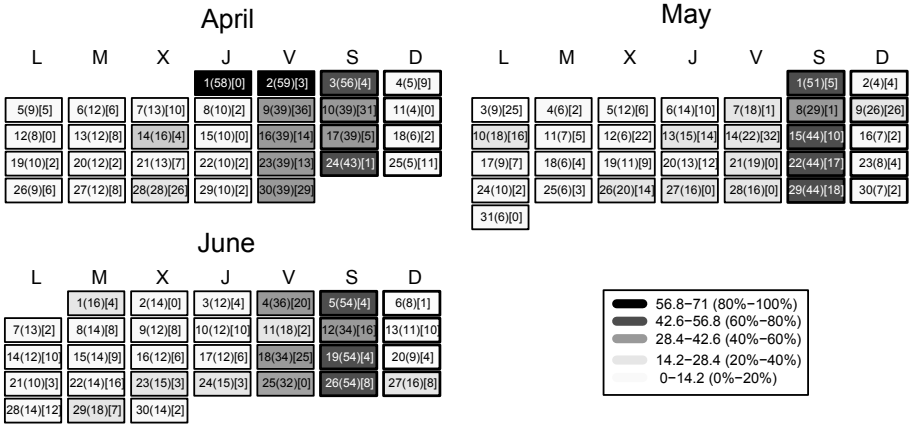


Fig. 7. Calendar with room booking forecast for April, May and June of 2010. Colour in each box represents the booking prediction level for each day

Finally, Figure 7 presents a forecasting calendar for April, May and June of 2010 calculated using the database created from years 2007 to 2009. The calendar gives the prediction for each day on the following manner: the first number on the left is the day of the month, the number shown in brackets indicates the value of the room booking prediction, and between squared brackets the confidence interval of 95% level is showed . Moreover, the booking level (type of day) is distinguished with grey scale with the interpretation of the references to grey color in the figure legend.

5 Conclusions and Future Work

In this article, a methodology has been designed with a heuristic strategy based on GAs and the combination of three steps: input selection, identification of the similarity between booking days, and the use of truncated mean for identified

days. The results with the truncated mean were more robust avoiding several days with an abnormal behaviour (i.e. congress meetings or weddings). Those days seemed to be equals if we did not use historical booking information, but they have very different room demand during the period of time selected.

As noted, global parameters have been optimized with a GA. The optimization process calculated 300 generations, but we want to emphasize that there were not significant improved after the 170 generation. This process is time-consuming (for calculating 300 generations was necessary almost two months) but however, in few number of generations correlation is almost stable.

Taking into consideration the problems in booking forecasting, we should point out that the respective model is limited by its high randomness. Nevertheless, results with the proposed model are considered to be above expectations. In conclusion, future research could be elaborated in order to obtain an important depth to this work. One of the additions to consider is the creation short time booking prediction models. The main objective of this amplifying decision is to predict prices for a short term, (week or days) by the means of actual booking curves and prices (stakeholder). Also to be noted, this course of action will also follow the type of day based on our "significant variable selection" of the long-term calendar.

Acknowledgments. We would like to convey our gratitude to *José Ignacio Pérez Moneo* for his support and accessibility with tools like RMS (`http://hoteloptimizer.com/`). On the same line, we would also like to thank to the *Autonomous Government of La Rioja* for the continuous encouragement by the means of the *"Tercer Plan Riojano de Investigación y Desarrollo de la Rioja"* on the project FOMENTA 2010/13, and to the *University of La Rioja* and *Santander Bank* for the project API11/13.

References

1. Aha, D.W., Kibler, D.: Instance-based learning algorithms. Machine Learning, 37–66 (1991)
2. Cantoni, L., Fans, M., Inversini, A., Passini, V.: Hotel websites and booking engines: A challenging relationship. In: Law, R., Fuchs, M., Ricci, F. (eds.) Information and Communication Technologies in Tourism 2011 (Proceedings of the Int. Conf. in Innsbruck, Austria), pp. 241–252. Springer, Heidelberg (2011)
3. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
4. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
5. Corchado, E., Herrero, A.: Neural visualization of network traffic data for intrusion detection. Applied Soft Computing (2010)
6. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
7. Deza, E., Deza, M.M.: Encyclopedia of distances, pp. 1–583. Springer, Heidelberg (2009)

8. Haensel, A., Koole, G.: Booking horizon forecasting with dynamic updating: A case study of hotel reservation data. International Journal of Forecasting 27, 942–960 (2011)
9. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall (1999)
10. Liu, W., Chawla, S.: A quadratic mean based supervised learning model for managing data skewness. In: SDM, pp. 188–198. SIAM/Omnipress (2011)
11. Mitchell, M.: An introduction to genetic algorithms. The MIT Press (1998)
12. Portnoy, S., Koenker, R.: The gaussian hare and the laplacian tortoise: Computability of squared- error versus absolute-error estimators. Statistical Science 12, 279–296 (1997)
13. Pölt, S.: Forecasting is difficult - especially if it refers to the future. In: AGIFORS - Reservations and Yield Management Study Group Meeting Proceedings (1998)
14. Quinlan, J.R.: Learning with continuous classes. In: 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348 (1992)
15. Rajopadhye, M., Ben-Ghalia, M., Wang, P.P., Baker, T., Eister, C.V.: Forecasting uncertain hotel room demand. Information Science 132, 1–11 (2001)
16. Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.: A soft computing method for detecting lifetime building thermal insulation failures. Integrated Computer-Aided Engineering 17(2), 103–115 (2010)
17. Sparks, B.A., Browning, V.: The impact of online reviews on hotel booking intentions and perception of trust. Tourism Management (2011)
18. Weatherford, L.R., Kimes, S.E.: A comparison of forecasting methods for hotel revenue management. International Journal of Forecasting 19, 401–415 (2003)
19. Weinberg, J., Brown, L.D., Stroud, J.R.: Bayesian forecasting of an inhomogeneous poisson process with applications to call center data. Journal of the American Statistical Association 102(480), 1185–1198 (2007)
20. Wilkinson, G.N., Rogers, C.E.: Symbolic description of factorial models for analysis of variance. Journal of the Royal Statistical Society. Series C (Applied Statistics) 22, 392–399 (1973)
21. Zakhary, A., Atiya, A.F., El-Shishiny, H., Gayar, N.E.: Forecasting hotel arrivals and occupancy using monte carlo simulation. Journal of Revenue and Pricing Management 42, 1–11 (2009)

A Social Network-Based Approach to Expert Recommendation System

Elnaz Davoodi[1], Mohsen Afsharchi[1], and Keivan Kianmehr[2]

[1] Institute for Advanced Studies in Basic Sciences
Zanjan, Iran
elnazood@gmail.com, afsharchim@iasbs.ac.ir
[2] University of Western Ontario
London, Ontario, Canada
kkianmeh@uwo.ca

Abstract. We present a hybrid method for an expert recommendation system that integrates the characteristics of content-based recommendation algorithms into a social network-based collaborative filtering system. Our method aims at improving the accuracy of the recommendation prediction by considering the social aspect of experts' behaviors. For this purpose, social communities of experts are first detected by applying social network analysis and using factors such as experience, background, knowledge level, and personal preferences of experts. Representative members of communities are then identified using a network centrality measure. Finally, a recommendation is made to relate an information item, for which a user is seeking for an expert, to the representatives of the most relevant community. Further from an expert's perspective, she/he has been suggested to work on relevant information items that fall under her/his expertise and interests.

Keywords: Recommendation Systems, Social Network Analysis, Clustering, Semantic-based Similarity, Information Retrieval, Knowledge Management.

1 Introduction

Identifying/classifying experts is an emerging research area that has been widely studied by many researchers in recent years. One objective in exploring the experts is to facilitate the process of finding the right people whom we may ask a specific question and who will answer that question for us. In the field of knowledge management, the concept of tacit knowledge refers to a type of knowledge possessed only by an individual. In general, it is difficult to communicate the tacit knowledge to others via words and symbols, or to codify it. One example of tacit knowledge is experience. Tacit knowledge usually resides in the expert's brain. Therefore finding relevant experts for a particular task is challenging. Profiling the expert and constructing the expert directories and yellow pages is an efficient and effective way to manage the tacit knowledge [9]. However,

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 91–102, 2012.

with the increasing complexity of tasks and the need for narrowed expertise in some highly on-demand areas, it is becoming more difficult to passively find the appropriate experts through directories. As an alternative approach, recommendation systems have been adopted into knowledge management systems to provide active knowledge sharing. In these systems, recommendations are made to the users according to the users' needs and interests. Many efforts have been made to improve the accuracy of explicit recommendation algorithms. However, fewer researches have focused on tacit knowledge recommendation. Recommendation systems are classified into three groups based on the way that the user models are constructed, the employed prediction methods, and also the type of items to be recommended [2]. These three groups are content-based, collaborative filtering, and hybrid methods. One important aspect that has been ignored until recently is the social element of the user behavior in making recommendations. People communicate with their peers, whom they trust, to get advice. Therefore, it seems more rational to deliver recommendations within an informal community of users and a social context. An approach that has recently received much attention is to use the social structure of user community, in addition to the users' profiles and previous behaviors, as an additional source of information in building recommender systems.

Hybrid intelligent systems are becoming popular due to their capabilities of handling many real world complex problems. In a hybrid intelligence system, a synergistic combination of multiple techniques is used to build an efficient solution to deal with a particular problem. [1,4,5]. For instance, logic-oriented neural networks greatly benefits from synergistic links between the technology of fuzzy sets and neural networks [19]. In logic-oriented neural networks some prior domain knowledge is incorporated to improve the structure of the network and establish some interesting and meaningful connections. This unique feature is not available in case of standard neural networks as they do not come with any direct interpretability capabilities which could be instantaneously taken advantage of domain knowledge [19].

This research work presents a hybrid recommendation system which is indeed a social network-based collaborative strategy that it also maintains the content-based profiles for each user. In order to design such a hybrid system, we make use of artificial intelligence-based information retrial methods and unsupervised learning (clustering) techniques for analyzing the characteristics of a social network. One advantage of this approach is that users can be recommended an item not only when this item is rated highly by users with similar profiles, but also directly, i.e., when this item gets highly scored against the user's profile. In the domain of the expert recommendation system, our proposed system discovers communities of experts and accordingly assist users to effectively find groups of experts who have users' desired tacit knowledge. In this context, the social structure of the experts' relations, captured in a social network, is used as the social component of the recommendation system. The social network of experts is constructed based on factors such as experience, background, knowledge level, and personal preferences of experts.

The rest of the paper is organized as follows. Section 2 presents several related works. Section 3 describes the proposed methodology. Section 4 demonstrates an example application of the proposed recommendation system. Finally, the paper is concluded in section 5.

2 Related Work and Our Contribution

With the rise of the eCommerce systems in the past decade, major internet retailers have begun to build recommender systems to personalize content to show to their users through an information filtering process. Recommendation systems were first employed by Amazon.com, which would show users personalized recommendations of items that the system thought they would like based on the items that they had bought or rated in the past [10]. Since then they have been widely and successfully used in the fields of movies such as the EachMovie database [3], music such as Last.fm website [8], books [17], and documents [11]. Since collaborative filtering recommendation systems carry the social characteristics of users, different concepts of social network analysis can be utilized to improve the accuracy and reliability of recommendations. Several studies have been conducted on the use of social networks in recommendation systems. For example, in [15] authors use two different social networks in a system to recommend possible collaborations for individuals. Ogata et al. [18] use social networks to facilitate finding a person to collaborate with. In [6], trust clusters are used to improve the recommendation in which clusters are based on trust rather than similarity. Further, several trust-aware recommendation methods have been proposed [12,13,14] in which it is shown that by using users' trust relations, the performance of the traditional recommender systems can be improved.

The main contribution of this paper is to design and develop a general framework that attempts to detect communities of experts in a social network and to build a recommendation system that utilizes the information extracted from expert communities to make predictions. The ultimate goal of this system is to recommend experts who carry the appropriate tacit knowledge with regards to the user information needs. To assess and evaluate the effectiveness and usability of the proposed expert recommendation system in the real world, an experiment with 315 researchers and 62 research topics (information items) has been conducted. Results have been evaluated against information collected from 23 subjects, who rated their research interests in a given list of research topics, through a questionnaire.

3 The Proposed Model

In the proposed framework, the expert recommendation system is built in four phases: 1) a profile is constructed for each individual expert by using the information collected from different online sources; 2) the semantic knowledge derived from *Wikipedia* is embedded into profiles; 3) a social network is constructed according to the similarities among the experts' profiles, communities of experts

are detected in the social network, and representatives of communities are identified; and 4) a prediction is made to recommend representatives from an expert community that has required expertise to fulfill the user's specific information need. In the rest of this section, components of the proposed system will be described in more details.

3.1 Constructing Experts' Profiles

To build rich profiles for experts, different types of relevant information need to be collected. The manual entering mean for each expert is a very time consuming task and obviously is not feasible. Therefore, to create a textual profile for each individual, a crawler automatically extracts information from relevant web pages to individuals and collects them in the profiles. Profiles constructed in this manner contain relevant information such as work experience, educational history, social and political activities, abilities and specialties, interests, etc. to each individual.

3.2 Integrating Semantic Knowledge into Profiles

In traditional text clustering methods, text documents are represented as "Bag of Words" (BOW) without considering the semantic relationships among words. In BOW approach, each document is considered as a vector in which dimensions represent all words that appear in the corpus (dictionary). The value associated to a given term reflects its frequency of occurrence within the corresponding document (term frequency or *tf*) and within the entire corpus (inverse document frequency or *idf*). Apparently, the BOW approach is limited since it only uses the set of terms explicitly mentioned in the document and ignores relationships between important terms that do not co-occur literally. For example, if two documents are about automobile sale markets, but one of them uses car and the other one uses auto as a core word, they may be falsely assigned to different clusters in spite the fact that both of them share the same topic and use synonym core words. The most common way to solve this problem is to enrich document representation with the background knowledge. There exist several ontologies like *WordNet* [16] which have been used as external sources for embedding background knowledge to text documents [7], but these ontologies are manually built and their coverage are too restricted. Their maintenance requires extreme effort as well. For these reasons, *Wikipedia*, the world largest electronic encyclopedia to date, has been recently used for text representation enrichment [20] as a more feasible strategy. *Wikipedia* is a well-formed document repository in that each article only describes a single topic. The title of each article is a succinct phrase which is considered as a concept. Equivalent concepts are related to each other by redirected links and are referred to the same page on the *Wikipedia* directory. Meanwhile, each article (concept) belongs to at least one category, and categories are organized in a hierarchical structure. All these features make *Wikipedia* a proper ontology which excels other ontologies to be used for extracting seman-

tic correlations among different concepts. In the context of our work, we take advantage of *Wikipedia* ontology to embed semantic information into profiles.

Extracting Semantic Knowledge: To extract semantic knowledge from *Wikipedia*, a content-based method is applied to enable system find proximity between *Wikipedia* concepts, thus connections between concepts can be established. In this method, each *Wikipedia* article (*i.e.*, concept) is represented by a *tf-idf* vector. The similarity between concepts are measured by computing the cosine similarity of their corresponding vectors. Then, a symmetric concept-concept matrix, called semantic kernel S, is created to present similarities among all pairs of *Wikipedia* concepts. Each element $S_{i,j}$ of this matrix determines the cosine similarity between a pair of concepts with indexes i and j, respectively, where $i, j \in \{1, 2, \ldots, c\}$ and c is the total number of concepts considered. If a row and a column refer to the same concepts or two synonym concepts, the similarity value is 1. Note that queries on synonym concepts are redirected to the same page by *Wikipedia*. Further, the more similar two corresponding concepts are, the higher the value of the corresponding entry is. This kernel represents semantic relationships among all *Wikipedia* concepts according to similarities of their corresponding articles.

Integrating Background Knowledge into Experts' Profiles: To integrate the semantic knowledge represented in matrix S into profiles, first a type of relation needs to be defined that associates profiles to *Wikipedia* concepts. For this purpose, a scheme based on the concept match is adopted to map the text document profiles to the *Wikipedia* concepts directly. In this mapping scheme, profiles are scanned and similarity-based correlations between *Wikipedia* concepts and each profile are measured. To calculate the similarity between a profile and a concept, the *tf/idf* representation method is utilized. Profiles and concepts are presented in form of vectors in which dimensions are *Wikipedia* concepts. Expert profiles are considered as a collection of documents and each concept is considered as a phrase query which can be assumed a short text document. In addition, all operations that are applied to documents in *tf/idf* approach, like porter stemmer or removing stop words, now are applied to concepts that are considered as query phrases. Finally, the cosine similarity is used to measure the similarity between pairs of corresponding vectors of document profiles and *Wikipedia* concepts. The result is presented in a document-concept matrix D in which a row entry represents a profile, columns are *Wikipedia* concepts, and each element $D_{i,j}$ denotes the cosine similarity between a document i and a concept j of *Wikipedia*, where $i \in \{1, 2, 3, \ldots, n\}$, $j \in \{1, 2, \ldots, c\}$, n is the number of documents, and c is the number of concepts.

Once the document-concept similarity matrix is built, the semantic knowledge represented by the semantic kernel can be integrated into the profile representation. For this purpose, a linear combination of the document-concept matrix D and the semantic kernel S is applied and and a new semantic-based document-concept similarity matrix R is generated. The new matrix represents the semantic-based profiles. Each element $R_{i,j}$ is calculated as follows:

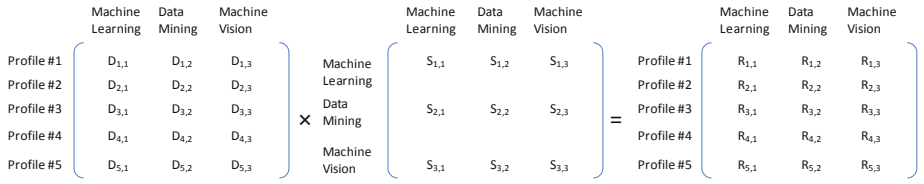


Fig. 1. Linear combination of D and S that produces R

$$R_{i,j} = \sum_{k=1}^{c} D_{i,k} \times S_{k,j} \tag{1}$$

, where k is the number of concepts, $1 \leq i \leq m$ is the row index and $1 \leq j \leq n$ is the column index. As the formula shows, the occurrences of all other concepts in i^{th} document affect the semantic relationship between j^{th} concept and i^{th} document as well by considering the weights of all concepts' similarities to the j^{th} concept. In other word, the weight of each concept's influence on the semantic relationship between a specific concept j and a document i is equal to the similarity of that concept to concept j. Figure 1 shows an example semantic-based document-concept similarity matrix resulted from the linear combination of a document-concept matrix D and a semantic kernel S. In this example, only three *Wikipedia* concepts are shown.

3.3 Construction the Social Network of Experts

In order to build the social network of experts, a relationship between experts should be defined. For this purpose, expert profiles are considered as nodes of the network and semantic-based similarities among all pairs of profiles are considered as edges. In order to compute the semantic proximity of profiles to each other, an operation widely used in social network analysis, namely folding, is applied. Assume the semantic-based document-concept similarity matrix R, in which rows represent documents and columns represent concepts. Multiplying the similarity matrix R, by its transpose R', will produce a new symmetric matrix in which rows and columns both represent profiles and elements quantify the semantic relationship between pairs of expert profiles. This recently generated similarity matrix is used to construct the links in the social network of experts. For each pair of profiles, if their corresponding similarity in the similarity matrix is a none zero value, then a link is established between the corresponding experts in the network.

Detecting Expert Communities: A community is typically thought of as a group of nodes with more interaction amongst its members than between its members and the remainder of the network. Different clustering algorithms can be applied for this purpose. In this study, the aim is to detect communities of experts such that there are stronger similarities between cluster members, in terms of expertise, knowledge, and experience, than between cluster members

and other members of network. We have chosen k-means clustering algorithm to detect the communities of experts. Further, two measures, homogeneity and separateness, are used to evaluate clustering solutions. Since these objectives are conflicting, k-means algorithm is applied with various numbers of clusters (k) until an acceptable compromise is achieved. In other words, we have to trade off between maximizing homogeneity and minimizing separation. In order to apply k-means algorithm to cluster the social network, each node (expert) is represented by a vector whose features are the semantic-based similarities to all other actors in the network. Clearly, the recentley generated similarity matrix can be used for the clustering purpose as each row of the matrix presents the similarity of an expert to all other experts.

Finding Communities Representatives: Usually clustering solution can be summarized by introducing a representative member for each cluster. In our work, since each cluster represents an expert community, the representative member of a cluster is in fact an expert who summarizes that community in terms of the knowledge, experience, and expertise carried by its members. To find a cluster representative, we have decided to use a centrality measure, called *eigenvector centrality*, which is widely used in social network analysis. According to the *eigenvector centrality*, a node is central to the extent that its neighbors are central. In other words, in a clique the individual most connected to others within the cluster and other clusters, is the leader of the cluster. Members who are connected to many otherwise isolated individuals will have a much lower score in this measure then those that are connected to groups that have many connections themselves. In our domain, the *eigenvector centrality* follows that an expert well-connected to well-connected experts can carry on valuable types of knowledge and experience much more widely than one who only has connections to lesser important experts in a community. Experts with higher scores of *eigenvector centrality* are more favorable when it is needed to find the right people whom we may ask a specific question and who will answer that question for us.

3.4 Building the Expert Recommendation System

In this work a hybrid approach, that integrates the content-based characteristics into a social network-based collaborative filtering system, is proposed to recommend the most appropriate information items to communities of experts. Information items are specified in forms of user's questions for which a user is seeking for the right experts. By applying similarity measures commonly used in information retrieval approaches, in particular cosine similarity measure, information items are recommended to members of a community if they highly match with the knowledge taste and preferences of that community members. The social network component of the proposed system captures the social aspect of the experts' behaviors. Experts collaborate with their peers on different knowledge areas to obtain new expertise and improve their own knowledge and experience. For a user who is looking for an expert for her/his information needs, our system recommends a representative of a social community whose members

have the relevant knowledge. We argue that a representative will be a better choice than an individual expert who has been recommended only based on the expert's individual profile regardless of her social relations. If more than one expert is required, more members of the same expert community are recommended. Experts in a social community are more similar to their community members than the other experts in terms of knowledge, experience, and expertise. In other words, all members of a community are experts in almost same topics. Thus, in a collaborative filtering recommendation system, to recommend information items to more than one expert, community members are better choices.

4 An Example Application

In this section, we present an example for an interesting application of our proposed expert recommendation system. We have chosen the problem of a conference chair assigning papers to be reviewed by the most relevant members of the program committee. For this purpose, 315 program committee members of the 16th *ACM SIGKDD*[1] conference have been selected as the system input. In addition, 62 keywords listed under the "conference topics" have been used as information items for which the program chair is seeking for relevant researchers. This set of keywords covers a wide range of scientific topics in the field of knowledge discovery and data mining. The main goal of this experiment is to recommend a subset of keywords to the most relevant research community with respect to the type of knowledge, expertise, and experience represented by that community.

A crawler, implemented for this work, extracts relevant information from online sources, and a profile is automatically constructed for each researcher by the system. For our experiment, the *DBLP*[2] bibliography has been crawled to collect the list of publications corresponding to each researcher. Information such as list of keywords and abstracts are retrieved for publications from digital libraries and *Google Scholar*. A profile that contains this information indicates a researcher's interests, experiences, and specialties, etc. Further, in order to build the semantic kernel S, we have to extract contents of *Wikipedia* concepts (articles). For this purpose, we automatically construct a tree structure, for a specific domain, e.g. *compute science*, which contains both category and concept pages. The tree structure will help us extract all pages related to that specific topic that appears in the root of the tree. For clustering analysis, *Weka*[3], an open source data mining tool was used. In addition, *ORA*[4], a social network analysis tool, was utilized for identifying representatives of expert communities. *ORA* calculates the eigenvector centrality of all members of a community and the member with the highest eigenvalue is reported as the representative of that community.

[1] http://www.kdd.org/kdd2010
[2] http://www.informatik.uni-trier.de/~ley/db/
[3] http://www.cs.waikato.ac.nz/ml/weka/
[4] http://www.casos.cs.cmu.edu/projects/ora/

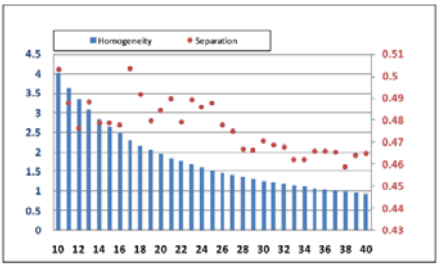


Fig. 2. Results of homogeneity and separateness for different clustering solutions

4.1 Clustering Experiments

In the community detection phase, two criteria are used to find the best clustering solution: homogeneity and separation. Indeed, the semantic-based similarities (relationships) between researchers in the social network are treated as features to describe corresponding nodes. The k-means algorithm clusters social network nodes in different groups based on values of these features. We examine various clustering solutions, generated by the algorithm using different values of k in the range of 10 to 40. Then we choose a clustering solution which is an acceptable trade off between maximizing homogeneity and minimizing separation as the best solution among others. The range of k is chosen based on the number of researchers as well as the number of information items such that the average number of researchers in each cluster varies in a reasonable range. In our experiment, the best clustering solution for the expert social network is the solution with 12 clusters. In Figure 2, the number of clusters for different clustering solutions is plotted on the horizontal axis against the values of homogeneity and separateness on vertical axes.

4.2 Recommendation Experiments

We have conducted two sets of experiments in order to investigate the performance accuracy of the recommendation system with and without the social network component. When the social network is not used, recommendations are made based on the similarity between researchers' profile and information items. In the other words, the importance of individuals in their community is neglected. In the second set of experiments, the system utilizes the social network of experts through the process. Recommendations are made based on the similarity between communities' representatives and information items. In this approach, the most appropriate experts are selected from a community whose representative has more expertise and knowledge about the requested item based on his/her profile information. In both approaches, if more than one expert is required, the system automatically suggests the second most relevant expert.

To measure the accuracy of our system, a set of 23 researchers is chosen to form a test set. Then, a questionnaire, for each researcher in the test set, is

Type of Measure	Result (%)	
	Recommendation without Social Network Component	Recommendation with Social Network Component
P@1	82.6	83.4
P@3	78.2	75.0
P@5	75.6	78.4

(a)

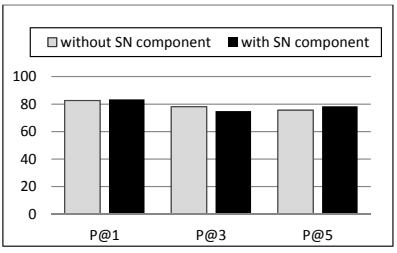


Fig. 3. The prediction accuracy of two recommendation models

designed to discover preferred information items that a researcher is interested in. The questionnaires would contain 15 items from relevant to irrelevant. Questionnaires designed for different researchers were different from each other because we prepared them based on recommended items by our system to researchers. Researchers were asked to score information items based on their relevancy to researchers' interests. To evaluate the accuracy of the recommended items, a metric called precision at n or $P@n$ was used. This precision is defined as the fraction of retrieved instances that are relevant. Precision takes all retrieved items into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. We consider k-top most relevant items that the system recommends to researchers and investigate how many of them are actually relevant considering the researchers real interests given in the questionnaire. In using of $P@n$, we set n to 1, 3 and 5. For example, $P@1$ indicates the percentage of researchers who are recommended relevant information when only one information item is considered. The same method is applied to evaluate the accuracy of the prediction when information items are recommended to members of communities whose representatives have expertise and knowledge relevant to information items for which we are looking for experts.

Figure 3.(a) demonstrates the precision values achieved when the above experiments were conducted. A $P@1$ value of 82.6%, appeared in the first column of the Table shown in Figure 3.(a) indicates that 19 out of 23 researchers in the test set, are recommended with relevant information item when only one information item is considered. In addition, the $P@1$ value of 83.4%, shown in the second column of the table, means that the first recommended information item to 83.4% of representatives are relevant. In other words, 10 out of 12 (12 is the number of communities achieved in the precious experiment) representatives are recommended with relevant item when only one information item is considered.

As described earlier, the proposed recommendation system helps users, who are looking for the most appropriate expert in a specific domain, choose representative member of each community to fulfill their information needs. In fact, a representative member can represent the knowledge and expertise of all members within the same community better than any other member in his/her community since his/her similarity to mate elements is the highest among all other mates. Thus, whenever a user searches for an expert who has relevant expertise

to a specific information domain, a reliable choice is to trust to a community representative who is recommended by the system. In addition, if more than one expert is needed, other community members can be recommended according to their importance indicated by eigenvector centrality measure; community members with higher eigenvector centrality are more reliable in that specific domain. Figure 3.(b) summarizes the performance results shown in the result Table in Figure 3.(a). As can be seen, the performance of recommendations with the social network component slightly outperforms the performance of recommendation system without the social network component. Indeed, considering three types of precisions that were calculated in each experiment, only $P@3$ value for recommendation system without the social network component is higher than its corresponding value in the second experiment. Therefore, based on the comparison made between the results, the use of social network seems to be reasonable in that it improves the prediction accuracy of the recommendation model.

5 Conclusion

We presented a hybrid expert recommendation system which is indeed a social network-based collaborative strategy that it also maintains the content-based profiles for each expert. These content-based profiles, once enriched with the semantic knowledge, are used to calculate the similarity between pairs of experts. Our system captures the social structure of the experts' relations by constructing a social network and utilizes the social characteristics of individuals while making recommendation. The proposed system employs a clustering analysis approach to discover expert communities. Representatives are identified by their centrality measures within their communities. Recommendations are made based on the relevancy of an information item, for which a user is looking for experts, to the knowledge carried by representatives of groups. The proposed framework was tested in a typical application domain with a real data set. Experimental results show that in the presence of the social network component, recommendations made by our system on average have higher accuracy than the recommendation predictions when the system neglects the social structure of individuals.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Allen, R.B.: User models: theory, method, and practice. Int. J. Man-Mach. Stud. 32, 511–543 (1990)
3. Cheung, K.-W., Tsui, K.C., Liu, J.: Extended latent class models for collaborative recommendation. IEEE Transactions on Systems, Man, and Cybernetics, Part A 34(1), 143–148 (2004)
4. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
5. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)

6. DuBois, T., Golbeck, J., Kleint, J., Srinivasan, A.: Improving Recommendation Accuracy by Clustering Social Networks with Trust. In: ACM RecSys 2009 Workshop on Recommender Systems and the Social Web (2009)
7. Hotho, A., Staab, S., Stumme, G.: WordNet improves text document clustering. In: Semantic Web Workshop of the 26th ACM SIGIR 2003, Toronto, Canada (2003)
8. Konstas, I., Stathopoulos, V., Jose, J.M.: On social networks and collaborative recommendation. In: 32nd International ACM SIGIR 2009, New York, pp. 195–202 (2009)
9. Li, M., Liu, L., Li, C.-B.: An approach to expert recommendation based on fuzzy linguistic method and fuzzy text classification in knowledge management systems. Expert Syst. Appl. 38, 8586–8596 (2011)
10. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80 (2003)
11. Liu, D.-R., Lai, C.-H., Huang, C.-W.: Document recommendation for knowledge sharing in personal folder environments. J. Syst. Softw. 81, 1377–1388 (2008)
12. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: 32nd International ACM SIGIR 2009, pp. 203–210 (2009)
13. Ma, H., Yang, H., Lyu, M.R., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: 17th ACM CIKM 2008, New York, pp. 931–940 (2008)
14. Massa, P., Avesani, P.: Trust-aware recommender systems. In: The 2007 ACM RecSys 2007, New York, pp. 17–24 (2007)
15. McDonald, D.W.: Recommending collaboration with social networks: a comparative evaluation. In: SIGCHI Conference on Human Factors in Computing Systems, CHI 2003, New York, pp. 593–600 (2003)
16. Miller, G.A., Beckwith, R., Fellbaum, C.D., Gross, D., Miller, K.: WordNet: An online lexical database. Int. J. Lexicograph 3(4), 235–244 (1990)
17. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: 5th ACM conference on Digital Libraries, DL 2000, New York, pp. 195–204 (2000)
18. Ogata, H., Yano, Y., Furugori, N., Jin, Q.: Computer supported social networking for augmenting cooperation. Comput. Supported Coop. Work 10, 189–209 (2001)
19. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
20. Wang, P., Hu, J., Zeng, H.-J., Chen, L., Chen, Z.: Improving text classification by using encyclopedia knowledge. In: 7th IEEE International Conference on Data Mining, Washington, DC, USA, pp. 332–341 (2007)

Decentralized Multi-tasks Distribution in Heterogeneous Robot Teams by Means of Ant Colony Optimization and Learning Automata

Javier de Lope[1,2], Darío Maravall[1], and Yadira Quiñonez[1]

[1] Computational Cognitive Robotics Group
Dept. Artificial Intelligence
Universidad Politécnica de Madrid
[2] Dept. Applied Intelligent Systems
Universidad Politécnica de Madrid
javier.delope@upm.es, dmaravall@fi.upm.es, ay.quinonez@alumnos.upm.es

Abstract. This paper focuses on the general problem of coordinating multiple robots. More specifically, it addresses the self-election of heterogeneous specialized tasks by autonomous robots. In this paper we focus on a specifically distributed or decentralized approach as we are particularly interested on decentralized solution where the robots themselves autonomously and in an individual manner, are responsible of selecting a particular task so that all the existing tasks are optimally distributed and executed. In this regard, we have established an experimental scenario to solve the corresponding multi-tasks distribution problem and we propose a solution using two different approaches by applying Ant Colony Optimization-based deterministic algorithms as well as Learning Automata-based probabilistic algorithms. We have evaluated the robustness of the algorithm, perturbing the number of pending loads to simulate the robot's error in estimating the real number of pending tasks and also the dynamic generation of loads through time. The paper ends with a critical discussion of experimental results.

Keywords: Multi-robot Systems, Stochastic Learning Automata, Ant Colony Optimization, Multi-tasks Distribution, Self-Coordination of Multiple Robots, Reinforcement Learning, Multi-Heterogeneous Specialized Tasks Distribution.

1 Introduction

In multi-robots systems, optimal task/job allocation or assignment is an active research problem [1], in which several central or global allocation methods have been proposed [2,3]. Some authors have also introduced decentralized or autonomous solutions, in particular inspired in the social labor division observed in some species of social insects [4,5].

In this work we take a specifically distributed or decentralized approach as we are particularly interested in experimenting with truly autonomous and decentralized techniques in which the robots themselves are responsible of choosing a

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 103–114, 2012.

particular task in an autonomous and individual manner. Under this approach we can speak of multitasks selection instead of multitasks allocation, as the agents or robots select the tasks instead of being assigned a task by a central controller.

We have already experimented with two different techniques. First, we applied the well-known threshold models inspired in the labor division of social insects [6]. Second, we employed stochastic reinforcement learning algorithms based on Learning Automata theory [7]. In this paper we also employ stochastic reinforcement learning algorithms as well as ants colony optimization-based deterministic algorithms as explained in the sequel.

Summarizing, this work focuses on the general problem of coordinating multiple robots, we propose a solution using two different approaches by applying Ant Colony Optimization-based deterministic algorithms as well as Learning Automata-based probabilistic algorithms to solve the corresponding multi-tasks distribution problem. We have considered several experiments to evaluate the system performance index for both approaches, and the results obtained are shown in article. This paper is structured as follows: section 2 describes the formal description of the problem and experimental scenario. Section 3 presents a brief introduction, basic definitions and stochastic reinforcement algoritms about learning automata methods. Section 4 briefly describes ant colony optimization methods. Section 5 describes experimental results of the evaluation of performance index, the conclusions and further work are presented at Section 6.

2 Formal Definitions

2.1 Formal Description of the Problem

The optimal multi-task allocation problem in multi-robot systems can be formally defined as follows: "Given a robot team formed by N heterogeneous robots, and given K different types of heterogeneous specialized tasks or equivalently, given K different robots roles or robots jobs and given a particular time-dependent load or number of tasks to be executed $L = \{l_1(t), l_2(t), ..., l_K(t)\}$ obtain an optimal distribution of the K tasks among the N robots in such a way that the robots themselves, autonomously and in an individual manner, select a particular task such that all the existing tasks are optimally executed".

Let $L = \{l_1(t), l_2(t), ..., l_K(t)\}$ be the different specialized tasks. Each $l_j \in L$ has a number of j sub-tasks or pending loads. Let $R = \{r_1, r_2, ..., r_N\}$ be the set of N heterogeneous mobile robots. To solve the problem, we have supposed that all members $R = \{r_1, r_2, ...r_N\}$ are able to participate in any sub-task l_j.

2.2 Experimental Scenario

We have established the following experimental scenario (Fig. 1) in order to analyze a particular strategy or solution for the coordination of multi-robot systems as regards the optimal distribution of the existing tasks. Given a set of

N heterogeneous mobile robots in a region, achieving an optimal distribution for different types of tasks. The set of N robots will form sub-teams for each type of task l_j. The sub-teams are dynamic over time, i.e. the same robots will not be always part of the same sub-team, but the components of each sub-team can vary depending on the situation.

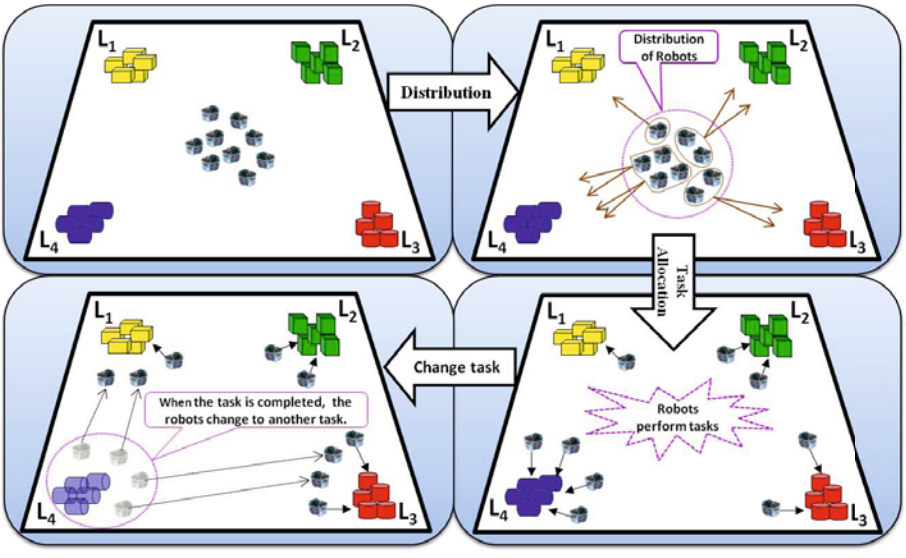


Fig. 1. Experimental scenario

Most of the proposed solutions in the technical literature are of a centralized nature, in the sense that an external controller is in charge of distributing the tasks among the robots by means of conventional optimization methods and based on global information about the system state [8]. However, we are mainly interested on truly decentralized solutions in which the robots themselves, autonomously and in an individual and local manner, select a particular task so that all the tasks are optimally distributed and executed. In this regard, we have experimented with stochastic reinforcement learning algorithms based on Learning Automata theory to tackle this hard self-coordination problem as described in the sequel.

3 Learning Automata Methods

3.1 A Brief Introduction

Learning automata have made a significant impact and have attracted a considerable interest in last years [9]. The first researches on learning automata models

were developed in Mathematical Psychology, that describe the use of stochastic automata with updating of action probabilities which results in reduction in the number of states in comparison with deterministic automata. They can be applied to a broad range of modeling and control problems, control of manufacturing plants, pattern recognition, path planning for manipulator, among other. An important point to note is that the decisions must be made with very little knowledge concerning of the environment, to guarantee robust behavior without the complete knowledge of the system. In a purely mathematical context, the goal of a learning system is the optimization of a function not known explicitly [10].

Learning is defined as any permanent change in behavior as a result of past experience, and an automata is a machine or control mechanism designed to automatically follow a predetermined sequence of operations or respond to encoded instructions [11]. The objective of stochastic learning automata is to determine how the choice of the action at any stage should be guided by past actions and responses, so when a specific action is performed the environment provides a random response which is either favorable or unfavorable [12].

3.2 Basic Definitions

A learning automaton is a sextuple $< x, Q, u, \boldsymbol{P}(t), G, \mathcal{R} >$, where x is the finite set of inputs, $Q = \{q_1, q_2, \ldots, q_m\}$ is a finite set of internal states, u is the set of outputs, $\boldsymbol{P}(t) = \{p_1(t), p_2(t), \ldots, p_m(t)\}$ is the state probability vector at time instant t, $G : Q \rightarrow u$ is the output function (normally considered as deterministic and one-one), and $\mathcal{R}$ is an algorithm called the reinforcement scheme, which generates $\boldsymbol{P}(t+1)$ from $\boldsymbol{P}(t)$ and the particular input at a discrete instant t.

The automaton operates in a random environment and chooses its current state according to the input received from the environment. The new state probabilities distribution $\boldsymbol{P}(t+1)$ reflects the information obtained from the environment. The random environment has a set of inputs u and its set of outputs is frequently binary $\{0, 1\}$, with '0' corresponding to the reward response and '1' to the penalty response. If the input to the environment is u_i the environment produces a penalty response with probability c_i.

Fig. 2 shows the feedback configuration of a learning automaton operating in a random environment. At each instant t the environment evaluates the action of the automaton by either a penalty '1' or reward '0'. The performance of the automaton's behaviors is the average penalty

$$I(t) = \frac{1}{m} \sum_{i=1}^{m} p_i(t) c_i \tag{1}$$

which must be minimized. In order to minimize the expectation of penalty (1), the reinforcement scheme modifies the state probability vector $\boldsymbol{P}$. The basic idea is to increase p_i if state q_i generates a reward and to decrease p_i when the same state has produced a penalty. A great number of reinforcement schemes for minimizing the expected value of penalty have been studied and compared. One

of the most serious difficulties that arise in learning automata is the dichotomy between learning speed and accuracy. If the speed of convergence is increased in any particular reinforcement scheme, this action is almost invariably accompanied by an increase of convergence to the undesired state [13,14].

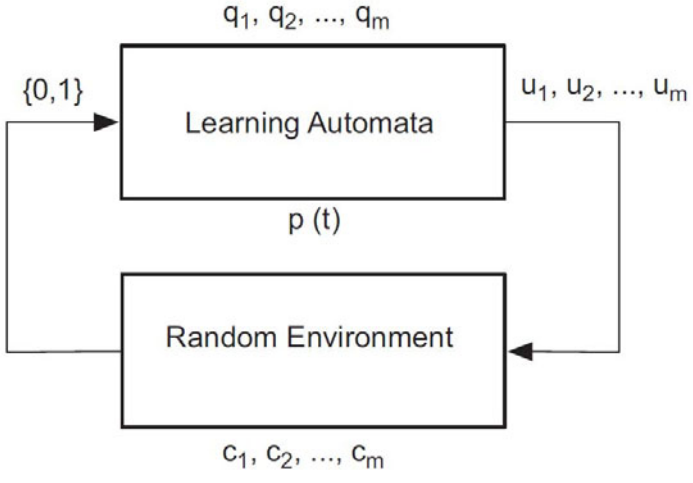


Fig. 2. Interaction of learning automaton with random environment

3.3 Stochastic Reinforcement Algorithms in Learning Automata Theory

In the technical literature a widely used stochastic reinforcement algorithms is L_{R-I}, which stands for Linear Reward-Inaction algorithm.

Let us suppose that the action chosen by the automaton at instant t is ϕ_i, for the L_{R-I} the updating of the action probabilities is as follows:

$$p_i(t+1) = p_i(t) + \lambda\beta(t)\left[1 - p_i(t)\right] \tag{2}$$

$$p_j(t+1) = p_j(t) - \lambda\beta(t)p_j(t) \quad \forall j \neq i, 1 \leq j \leq N \tag{3}$$

where $0 < \lambda < 1$ is the learning rate and $\beta(t)$ is the environment's response: $\beta = 1$ (favorable response or reward) or $\beta = 0$ (unfavorable response or penalty in which case the algorithm do not change the probability, i.e. inaction).

Let's suppose that there are K different specialized tasks, then we designate by $p_{ij}(t)$, the probability at instant t that robot r_i selects task l_j these probabilities hold:

$$0 \leq p_{ij}(t) \leq 1;\ \sum_{i=1}^{N} p_{ij}(t) = 1;\ i = 1, 2, ..., N\ robots;\ j = 1, 2, \ldots, K\ tasks \tag{4}$$

Initially, without previous robot's experience these probabilities are initialized at the "indifference" position as follow:

$$p_{ij}(0) = \frac{1}{K} \; for \; i = 1, 2, ..., N \; robots \; and \; j = 1, 2, \ldots, K \; tasks \tag{5}$$

Afterwards it starts the learning process in which each robot updates its election probabilities according to the following conventional updating rule:

$$p_{ij}(t+1) = p_{ij}(t) + \lambda\beta(t)\left[1 - p_{ij}(t)\right] \tag{6}$$

where $0 < \lambda < 1$ is the learning rate with a fixed value of 0.2; $\beta(t)$ is the usual reward signal generated by the environment of the learning automata with the following interpretation: $\beta(t) = 1$; reward if and only if for the corresponding task l_j at instant t it holds that $\#R_j(t) \leq \#L_j(t)$, i.e. the number of robots performing task l_j is lower than the number of tasks l_j to be executed; $\beta(t) = 0$; penalty if and only if $\#R_j(t) > \#L_j(t)$; i.e. when the number of robots performing task l_j is greater than the number of tasks l_j or whenever there are not pending tasks to be executed the automata receives a penalty signal. In few words: at each instant t the environment evaluates the action of the automata, when the response generated by environment is 1 means that the action is "favorable" and if the response value is 0 corresponds to an "unfavorable" as follow:

$$\beta_{L_j}(t) = \frac{\#R_j}{\#L_j} = \begin{cases} \text{If} \leq 1 \text{ then reward } \beta = 1 \\ \text{If} > 1 \text{ then penalty } \beta = 0 \end{cases} \tag{7}$$

4 Ant Colony Optimization Methods

For over many years, communities or colonies of social insects have been deeply studied by some researchers, as they provide fascinating examples of functional collective behavior. Ant Colony Optimization (ACO) is a meta-heuristic approach that was introduced in the early 1990's by Dorigo et al. in [15,16]. The general idea of ACO approach is to solve combinatorial optimization problems based by the behavior of real ants, more specifically, the inspiring source is how ants can find shortest paths between food sources their nest. ACO algorithms are stochastic search procedures based on a parameterized probabilistic model [17], called by the authors "the pheromone model".

In this case, a generic robot r_i selects the tasks in a deterministic way based on "forces" $f_{ij}(t)$. These forces are updated, after being initialized at the "indifference" position, as follows:

$$f_{ij}(t+1) = \rho f_{ij}(t) + (1-\rho)\beta(t); 0 \leq \rho \leq 1 \tag{8}$$

where ρ is the usual learning rate of ant colony optimization-like algorithms and $\beta(t)$ is the reward/penalty signal at instant t with the same exact interpretation than for the learning automata-based probabilistic algorithms.

5 Experimental Results

We have carried out a series of experiments to evaluate the system performance index by applying Ant Colony Optimization-based deterministic algorithms as well as Learning Automata-based probabilistic algorithms to solve the optimal distribution of the tasks among the N robots; so that all of them are executed by means of the minimum number of robots. The ideal objective is that the performance index or learning curve corresponding to the load $l_j(t)$ of each task tend asymptotically to zero for all curves in the minimum time and using the minimal possible number of robots for task execution.

In the simulations we have considered some variants such as: the multi-robot system size, different loads $l_j(t)$ for each type of task, two different ways to carry out the tasks selection, the additive noise generation to simulate the robot's error and the dynamic generation of tasks $l_j(t)$ over time. According to the results obtained with eq. 6 and eq. 8 we have used for the learning automata-based probabilistic algorithms and for ant colony optimization-based deterministic algorithms two mechanisms for the selection of tasks:

1. Maximum principle: at each instant t choose the task that has the highest probability for all $p_{ij}(t)$.
2. The strictly random method: using the probabilities $p_{ij}(t)$ in the strict sense of the word, it generates a random number with uniform distribution $(0-1)$ and it selects the appropriate task to the value obtained by the method of inversion of discrete probability distributions.

Table 1. Shows a scheme of the experiments performed with their respective variants

		Without Noise		With Noise	
	Mechanisms / Approaches	Maximum principle	Strictly random method	Maximum principle	Strictly random method
Not dynamic task	Ant Colony Optimization	Fig.4(a)		Fig.4(b)	
	Learning Automata	Fig.5(a)		Fig.5(b)	
Dynamic task	Ant Colony Optimization	Fig.6(a)		Fig.6(b)	
	Learning Automata	Fig.7(a)		Fig.7(b)	

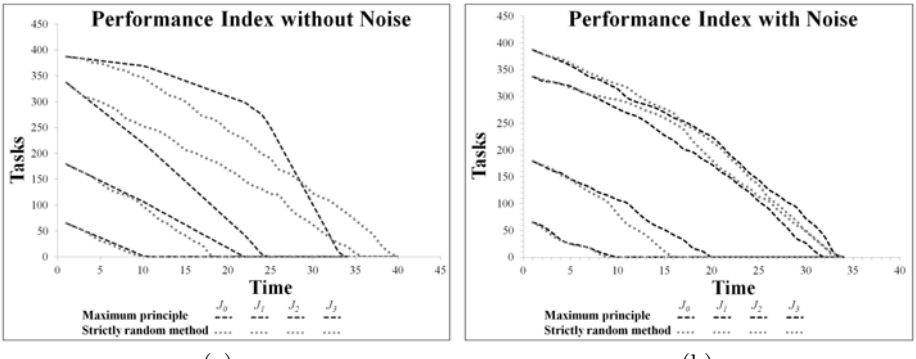
5.1 Evaluation of the Performance Index

To evaluate the evolution of the performance index we have introduced additive noise, perturbing the number of pending loads to simulate the robot's error in estimating the real number of pending tasks. The noise generated is modeled using a normal distribution ("White Noise") as follows:

$$Noise = R + R * S = R(1 + S) \tag{9}$$

where $Noise$ is the noise generated to the number of pending loads $l_i(t)$, which is proportional to the amplitude of the noise R without perturbing, S is a Gaussian distribution with a mean of '0' and a typical deviation '0.005' $N(0, 0.005)$.

Fig. 3 and Fig. 4 show the evolution of the system performance index obtained for self-election of heterogeneous specialized tasks through ant colony optimization-based deterministic algorithm as well as learning automata-based probabilistic algorithms, using both mechanisms: maximum principle and the strictly random method, with a team of robots formed by 20 – 30 heterogeneous robots and 4 types of heterogeneous specialized tasks with different loads. Each experiment has been run 10 times and the results shown are the mean of all.

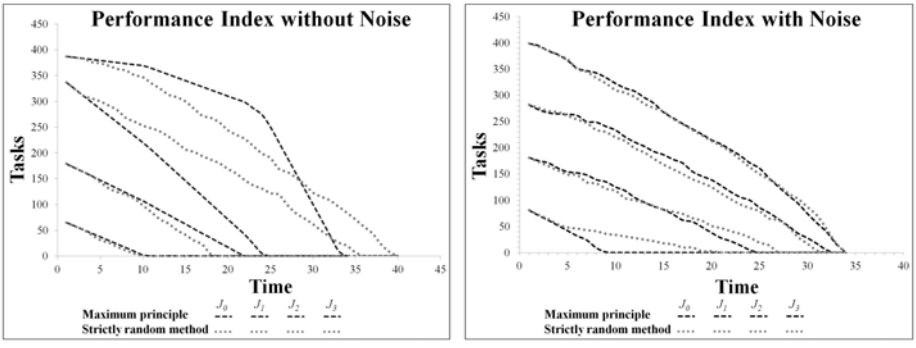


(a) (b)

Fig. 3. Learning curves with the evolution of the system performance index for self-election of tasks using Ant Colony Optimization-based deterministic algorithms

Fig. 3(a) shows the performance index without noise for both mechanisms, it can be observed that the maximum principle provides better performance instead of strictly random method. However, Fig. 3(b) shows the performance index perturbing the number of pending loads and for both mechanisms the performance is better because they finish in fewer time than in Fig. 3(a). In this case, the best results are obtained with strictly random method instead of the maximum principle.

Fig. 4(a) shows the performance index without noise for both mechanisms and Fig. 4(b) presents the performance index generating additive noise in the number of pending tasks. It can be noted that Fig. 4(b) obtains better results for both mechanisms than Fig 4(a) using Learning Automata-based probabilistic algorithms. It can be observed that learning curves corresponding to the load $l_j(t)$ of each task tend asymptotically to zero for both methods. Also the results shown that the generation of additive noise does not affect the performance of the approach, on the contrary, in some cases better results are obtained with the generation of noise.



(a) (b)

Fig. 4. Learning curves with the evolution of the system performance index for self-election of tasks using Learning Automata-based probabilistic algorithms

5.2 Dynamic Tasks Generation

In the previous experiments, the number of loads for each type of task is determined from the beginning of the simulation and there is not any change until the end of the execution. To evaluate the performance of the algorithm we have generated dynamic tasks. This idea was rescued from classical models of queues simulation, so we have used Poisson distribution to determine the probability of generating a number of tasks through time:

$$f(k;\lambda) = \frac{e^{\lambda}\lambda^{k}}{k!} \tag{10}$$

Specifically we will have a different distribution for $k = 1$ to 100. Each λ is a positive real number that representing the number of tasks expected to be generated during a time interval. For that expected number of tasks generated is decreasing, and therefore the system is stable, we have parameterized this constant λ as follows:

$$\lambda(t) = \sigma - \alpha * t \tag{11}$$

where σ is the initial value (for example, 10 or 20) and α is a factor of "reduction tasks" that initially we have defined to 1. Finally, t corresponds the time of execution at each instant.

Fig. 5 and Fig. 6 show the evolution of the system performance index with dynamic tasks generation through time using the Poisson distribution. Experiments have been performed 10 times and the results shown are the mean of all, we have also additive noise generated in the loads with the maximum principle and the strictly random method. In the results it can be observed dynamic tasks generation, the tasks number generated is decreasing over time. All learning curves tend to zero in both methods and not affected the performance of the approach with the generation of additive noise, sometimes there are better results with noise.

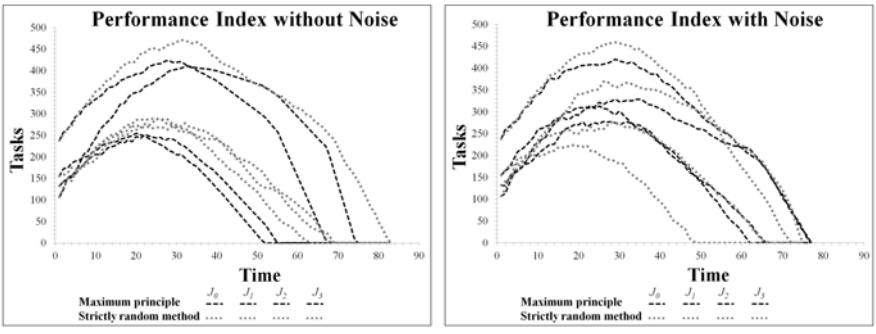


Fig. 5. Dynamic task generation: learning curves with the evolution of the system performance index using Ant Colony Optimization-based deterministic algorithms

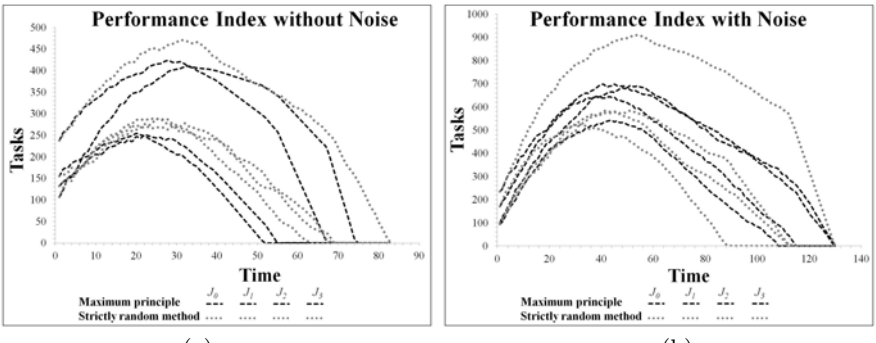


Fig. 6. Dynamic task generation: learning curves with the evolution of the system performance index using Learning Automata-based probabilistic algorithms

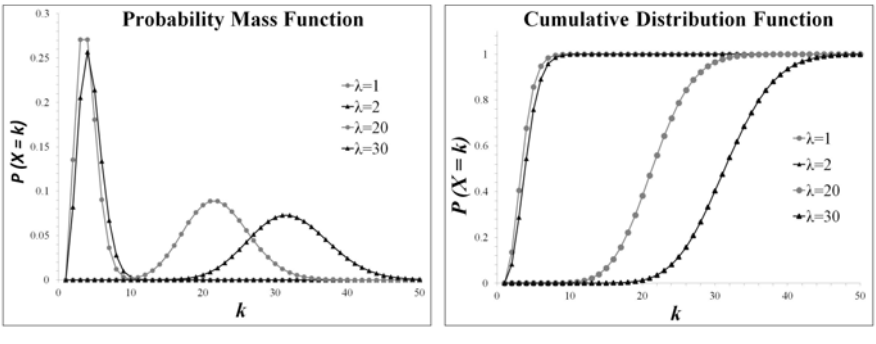


Fig. 7. The index k represents the number of tasks expected to be generated during a time interval for different values of λ and $P(X = k)$ describes the probability that a value of variable X with a given probability distribution is equal to k

Fig. 7 shows the probability mass function and the cumulative distribution function obtained in experiments with dynamic task generation using the Poisson distribution.

6 Conclusions and Further Work

In this paper we have applied two different approaches to the self-coordination problem of multi-robot systems in the heterogeneous multi-tasks distribution by applying Ant Colony Optimization-based deterministic algorithms as well as Learning Automata-based probabilistic algorithms. To carry out the selection of tasks in both approaches we used two mechanisms: maximum principle and the strictly random method and, in most experiments the best results are obtained with strictly random method instead of the maximum principle. We have generated additive noise to evaluate the robustness to both approaches, perturbing the number of pending load, to simulate the robot's error in estimating the real number of pending tasks, according to the results obtained the noise generated does not affect the performance of the approaches since the best result are obtained by generating noise in the pending loads. We have also studied the performance index with dynamic generation of loads through time and the results confirm that the robots are capable to select in an autonomous and individual manner the existing tasks without the intervention of any global and central tasks scheduler. We have shown that both approaches can be efficiently applied to solve this self-coordination problem in multi-robot systems obtaining truly decentralized solutions.

References

1. Gerkey, B., Mataric, M.: Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures. In: IEEE International Conference on Robotics and Automation, pp. 3862–3868 (2003)
2. Gerkey, B., Mataric, M.: A formal analysis and taxonomy of task allocation in multi-robot systems. Intl. J. of Robotics Research, 939–954 (2004)
3. Farinelli, A., Locchi, L., Nardi, D.: Multirobot systems: a classification focused on coordination. IEEE Transactions on Systems, Man and Cybernetics, 2015–2028 (2004)
4. Oster, G., Wilson, E.: Caste and ecology in the social insects. Monographs in Population Biology. Princeton Univ. Press (1978)
5. Robinson, G.: Regulation of division of labor in insect societies. Annu. Rev. Entomol., 637–665 (1992)
6. Quiñonez, Y., de Lope, J., Maravall, D.: Bio-inspired Decentralized Self-coordination Algorithms for Multi-heterogeneous Specialized Tasks Distribution in Multi-Robot Systems. In: Ferrández, J.M., Álvarez Sánchez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2011, Part I. LNCS, vol. 6686, pp. 30–39. Springer, Heidelberg (2011)
7. Quiñonez, Y., Maravall, D., de Lope, J.: Stochastic Learning Automata for Self-coordination in Heterogeneous Multi-Tasks Selection in Multi-Robot Systems. In: Batyrshin, I., Sidorov, G. (eds.) MICAI 2011, Part I. LNCS, vol. 7094, pp. 443–453. Springer, Heidelberg (2011)

8. Gerkey, B., Mataric, M.: Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In: IEEE International Conference on Robotics and Automation, pp. 3862–3868 (2003)
9. Narendra, K., Thathachar, M.: Learning Automata: An Introduction. Prentice-Hall, Englewood Cliffs (1989)
10. Narendra, K., Thathachar, M.: Learning Automata: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, 323–334 (1974)
11. Obaidat, M., Papadimitriou, G., Pomportsis, A.: Guest Editorial Learning Automata: Theory, Paradigms, and Applications. IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics, 706–709 (2002)
12. Maravall, D., De Lope, J.: Fusion of Learning Automata Theory and Granular Inference Systems: ANLAGIS. Applications to Pattern Recognition and Machine Learning. Neurocomputing 74, 1237–1242 (2011)
13. Narendra, K., Wright, E., Mason, L.: Applications of Learning Automata to Telephone Traffic Routing and Control. IEEE Transactions on Systems, Man, and Cybernetics, 785–792 (1977)
14. Narendra, K., Viswanathan, R.: A Two-Level System of Schotastic Automata for Periodic Random Environments. IEEE Transactions on Systems, Man, and Cybernetics, 285–289 (1972)
15. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: an autocatalytic optimizing process, Technical Report TR91-016, Politecnico di Milano (1991)
16. Dorigo, M.: Optimization, learning and natural algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan (1992)
17. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical Computer Science 344(2-3), 243–278 (2005)

Lipreading Procedure for Liveness Verification in Video Authentication Systems

Agnieszka Owczarek and Krzysztof Ślot

Institute of Electronics, Technical University of Lodz,
Wolczanska Street 211/215, 90-924 Lodz, Poland
{agnieszka.owczarek,kslot}@p.lodz.pl

Abstract. The following paper proposes a novel procedure for liveness verification in video authentication systems. Decision on granting or denying access to the system is based on visual information only. System prompts random sequences of predefined syllables in which isolated visems are to be recognized. Features that represent outer lip contour are extracted on the basis of dynamic programming and B-spline approximation. Experiments for Polish language utterances show that seven classes of visems can be recognized with 76% of efficiency. Average false acceptance rate for liveness does not exceeds 9% and false rejection rate 7%.

Keywords: liveness verification, lip contour extraction, video authentication system.

1 Introduction

Liveness verification is a fundamental and challenging problem in video authentication systems, which need not only to ensure reliable recognition but also protect against possible spoofing attacks. In majority of cases there is no possibility to ensure supervised authentication process. Therefore, biometric data presented to a system have to be examined whether it is captured from a physically present person as opposed to prerecorded videos or still face images. Although, there have been much publications regarding liveness verification in bimodal audio-video system [1,2,3], liveness aspects in visual only systems are greatly limited [4]. In majority of published research only a need of such procedures is highlighted [5,6].

In this paper a lipreading procedure for liveness verification is proposed. Decision on liveness is made on the basis of visual recognition of isolated visems in syllable sequences randomly prompted by the system. A set of 21 syllables of Polish language, representing seven visems has been selected for the purpose of liveness verification. Also, a novel approach for lip feature extraction was presented. Visems are modeled by coordinates of control points of B-splines that approximate outer lip contours, extracted using dynamic programming technique. Experiments show good performance of the proposed solution.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 115–124, 2012.

2 State of the Art

A fundamental task in visual authentication field is to find a suitable representation for speech-related features. By analyzing shape and motion of visible articulators of a speaker - such as lips, tongue or teeth, speech reading can become possible. There have been many studies in this area so far.

Firstly, lip texture information can be exploited for this purpose (this is a widely used 'low-level pixel-based' approach). In [7,8] principal component analysis has been applied to raw lip intensity images to reduce its dimensionality, and the reduced vector has been used as a visual feature descriptor. In [9] discrete cosine transform (DCT) coefficients of gray-scale lip images have been adopted for visem representation. Unfortunately, lip textural features are sensitive to illumination mismatches that typically exist between the training and test data sets.

The second direction in research is based on lip geometry modeling (high-level lip contour based approach). It usually requires lip contour extraction and contour fitting methods, followed by computation of a variety of features such as contour perimeter, lip area or horizontal/vertical openings. The relevant methods typically apply deformable templates [10,11], active shape models (ASM) [12,13,14], and snakes [15]. For example, snakes allow to parameterize a closed contour. Deformable templates use simpler parameterized models, such as modeling the outer contour of the mouth by two intersecting parabola segments. More complex models can also be built. For example, the mouth-open template given in [16] consists of five parabolic segments.

Most of the practical techniques utilize a combination of lip texture and geometric lip shape features. For example, in [17], the lip feature vector is formed by concatenating the Karhunen Loeve transform (KLT) coefficients of the inner-outer lip contour points with texture information embedded in eigenlip images (an analog to eigenfaces). Similarly, in [18] snake is used to extract visual features of geometric space and KLT to extract principal components in the color eigenspace. Hybridization of different intelligent techniques frequently lets to build an efficient solution to deal with drawbacks of individual techniques [19].

Finally, explicit lip motion information can also be used for speech-reading. In [20] gradient vector flow (GVF) snakes are applied to extract outer lip contours. Then lip movements are calculated at several (ten) predefined points and the resulting feature dimensionality is reduced by PCA.

3 Proposed Solution

Proposed liveness verification system structure is shown in Fig. 1. It consists of four processing stages. Firstly, a random sequence of predefined syllables, containing set of appropriately selected visems, is prompted by the system. Video frames corresponding to every syllable are stored separately (I). Then for every frame lip region is found, outer lip contour is extracted and appropriate feature vector is built (II). In (III) classification is made. For every frame one of possible visems is assigned

using Ada Boost algorithm [21]. Finally, decision on liveness is made (IV) . For each syllable, it is checked if in a set of frames corresponding to this syllable, some expected visem was identified. If some minimum correct visem identification rate in syllable sequence is achieved, liveness verification test is successfully passed.

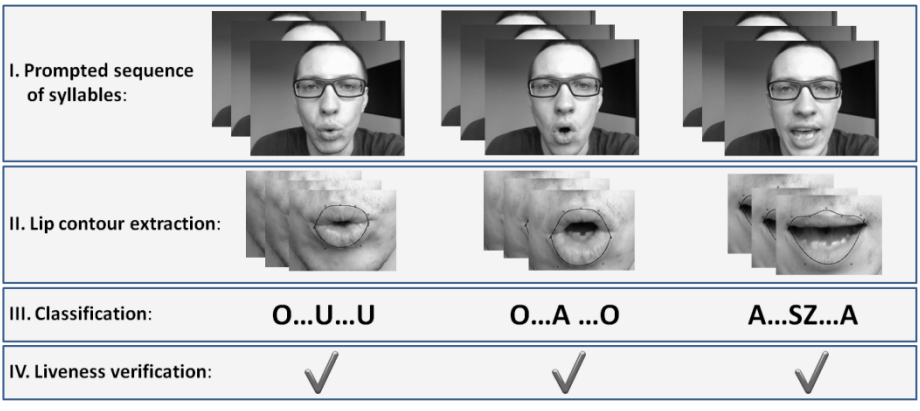


Fig. 1. The proposed visual liveness verification system components

To increase reliability of the system it is vital to use visems that correspond to phones articulated with a strong contribution of lips.

3.1 Lip Contour extraction

The flowchart of lip contour extraction procedure is presented in Fig. 2. Firstly, in order to roughly estimate lip region and to find lip key points the initial preprocessing is done (I). The Haar classifier [22] is applied to determine face position in each video frame. Lip area is estimated on the basis of typical human face proportions as a region between 30-70% of face width and 60-90% of face height (a). The identified mouth region is subject to further analysis.

Lip key points (left and right lip corners and uppermost and lowermost points of lips) are searched for in a modified LAB color space. Contrast enhancement between lip and skin colors is achieved by subtracting the channel ‘A’ from the channel ‘B’ (b) as proposed in [23]. Then, the resultant image is thresholded using Otsu method [24] (c). Because different skin artifacts such as hyperpigmentation or spots can cause errors, blob detection is applied and the biggest region is left as belonging to mouth (d). Finally, the resultant binary image is scanned for terminal lip points (e).

The outer lip contour is extracted using dynamic programming. Dynamic programming enables to solve complex optimization problems by breaking them down into smaller subproblems and searching for optimal subsolutions. Because, lip contour extraction can be viewed as a search for an optimal path (in the sense of a cost) between two vertices in a 2D graph, dynamic programming approach can be applied to this task. This requires defining a cost function, transition rules and the corresponding 2D graph.

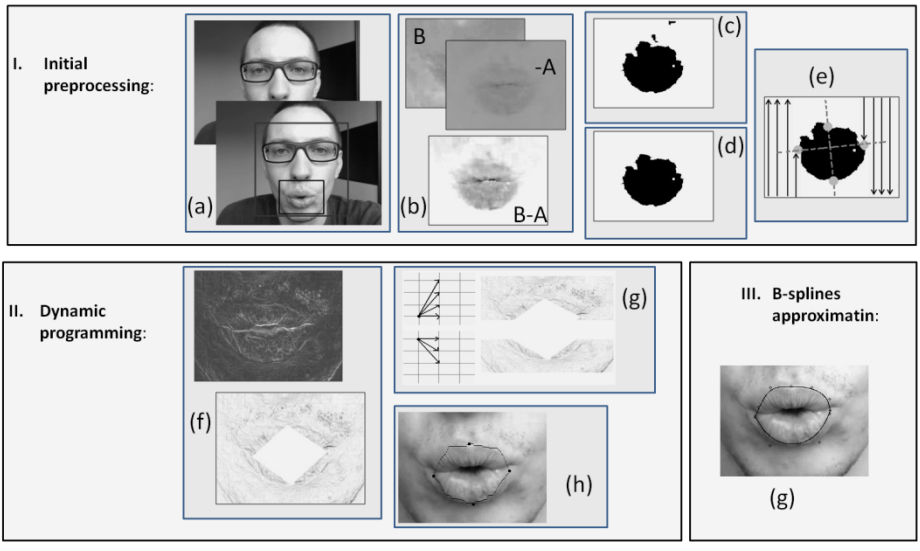


Fig. 2. The lip contour extraction procedure

The cost function to be minimized through dynamic programming is defined as:

$$c_k^{opt} = \min_{w(k-1)} \{c_{k-1}^{opt} + d[(i_{k-1}, j_{k-1}) \rightarrow (i_k, j_k)]\} \tag{1}$$

where: c_k^{opt} – is the optimal cost of reaching a node k, c_{k-1}^{opt} – the optimal cost of reaching a node k-1, $d[(i_{k-1}, j_{k-1}) \rightarrow (i_k, j_k)]$ – is a cost of transition from node k-1 to node k.

The transition cost consists of two components – a cost associated with branches of 2D graph structure and a cost of visiting a node. In the proposed solution the first component is not considered. Hence, the total cost of transition reduces to a node-visiting cost, which is defined as a value of a pixel of a modified gradient map (f). This map is created by inverting gradients and by masking-out regions with strong teeth-contours (g).

Although the outer lip contour, extracted separately for every quarter of lips using dynamic programming, is optimal in the sense of Bellman theory [25], it is not smooth. Therefore the final lip contour is obtained by B-spline approximation [26] of the dynamic programming results. Upper and lower lip contours are approximated separately. Experiments have shown that the best results can be achieved by modeling upper lip contour with 7 control points and the lower one with 4 control points. The 3rd order B-splines were used.

3.2 Building Feature Descriptor

Coordinates of B-spline control points that model lip shape represent initial feature descriptor. In the proposed procedure it consist of 22 elements. To remove

information redundancy and to reduce descriptor dimensionality, the Supervised Principle Component Analysis (S-PCA) is performed [27]. Contrary to the standard Principle Component Analysis (PCA) it lets to eliminate impact of within class scatter by involving a-priori knowledge on class affinity.

A matrix containing initial feature vectors for N observation is an input for S-PCA procedure. Firstly, standard PCA analysis is done for all samples. As a result eigenvectors $E = [e_0, e_1, \ldots, e_{N-1}]$ and corresponding eigenvalues $L = [l_0, l_1, \ldots, l_{N-1}]$ are calculated. Next, the standard PCA analysis is repeated for all considered visems separately. Contrary to the first step, where directions of the largest scatter for all visems are found, this step lets to identify directions of the highest within-class variability. Denoting $E_k = [e^k_0, e^k_1, \ldots, e^k_{N-1}]$ and $L_k = [l^k_0, l^k_1, \ldots, l^k_{N-1}]$ as eigenvectors and eigenvalues for a class k, correction weight of within-class scatter can be calculated as follows:

$$w_i = \sum_{j=1}^{N} \sum_{k=1}^{K} \left| l_j^k (e_i)^T e_j^k \right| \tag{2}$$

where: K– number of all classes, N – number of eigenvectors, l_j^k – i-th eigenvalue for class k, e_j^k – i-th eigenvector for class k.

These coefficients represent cumulative within-class variation in a direction of principal components found for all samples. By modifying original eigenvalues L according to:

$$l_i^{'} = \frac{l_i}{w_i} \tag{3}$$

the new feature space, built from directions featuring the highest between-class scatter only, is found, where: $l_j^{'}$ – i-th corrected eigenvalue, l_j – i-th eigenvalue calculated for all observations, w_j – i-th correction coefficient.

The presented procedure lets to reduce the feature vector to 9-10 elements (depends on a speaker) while preserving 90% of information associated with input data.

3.3 Liveness Verification

In the proposed method, an average visem identification rate (probabilities of false/correct assignment) for an individual speaker is used for defining liveness detection threshold. Given a probability estimate of visem correct assignment p_v, the probability of false assignment p_v' for a visem v is defined as:

$$p'_v = (1 - p_v) \tag{4}$$

An overall false assignment rate p' for a visem sequence will be then:

$$p' = \prod_{i=1}^{M} p'_v \tag{5}$$

where: M – total number of prompted visems.

The false assignment rate can be used as liveness pseudo score. The system knows what visems should be identified, as they are prompted. Consequently, the incorrect visem-identification rate in random syllable sequence can be measured. Finally, the liveness scores S can be calculated as:

$$S = \begin{cases} 1 \\ 0 \end{cases} if \quad \begin{matrix} p' <= E \\ p' > E \end{matrix} \tag{6}$$

where: E – is some assumed maximum classification error.

The higher classification error for an individual visem r'_v, the less its erroneous identification affects overall sequence recognition. Additionally, in order to avoid random acceptance in case of prompting the same syllable as replayed accidently, additional restriction is imposed. Number of correctly recognized visems must be at least a half of the sequence length. The final liveness decision T is defined as:

$$T = \begin{cases} 1 \\ 0 \end{cases} if \quad \begin{matrix} S = 1 \ \& \ l_e < L / 2 \\ otherwise \end{matrix} \tag{7}$$

where, l_e – number of incorrectly recognized visems, L – sequence length.

In the proposed approach, the prompted syllable is classified as correctly identified if expected visem is recognized at least twice in a syllable frame set.

The minimum length of the sentence results from the assumption, that classification error for a whole sentence should not be greater than E. For example, if average classification error equals 0.5, sequence of at least 7 syllables should be generated. Increasing a length of a sentence of appropriately selected recognized phonemes to be uttered exponentially increases correct sequence verification.

4 Experimental Evaluation

For experimental purposes an audio-visual database was recorded. Resolution of images was 960x720. Speakers were prompted to repeat 100 randomly generated sequences of syllables (10 syllables per sentence). Each syllable contains at least one out of seven visems selected as a basis for verification (these were: ‘A’, ‘M’, ‘U’, ‘O’, ‘W’, ‘S’, ‘SZ’). All recordings were manually labeled.

To validate the proposed approach two tests were performed. Firstly, the prepared database was used for identifying visems in single images. For classification purposes Ada Boost method was used (training set contains ca. 600 samples). Results for three speakers are presented in tables 1-4 and in figure 3. Average classification rate was 72.9% for speaker A, 80.8% for speaker B and 74% for speaker C.

Table 1. Isolated visem recognition results for three speakers (2nd and 3rd column for speaker A, 4th and 5th for speaker B and 6th and 7th for speaker C)

Visem	Train/Test Samples	Efficiency [%]	Train/Test Samples	Efficiency [%]	Train/Test Samples	Efficiency [%]
A	90/247	36.4	311/420	74.0	221/278	79.5
M	420/477	88.1	439529	83.0	235/275	85.5
O	131/343	38.2	308/397	77.6	94/145	64.8
S	125/193	64.8	169/252	67.1	44/90	48.9
U	611/619	98.7	406/493	82.4	304/328	92.7
W	144/210	68.6	245/297	82.5	57/149	38.3
SZ	251/339	74.0	425/460	92.4	83/135	61.5

Fig. 3. Results for outer lip contour extraction for recognized visems (from left: 'O','W','A','SZ','S','M','U')

Table 2. Confusion matrix for visem identification (Speaker A)

Visem	A	M	O	S	U	W	SZ
A	0.36	0.00	0.09	0.00	0.00	0.00	0.02
M	0.01	0.88	0.00	0.09	0.00	0.22	0.01
O	0.06	0.00	0.38	0.00	0.01	0.00	0.04
S	0.04	0.00	0.00	0.65	0.00	0.01	0.00
U	0.02	0.05	0.50	0.00	0.99	0.05	0.19
W	0.01	0.06	0.00	0.24	0.00	0.69	0.00
SZ	0.49	0.01	0.03	0.03	0.00	0.03	0.74

The second test was aimed at validation of liveness verification aspects of the proposed method. From the defined set of available syllables random sequences were generated and prompted. For half of the sequences, correct responses were presented to the system and for second half randomly generated ones. False acceptance rate (FAR) and false rejection rate (FRR) were calculated. Results are shown in table 5-7.

Table 3. Confusion matrix for visem identification (Speaker B)

Visem	A	M	O	S	U	W	SZ
A	0.74	0.02	0.19	0.02	0.01	0.01	0.02
M	0.00	0.83	0.00	0.00	0.00	0.00	0.00
O	0.10	0.00	0.78	0.00	0.14	0.00	0.05
S	0.03	0.04	0.00	0.67	0.00	0.15	0.00
U	0.01	0.06	0.02	0.17	0.82	0.00	0.00
W	0.02	0.05	0.00	0.13	0.03	0.82	0.01
SZ	0.10	0.00	0.02	0.01	0.00	0.00	0.92

Table 4. Confusion matrix for visem identification (Speaker C)

Visem	A	M	O	S	U	W	SZ
A	0.79	0.00	0.23	0.16	0.01	0.00	0.18
M	0.02	0.85	0.01	0.28	0.03	0.54	0.03
O	0.08	0.01	0.65	0.00	0.03	0.00	0.09
S	0.00	0.03	0.00	0.49	0.00	0.06	0.05
U	0.00	0.02	0.08	0.00	0.93	0.00	0.01
W	0.01	0.07	0.00	0.02	0.00	0.38	0.02
SZ	0.09	0.01	0.03	0.06	0.00	0.01	0.61

Table 5. Liveness verification results for Speaker A

Speaker	A	Rejected/All	Accepted/All
FRR [%]	15.6	7/45	-
FAR [%]	10.8	-	4/37

Table 6. Liveness verification results for Speaker B

Speaker	B	Rejected/All	Accepted/All
FRR [%]	0.0	0/42	-
FAR [%]	7.1	-	3/42

Table 7. Liveness verification results for Speaker C

Speaker	C	Rejected/All	Accepted/All
FRR [%]	4.2	2/48	-
FAR [%]	10.4	-	5/48

5 Conclusion

The lipreading procedure for liveness verification in video authentication systems was proposed in the paper. A novel approach of outer lip contour extraction, based on dynamic programming was presented and evaluated. Experiments performed for the

prepared database confirm reliability of the presented approach for liveness verification.

The main directions of further research will include efforts to improve robustness of image preprocessing stage (as poor illumination or skin artifacts can cause erroneous lip key point detection and, consequently, erroneous visem classification). Simultaneously, attempts will be taken to propose a new way of liveness detection for sequences that are built of whole words (although prompting for individual syllables has an advantage of better visem articulation, it can be considered as not user-friendly – uttering whole words is more natural).

References

1. Eveno, N., Besacier, L.: Co-Inertia Analysis for "Liveness" Test in Audio-Visual Biometrics. In: Proc. of the 4th Symposium on Image and Signal Processing and Analysis, pp. 257–261 (2005)
2. Chetty, G.: Biometric Liveness Detection Based on Cross Modal Fusion. In: 12th International Conference on Information Fusion, pp. 2255–2262 (2009)
3. Frischholz, R.W., Werner, A.: Avoiding Replay-Attacks in a Face Recognition System using Head Pose Estimation. In: Proc. of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures (2003)
4. Faraj, M.I., Bigun, J.: Synergy of Lip-Motion and Acoustic Features in Biometric Speech and Speaker Recognition. IEEE Transactions on Computers 56(9), 1169–1175 (2007)
5. Luettin, J., Thacker, N.A.: Speechreading using probabilistic models. Computer Vision and Image Understanding 65(2), 163–178 (1997)
6. Matthews, I., Cootes, T.F., Bangham, J.A., Cox, S., Harvey, R.: Extraction of visual features for lipreading. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2), 198–213 (2002)
7. Bregler, C., Konig, Y.: Eigenlips for robust speech recognition. In: Proc. IEEE Conf. Acoustics, Speech and Signal Processing, pp. 669–672 (1994)
8. Tomlinson, M.J., Russell, M.J., Brooke, N.M.: Integrating audio and visual information to provide highly robust speech recognition. In: Proc. Int. Conf. Acoustics, Speech and Signal Processing, vol. II, pp. 821–824 (1996)
9. Potamianos, G., Neti, C., Gravier, G., Garg, A., Senior, A.W.: Recent advances in the automatic recognition of audio-visual speech. Proc. IEEE 91(9), 1306–1326 (2003)
10. Foo, S.W., Lian, Y., Dong, L.: Recognition of visual speech elements using adaptively boosted hidden Markov models. IEEE Trans. Circuits Syst. Video Technol. 14(5), 693–705 (2004)
11. Chen, T.: Audiovisual speech processing. IEEE Signal Process. Mag. 18, 9–21 (2001)
12. Wang, S.L., Lau, W.H., Leung, S.H., Yan, H.: A real-time automatic lipreading system. In: Proc. 2004 Int. Symp. Circuits and Systems, vol. 2, pp. 101–104 (2004)
13. Perez, J.F.G., Frangi, A.F., Solano, E.L., Lukas, K.: Lip reading for robust speech recognition on embedded devices. In: Proc. Int. Conf. Acoustics, Speech and Signal Processing, vol. I, pp. 473–476 (2005)
14. Matthews, I., Cootes, T.F., Bangham, J.A., Cox, S., Harvey, R.: Extraction of visual features for lipreading. IEEE Trans. Pattern Anal. Mach. Intell. 24(2), 198–213 (2002)
15. Kass, M., Witkin, A., Terzopoulos, D.: Active contour models. Internaltional Journal of Computer Vision, 321–331 (1987)

16. Yuille, A., Hallinan, P., Cohen, D.: Feature extraction from faces using deformable templates. Int. J. Comput. Vision 8(2), 99–111 (1992)
17. Dupont, S., Luettin, J.: Audio-visual speech modeling for continuous speech recognition. IEEE Trans. Multimedia 2(3), 141–151 (2000)
18. Chiou, G.I., Hwang, J.-N.: Lipreading from color video. Trans. Image Processing 6, 1192–1195 (1997)
19. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
20. Aleksic, P.S., Williams, J.J., Wu, Z., Katsaggelos, A.K.: Audiovisual speech recognition using MPEG-4 compliant visual features. EURASIP J. Appl. Signal Process. 1213–1227 (2002)
21. Schapire, R.E.: The boosting approach to machine learning: An overview. In: Nonlinear Estimation and Classification. Springer, Heidelberg (2003)
22. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. Information Journal of Computer Vision 57(2), 137–154 (2004)
23. Nowak, H.: Lip-reading with discriminative deformable models. Machine Graphic and Vision International Journal 15 (2006)
24. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Sys., Man., Cyber. 9(1), 62–66 (1979)
25. Bellman, R.E., Dreyfus, S.E.: Applied dynamic programming. Princeton University Press (1971)
26. Lee, E.T.Y.: Comments on some B-spline algorithms. Computing 36(3), 229–238
27. Bair, E., Hastie, T., Paul, D., Tibshirani, R.: Prediction by supervised principal components. J. Amer. Statist. Assoc., 119–137 (2006)

Fuzzy Sliding Mode Control with Chattering Elimination for a Quadrotor Helicopter in Vertical Flight

S. Zeghlache[1], D. Saigaa[1], K. Kara[2], Abdelghani Harrag[1], and A. Bouguerra[1]

[1] LASS Laboratory, Department of Electronics, Faculty of Technology, University of M'Sila BP 166 Ichbilia 28000 Algeria
[2] Department of Electronics, Faculty of Engineering Sciences, University of Blida Algeria
{zeghlache_samir,saigaa_dj}@yahoo.fr

Abstract. The control of the helicopter includes nonlinearities, uncertainties and external perturbations that should be considered in the design of control laws. This paper presents a control strategy for quadrotor helicopter, based on the coupling of the fuzzy logic control and sliding mode control (SMC), using a nonlinear sliding surface. The main purpose of this work is to eliminate the chattering phenomenon. To achieve our purpose we have used a fuzzy logic control to generate the hitting control signal, the results of our simulations indicate that the control performance of the quadrotor are satisfactory and the proposed fuzzy sliding mode control (FSMC) can achieve favorable tracking performance.

Keywords: Sliding mode, Fuzzy Logic, Fuzzy Sliding Mode Control, quadrotor, Dynamic modeling.

1 Introduction

Autonomous Unmanned Air vehicles (UAV) are increasingly popular platforms, due to their use in military applications, traffic surveillance, environment exploration, structure inspection, mapping and aerial cinematography, in which risks to pilots are often high. Rotorcraft has an evident advantage over fixed-wing aircraft for various applications because of their vertical landing/take-off capability and payload. Among the rotorcraft, quadrotor helicopters can usually afford a larger payload than conventional helicopters due its four rotors. Moreover, small quadrotor helicopters possess a great maneuverability and are potentially simpler to manufacture. For these advantages, quadrotor helicopters have received much interest in UAV research [1].

The quadrotor is an underactuated system with six outputs and four inputs, and the states are highly coupled, Many efforts have been made to control quadrotor helicopter and some strategies have been developed to solve the path following problems for this type of system, First of this works the quadrotor has been controlled in 3 DOF such as the author in [2] take into account the gyroscopic effects and show that the classical model independent PD controller can stabilize asymptotically the attitude of the quadrotor aircraft. Moreover, they used a new Lyapunov function,

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 125–136, 2012.

which leads to an exponentially stabilizing controller based upon the PD2 and the compensation of coriolis and gyroscopic torques. While in [3] the authors develop a PID controller in order to stabilize altitude. In [4] a PID controller and a LQ controller were proposed to stabilize the attitude. The PID controller showed the ability to control the attitude in the presence of minor perturbation and the LQ controller provided average results. In [5] the authors the combination of the backstepping technique and a nonlinear robust PI controller. The integral action gain is nonlinear and based on a switching function that ensures a robust behaviour for the overall control law. In [6] they proposed the Backstepping Fuzzy Logic controller (BFL) and Backstepping Least Mean Square controller (BLMS) as new approaches to control the attitude stabilization of quadrotor UAV. And there are many works which control the quadrotor in 6 DOF, First of all, several backstepping and feedback linearization controllers have been developed. In [7] present the nonlinear control techniques applied to an autonomous micro helicopter type Quadrotor using the backstepping approach, In [8] presented the Backstepping Approach for Controlling a quadrotor Using Lagrange Form Dynamics In addition, two neural networks are introduced to estimate the aerodynamic components, one for aerodynamic forces and one for aerodynamic moments. In [9] a mixed robust feedback linearization with linear $GH\infty$ controller is applied to a nonlinear quadrotor unmanned aerial vehicle. In [10] the control strategy includes feedback linearization coupled with a PD controller for the translational subsystem and a backstepping-based PID nonlinear controller for the rotational subsystem of the quadrotor. And there is another non linear control technique applied to the quadrotor such as in [11] applied a robust adaptive-fuzzy control. This controller showed a good performance against sinusoidal wind disturbance. In [12] presented the comparison between a based model method and a fuzzy inference system to controlling a drone.

The sliding mode control has been applied extensively to control quadrotors. The advantage of this approach is its insensitivity of the model errors, parametric uncertainties, ability to globally stabilize the system and other disturbances [13]. In [14] author used the sliding mode approach to control a class of underactuated systems (quadrotor), In [15] the authors presents a continuous sliding mode control method based on feedback linearization applied to a Quadrotor UAV, In [7, 16] These papers present a new controller based on backstepping and sliding mode techniques for miniature quadrotor helicopter, In [1] presents two types of nonlinear controllers for an autonomous quadrotor helicopter. The first type is a feedback linearization controller that involves high-order derivative terms and turns out to be quite sensitive to sensor noise as well as modelling uncertainty. The second type involves a new approach to an adaptive sliding mode controller using input augmentation in order to account for the underactuated property of the helicopter. In the literature there are many works in the field of hybrid artificial intelligence systems such as [21, 22, 23]. Our contribution is based on the combination between the sliding mode and fuzzy logic technique (hybrid control low) using the nonlinear sliding surface in order to eliminate the charting phenomenon. Then, we present a control technique based on the development and the synthesis of a control algorithm based upon sliding mode to

ensure the locally asymptotic stability and the desired tracking trajectories expressed in terms of the center of mass coordinates along (X, Y, Z) axis and yaw angle, while the desired roll and the pitch angles are deduced unlike to [8]. Finally all synthesized control laws are highlighted by simulations which gave results considered to be satisfactory.

2 Quadrotors Dynamics Modeling

A quadrotor helicopter is a highly nonlinear, multivariable, strongly coupled, and underactuated system (six degrees of freedom (6 DOF) with only 4 actuators). The main forces and moments acting on the quadrotor are produced by propellers. There are two propellers in the system rotating in opposite direction to balance the total torque of the system. Changing the 2 and 4 propeller's speed conversely produces roll rotation coupled with lateral motion. Pitch rotation and the corresponding lateral motion; result from 1 and 3 propeller's speed conversely modified. Yaw rotation is more subtle, as it results from the difference in the counter-torque between each pair of propellers. the quadrotor configuration as shown in Fig. 1.

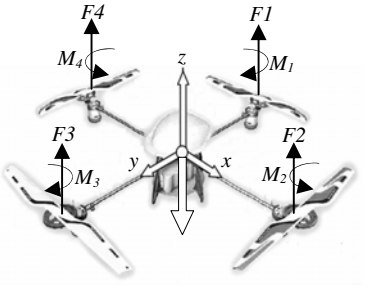


Fig. 1. Quadrotor configuration

The dynamic model presented in [7] includes the gyroscopic effects resulted from both the rigid body rotation in space and the four propulsion groups rotation that can be represented as:

$$\begin{cases} \ddot{\phi} = \frac{1}{I_x}\left\{\dot{\theta}\dot{\psi}\left(I_y - I_z\right) - J_r\overline{\Omega}\dot{\theta} + dU_2\right\} \\ \ddot{\theta} = \frac{1}{I_y}\left\{\dot{\phi}\dot{\psi}\left(I_z - I_x\right) - J_r\overline{\Omega}\dot{\theta} + dU_3\right\} \\ \ddot{\psi} = \frac{1}{I_z}\left\{\dot{\phi}\dot{\theta}\left(I_x - I_y\right) + U_4\right\} \\ \ddot{x} = \frac{1}{m}U_1\{\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\} \\ \ddot{y} = \frac{1}{m}U_1\{\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\} \\ \ddot{z} = \frac{1}{m}\{(\cos\phi\cos\theta)\,U_1\} - g \end{cases} \tag{1}$$

Where U_1, U_2, U_3 and U_4 are the control inputs of the system which are written according to the angular velocities of the four rotors as follows:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ b & 0 & -b & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{2}$$

And $$\overline{\Omega} = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{3}$$

And the parameters of the quadrotor used in dynamic modelling are given in Table 1.

Table 1. Physical parameters of the quadrotor

Symbol	Definition
$I_{x,y,z}$	body inertia
$\omega_{1,\ldots,4}$	rotor speed
ϕ	roll angle
θ	pitch angle
ψ	yaw angle
g	Gravitational acceleration
m	Total weight of the quadrotor
J_r	rotor inertia
b	thrust factor
d	drag factor

3 Rotor Dynamics

The rotor is a unit constituted by D.C-motor actuating a propeller via a reducer. The D.C-motor is governed by the following dynamic equations:

$$\begin{cases} V = ri + L\dfrac{di}{dt} + k_e\omega \\ k_m = J_r + C_s + k_r\omega^2 \end{cases} \tag{4}$$

The different parameters of the motor are defined such as:

V : Motor input — k_e, k_m : Electrical and mechanical torque constant respectively

k_r : Load constant torque — r : Motor internal resistance

J_r : Rotor inertia — C_s : Solid friction

Then the model chosen for the rotor is as follows

$$\dot{\omega}_i = bV_i - \beta_0 - \beta_1\omega_i - \beta_2\omega_i^2 \qquad i \in [1,4] \tag{5}$$

With $$\beta_0 = \frac{C_s}{J_r}, \beta_1 = \frac{k_e k_m}{rJ_r}, \beta_2 = \frac{k_r}{J_r} \text{ and } b = \frac{k_m}{rJ_r}$$

4 Control Strategy

To achieve a robust path following for the quadrotor helicopter, two techniques, capable to control the helicopter in presence of sustained external disturbances, parametric uncertainties and unmodelled dynamics, are combined. The proposed control strategy is based on the decentralized structure of the quadrotor helicopter system, which is composed of the dynamic Equation (1). The overall scheme of the control strategy is depicted in Fig. 2. The translational motion control is performed in two stages. In the first one, the helicopter height, z, is controlled and the total thrust, U_1, is the manipulated signal. In the second stage, the reference of pitch and roll angles (θ_r and ϕ_r, respectively) are generated through the two virtual inputs U_x and U_y, computed to follow the desired xy movement. Finally the rotation controller is used to stabilize the quadrotor under near quasi-stationary conditions with control inputs U_2, U_3, U_4.

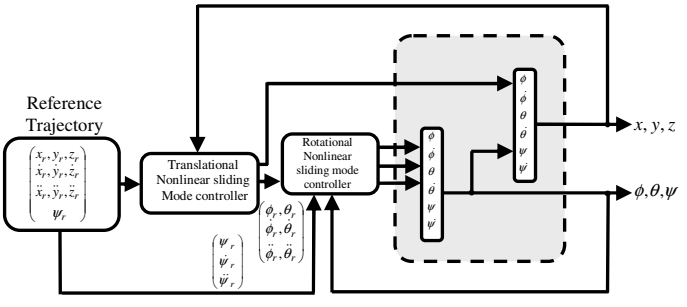


Fig. 2. Quadrotor helicopter control structure

5 Sliding Mode Control Using Non Linear Sliding Surface

In this section, the first goal in our topic is to characterize a class of manifold on which the control objective is achieved. We recall that the sliding mode control objective consist of first designing a suitable manifold $\psi(x,t) \in \Re^m$ defined by $\psi = \{x \in \Re^n / S(x) = 0\}$, to restrict state trajectories of the plant to this manifolds to result in the desired behaviors such as tracking, regulation, and stability then, determining a switching control low $u(x,t)$, which is able to drive the state trajectory to this manifolds and maintain it on this manifolds for all the time that is $u(x,t)$ is determined such that the selected manifold $\psi(x,t)$ is made attractive and invariant. Generally in sliding mode control, the switching surfaces are linear functions. Slotine and Li [17] gave a form of this sliding manifold which was a Hurwitz polynomial of the error and its derivative were up to $(r-1)$, where r is the relative degree of the output. For SMC using linear elements, the linear switching surfaces often yield a satisfactory systems response, in terms of stability and robustness, to the parameter variation and disturbances. However, with linear switching surface the speed of

transient response is relatively slow [18]. In this paper, the design and analysis of SMC with nonlinear switching surface are considered.

From the fact that the output $Y = [x, y, z, \phi, \theta, \psi]^T$, where the relative degree is two for each subsystem, in order order to obtain static feedback, we define the manifolds $\psi(e)$ as follows:

$$\psi(e) = \left\{ e \in \Re^n / S(e) = \dot{e} + \Lambda(e) = 0 \right\} \tag{6}$$

With $e = Y - Y_d$ is the tracking error, $\Lambda(.)$ is any given C^1 function whose property will be derived below. One has following result.

5.1 Proposition 1

Consider the manifolds $\psi(e)$ in (6) and assume that $\Lambda(.)$ is a continues function such that $e\,\Lambda(e) > 0,\ \forall e \neq 0$. Then on the manifolds ψ, the output error e converges at least asymptotically to zero.

Proof: Due to the manifolds ψ we have

$$\dot{e} = -\Lambda(e) \tag{7}$$

Let use the Lyaponov function given by $V = \frac{1}{2} e^2$. Its derivative is then

$$\dot{V} = -e\,\Lambda(e) \tag{8}$$

In order to make $\dot{V}$ negative definite, it is enough that $e\,\Lambda(e) > 0,\ \forall e \neq 0$. Hence, the outout error *e* is bounded and moreover, it tends asymptotically to zero.

Hence ψ are suitable manifolds for our control system, since the control objective is achieved on it. Let us now design the control low u that make, ψ attractive and invariant. The function $\Lambda(.)$ is given in (9) and characterized in proposition 1. Then, ψ is globally attractive and invariant.

$$\begin{cases} \Lambda(e) = \dfrac{2}{1 + e^{-\mu e}} - 1 \\ \dfrac{d\Lambda(e)}{de} = \dfrac{\mu}{2}\left[1 - \Lambda(e)^2\right] \end{cases} \qquad \mu > 0 \tag{9}$$

The desired roll and pitch angles in terms of errors between actual and desired speeds are, thus, separately given by:

$$\phi_r = \arcsin\left(U_x \sin\psi - U_y \cos\psi\right) \tag{10}$$

$$\theta_r = \arcsin\left(\frac{U_x}{\cos\phi\cos\psi} - \frac{\sin\phi\sin\psi}{\cos\phi\cos\psi}\right) \tag{11}$$

6 Fuzzy Sliding Mode Controller Design

The conventional sliding mode controller results in high frequency oscillations in its outputs, causing a problem known as chattering. The chattering is undesirable because it can excite the high frequency dynamics of the system. To eliminate chattering, a continuous fuzzy logic control $\Delta u_{fuzzy-slid}$ is used to approximate the discontinues control. The design of the fuzzy controller begins with extending the crisp sliding surface $S = 0$ to the fuzzy sliding surface defined by linguistic expression [18]:

$$\tilde{s} \text{ is zero} \tag{12}$$

Where $\tilde{s}$ is the linguistic variable for S and ZERO is one of its fuzzy sets. In order to partition the universe of discourse of S, the following fuzzy sets are introduced.

$$T(\tilde{s}) = \{NB, NM, ZE, PM, PB\} = \{F_s^1, ..., F_s^5\} \tag{13}$$

where $T(\tilde{s})$ is the term set of $\tilde{s}$, and NB, NM, ZR, PM, and PB are labels of fuzzy sets, which are negative big, negative medium, zero, positive medium, and positive big, respectively. For the control output $\Delta u_{fuzzy-slid}$, its term set and labels of the fuzzy sets are defined similarly by

$$T(\tilde{u}_s) = \{NB, NM, ZE, PM, PB\} = \{F_u^1, ..., F_u^5\} \tag{14}$$

The membership functions of these fuzzy sets are depicted in Fig. 3. In Fig. 3 (a), $r \in [0, 1]$ is a coefficient to be used to adjust the input centre point, and Φ is the defined boundary layer around the switch surface.

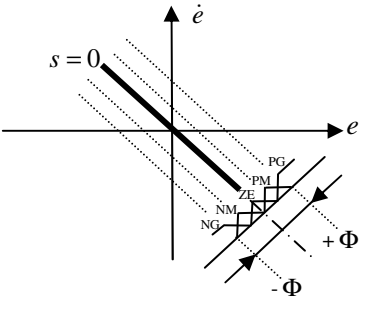


Fig. 3. Fuzzy partition of the space around the sliding surface

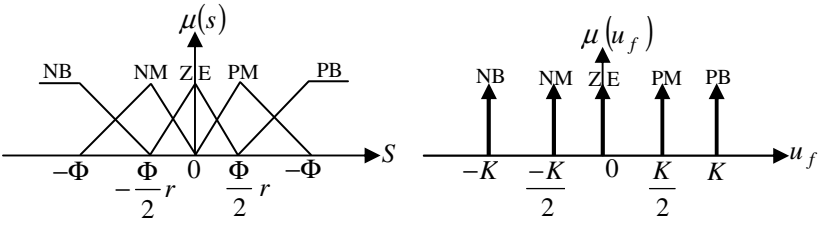


Fig. 4. Representation of term sets, $T(\tilde{s})$ and $T(\tilde{u}_s)$

From these two term sets, we can build the following fuzzy rules [19]:

R1 : IF s est NB Than $\Delta u_{fuzzy-slid}$ is PB R2 : IF s est NM Than $\Delta u_{fuzzy-slid}$ is PM

R3 : IF s est ZE Than $\Delta u_{fuzzy-slid}$ is ZE R4 : IF s est PM Than $\Delta u_{fuzzy-slid}$ is NM

R5 : IF s est PB Than $\Delta u_{fuzzy-slid}$ is NB

Once the membership functions and fuzzy rules are determined, the final step is the defuzzification, which is the procedure to determine a crisp control for $\Delta u_{fuzzy-slid}$. There are many defuzzification strategies such as the maximum criterion, the mean of maximum, the centre of area, and the weighted average method [18, 19]. We use the weighted average method to get the crisp control for $\Delta u_{fuzzy-slid}$. Then

$$\Delta u_{fuzzy-slid} = \frac{\sum_{i=1}^{5} C_{fi}\ \mu_i(s)}{\sum_{i=1}^{5} \mu_i(s)} \tag{15}$$

Where C_{fi} is the associated singleton membership function of $\Delta u_{fuzzy-slid}$.

7 The Proposed Fuzzy Sliding Mode Control of the Quadrotor

In this section, the objective is to apply the hybrid fuzzy sliding mode control in order to solve the problem of chattering phenomenon.

The control system applied to the quadrotor is given by:

$$u = u_{eq} + \Delta u_{fuzzy-slid} \tag{16}$$

With the discontinued control $\Delta u_{fuzzy-slid}$ is calculated by a fuzzy inference system, its description is already given in section 6.

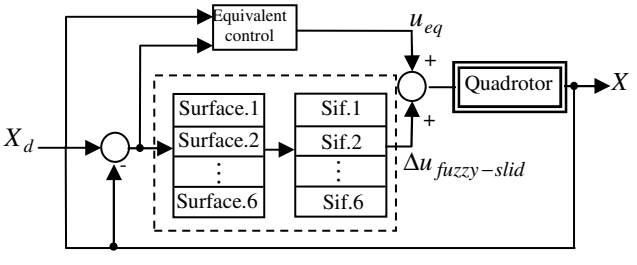


Fig. 5. Block diagram of the fuzzy-sliding control system applied to quadrotor

8 Simulation Results

This section presents the simulation results of the position control of quadrotor helicopter. The simulation results are presented in Fig.6,Fig.7,Fig.8,Fig.9, and Fig.10 we can see that, the controller ensures a good tracking performance, The control

efforts applied to the system are presented in the Fig.8 we remark the elimination of chattering problem permits the smoothness of the control law, The propellers speeds are given in Fig.9 where we can see their stabilization at the value 200 (radian/sec) in a finite time. Our approach (fuzzy sliding mode using non linear surface) presented better results in comparison to another works such as [10, 20].

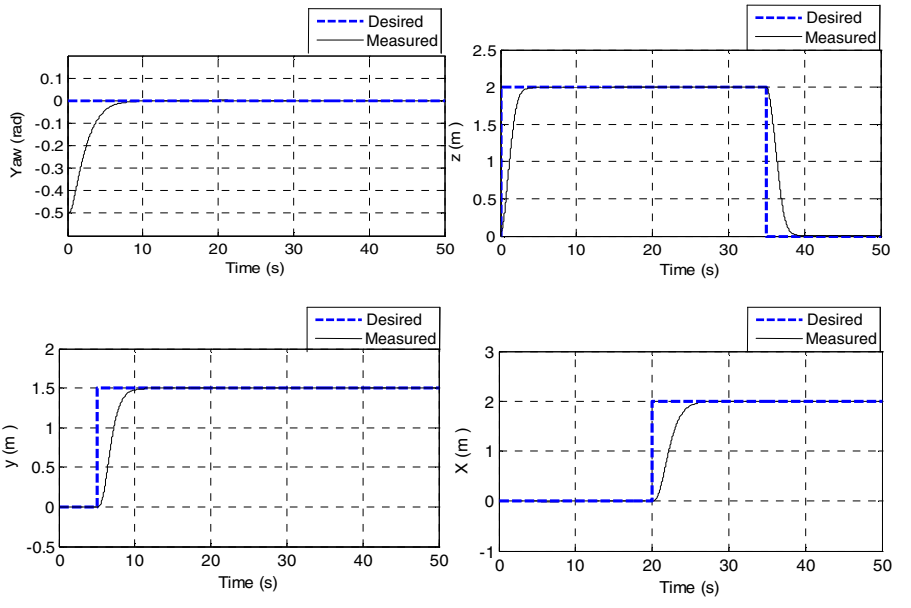


Fig. 6. Tracking simulation results of the desired trajectories along yaw angle (ψ) and the (X,Y,Z) axis using fuzzy Sliding Mode with nonlinear sliding surface

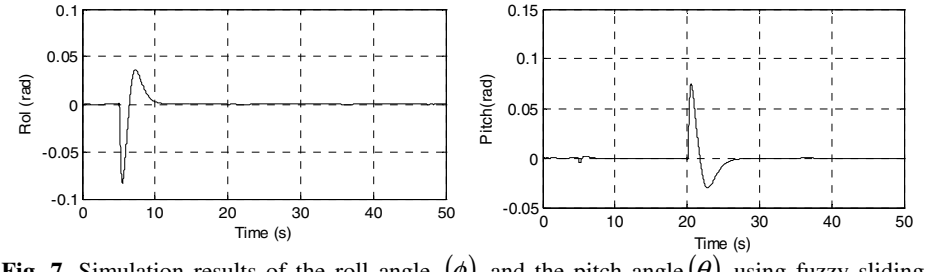


Fig. 7. Simulation results of the roll angle (ϕ) and the pitch angle (θ) using fuzzy sliding mode with nonlinear sliding surface

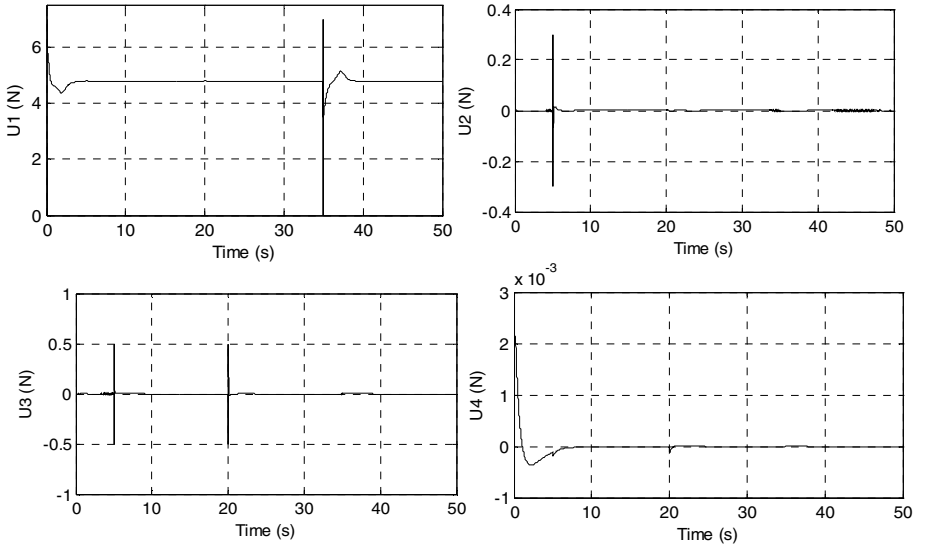


Fig. 8. Control response of a quadrotor helicopter using fuzzy sliding Mode with nonlinear sliding surface

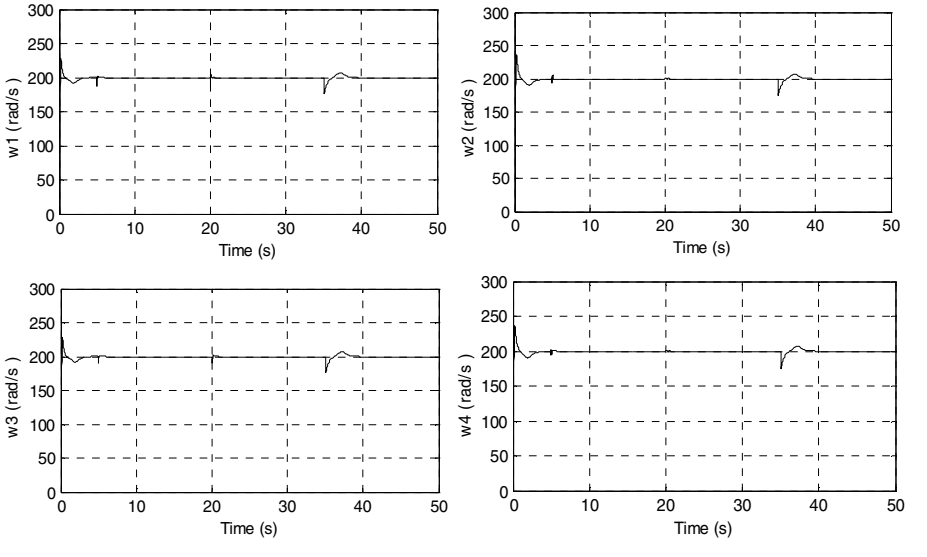


Fig. 9. Angular velocities of the four rotors using fuzzy sliding mode with nonlinear sliding surface

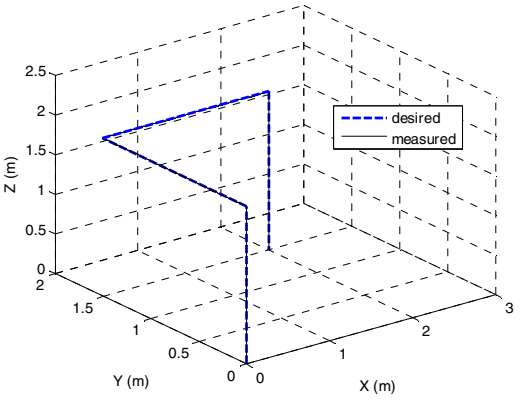


Fig. 10. Global trajectory of the quadrotor in 3D using fuzzy Sliding Mode

9 Conclusion

In this work, we addressed the position control problem of the helicopter's quadrotor taking into account the dynamics. A solution based on the fuzzy sliding mode control method using non linear sliding surface is proposed. This method is a combination of fuzzy logic control and the conventional sliding mode control, called fuzzy sliding mode control. This combination forces the real position towards the values required to achieve the control objective. Through the simulation results, we can see that fuzzy logic control can be applied to reduce the chattering of the sliding mode control.

References

1. Lee, D., Jin Kim, H., Sastry, S.: Feedback Linearization vs. Adaptive Sliding Mode Control for a Quadrotor Helicopter. International Journal of Control, Automation, and Systems 7(3), 419–428 (2009)
2. Tayebi, A., McGilvray, S.: Attidude Stabilisation of a VTOL Quadrotor Aircraft. IEEE Transactions on Control Systems Technology 14(3) (May 2006)
3. Derafa, L., Madani, T., Benallegue: Dynamic modelling and experimental identification of four rotor helicopter parameters. ICIT, Mumbai (2006)
4. Bouabdallah, S., Noth, A., Siegwart, R.: PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor. Autonomous Systems Laboratory Swiss Federal Institute of Technology, Lausanne, Switzerland (2004)
5. Bouchoucha, M., Tadjine, M., Tayebi, A., Müllhaupt, P.: Step by Step Robust Nonlinear PI for Attitude Stabilisation of a Four-Rotor Mini-Aircraft. In: 16th Mediterranean Conference on Control and Automation Congress Centre, Ajaccio, France, June 25-27 (2008)
6. AI-Younes, Y., Jarrah, M.: Attitude Stabilization of Quadrotor Uav Using Backstepping Fuzzy logic Backstepping Least-Mean-Square Controllers. In: Proceeding of the 5th International Symposium on Mechatronics and its Applications (ISMA 2008), Amman, Jordan, May 27-29 (2008)

7. Bouabdallah, S., Siegwart, R.: Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain (April 2005)
8. Das, A., Lewis, F., Subbarao, K.: Backstepping Approach for Controlling a Quadrotor Using Lagrange Form Dynamics. J. Intell. Robot. Syst. 56, 127–151 (2009)
9. Mokhtari, A., Benallegue, A., Daachi, B.: Robust Feedback Linearization and Gh∞ Controller for a Quadrotor Unmanned Aerial Vehicle. Journal of Electrical Engineering 57(1), 20 – 27 (2006)
10. Ahmad, A., Mian, W.D.: Modeling and Backstepping-based Nonlinear Control Strategy for a 6 DOF Quadrotor Helicopter. Chinese Journal of Aeronautics 21, 261–268 (2008)
11. Coza, C., Macnab, C.J.B.: A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization. Department of Electrical and Computer Engineering. University of Calgary, Calgary
12. Zemalache, K.M., Maaref, H.: Controlling a drone: Comparison between a based model method and a fuzzy inference system. Applied Soft Computing 9, 553–562 (2009)
13. Utkin, V.I.: Sliding Modes in Control and Optimization. Springer (1992)
14. Xu, R., Özgüner, Ü.: Sliding mode control of a class of underactuated systems. Automatica 44, 233–241 (2008)
15. Fang, Z., Zhi, Z., Jun, L., Jian, W.: Feedback Linearization and Continuous Sliding Mode Control for a Quadrotor UAV. In: Proceedings of the 27th Chinese Control Conference, Kunming, Yunnan, China, July 16-18 (2008)
16. Madani, T., Benallegue, A.: Backstepping Sliding Mode Control Applied to a Miniature Quadrotor Flying Robot. Laboratoire d'Ingénierie des Systèmes de Versailles 10-12, avenue de l'Europe, 78140 Vélizy – France
17. Yeganefar, N., Dambrine, M., Kokosy, A.: Stabilisation pratique par modes glissant pour un systeme linéaire à retard. In: Conférence Internationale D'automatique, Tunisia (2004) (in French)
18. Liu, J.Z., Zhao, W.J., Zhang, L.J.: Design of sliding mode controller based on fuzzy logic. In: Proceedings of the 3rd IEEE Conference on Machine Learning and Cybernetics, pp. 616–619. IEEE press, Shanghai (2004)
19. Kim, S.W., Lee, J.J.: Design of a fuzzy controller with fuzzy sliding surface. Fuzzy Sets and System 71(3), 359–367 (1995)
20. Mian, A.A., Wang, D.-B.: Dynamic modeling and nonlinear control strategy for an underactuated quad rotor rotorcraft. Journal of Zhejiang University Science A 9(4), 539–545 (2008)
21. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
22. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
23. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)

Ensemble of Binary Learners for Reliable Text Categorization with a Reject Option

Giuliano Armano[1], Camelia Chira[2], and Nima Hatami[1]

[1] Department of Electrical and Electronic Engineering
University of Cagliari
Piazza D'Armi, I-09123 Cagliari, Italy
{armano,nima.hatami}@diee.unica.it
[2] Department of Computer Science
Babes-Bolyai University
Kogalniceanu 1, Cluj-Napoca 400084, Romania
cchira@cs.ubbcluj.ro

Abstract. Text categorization is a key task in information retrieval and natural language processing. Providing a reliability measure of the classification result for a text document into a particular category can benefit the recognition rate as well as better inform the user with regard to the confidence that should be attributed to the output. A novel reliability measure is proposed starting from running different binary classifiers in the Error-Correcting Output Codes (ECOC) framework. Documents classified in a particular category which have a higher ECOC-computed distance from their classification in the next ranked category also have a higher associated reliability. This is the main idea explored in the proposed ECOC-based text classifier with a reject option. Experiments performed for some commonly used text categorization benchmark datasets demonstrate the potential of the proposed method.

Keywords: Classifier Ensemble, Text Categorization, Reliability, Error-Correcting Output Codes, Classification with a Reject Option.

1 Introduction

Information growth in the world wide web and digital form makes vital the development of automatic intelligent tools to manage massive amount of data. A large percentage of the information available on the web is stored as text data which needs to be analysed and processed in an efficient way.

Text classification (TC) is a typical information retrieval task with important real-world applications such as document indexing, routing and filtering, web page hierarchical categorization [1]. The problem refers to the classification of text documents into one or more predefined categories. A machine learning approach to this problem builds a classifier based on a training set of labeled documents. Several classifiers have been investigated in the literature for the text categorization task including neural networks, k-nearest neighbors, support vector machines, naive Bayes and multiple classifier systems [2,3].

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 137–146, 2012.

In some applications such as search engines, the prediction of web pages that match user-given keywords does not require to be fully precise since the user is involved in deciding which of the top ranked documents is the most relevant to his/her search. However, the reliability of a classification result is crucial in many other real-world applications. This latter issue is the focus of the current research.

Reliability measures in the context of text categorization can be used to improve the classification accuracy and inform the user (or human operator) on the level of decision confidence given by the automatic system. Research in this area is still in the early stages [4,5,6,7]. The methods proposed so far in the literature need further improving and more efficient reliability models are necessary.

In this paper, we propose an ensemble-based reliability measure for the text categorization task. To be more specific, the Error-Correcting Output Codes (ECOC) algorithm [8] is engaged to define the distance of a given document to a category. Comparing the distances of a given document from each category, we can asses the reliability of the candidate class as an output result. The classes with a reliability lower than a threshold suggest the user to follow another classification strategy. Computational experiments are performed for the Reuters benchmark text classification dataset and results are compared with that of related work.

The structure of the paper is as follows: Section 2 presents a brief review of related work on reliability measures for text classification, Section 3 describes the proposed method starting with a brief ECOC description, Section 4 present the numerical experiments and Section 5 contains the conclusions of the paper and directions for future research.

2 Related Work

This section focuses on related work devoted to reliability measures for text categorization. Given a classifier result stating that a document d belongs to a class c_i, how reliable is this decision and how can it be explored in obtaining better classification accuracies?

In [6], the concept of reliability is used in connection with the problem of literature-based discovery. The authors describe an association rule mining approach to determining relationships among scientific publications. Documents are represented as vectors of weights (the importance of a word in a document). A number of preprocessing techniques including n-gram representation of text (more specifically unigram and bigram), stemming and stopwords are considered. Additionally, a term weighting scheme for indicating the importance of a term in a document is employed. In this context, a reliability measure is defined to assess quality of the discovered patterns (docsets). The reliability measure proposed is based on a citation matrix built from citing and cited information available in a scientific publication database. A n-th order association citation matrix of size $m \times m$ (where m is the number of considered documents) is created to indicate the citation paths (of size n) from a document to another. Experiments are

performed for a dataset collected by the authors from the ACM Digital Library (more than 10000 articles considered). As the authors indicate, the document representation is shown to highly influence the reliability score while bigram scheme significantly outperforms the unigram one.

Fumera et al [4] investigate the potential of introducing a reject option for text categorization. To do this, the authors propose a strategy for deciding if the classifier result is reliable. Normally, the classifier computes a score called *posterior probability* $s_i \in [0,1]$ for a document d to belong to category c_i. A threshold τ_i is then used for each category to decide if document d should be assigned to category c_i i.e. if $s_i(d) \geq \tau_i$ then $d \in c_i$. Fumera et al observe that the classifier decision is highly reliable when the value of s_i is much higher than the threshold τ_i. Based on this observation, the reject option is simply implemented by using two thresholds (denoted by τ_{Hi} and τ_{Li}) instead of only one as follows: if $\tau_{Li} < s_i(d) < \tau_{Hi}$ then d is rejected (and later manually classified). Of course, if $s_i(d) \geq \tau_{Hi}$ then $d \in c_i$ and if $s_i(d) \leq \tau_{Li}$ then $d \notin c_i$. This work is extended in [5] to include a second stage in which the documents rejected in the first stage are automatically categorized using another classifier. Experiments on the Reuters 21578 benchmark dataset indicate a performance improvement with a reasonable number of rejected documents. It should be noted that in [4,5] the constraint is put on the maximum number of withheld decisions for each individual category.

Recently, the same authors - Pillai et al [7] - report the results for multi-label classification when the objective is to maximize the classification accuracy on the non-rejected documents with a constraint on the maximum number of rejected ones. A reliability measure is used to decide whether a document should be rejected or not. For each category, the Van Rijsbergen's F measure is used as a scalar performance measure based on the values of precision (the probability that a retrieved document is relevant to a given topic) and recall (the probability that a relevant document is retrieved). Considering multi-label classifiers, the global precision and recall over all categories are obtained by averaging (at micro or macro level) the class values. Correspondingly, the value of the F measure is obtained and denoted by $\hat{F}$. To decide if a document d has to be rejected, the authors define a reliability measure $R(d)$ based on maximizing $\hat{F}$ for the non-rejected documents. The idea is that the higher the value of $\hat{F}$ for all documents except d the less reliable is the classification of d. All documents d for which $R(d)$ falls below a prespecified rejection threshold are removed. Experiments are reported for two text categorization benchmark datasets (Reuters 21578 - ModApte and Heart Disease subset of the Ohsumed dataset) and one image annotation task (Scene dataset). The authors report the improvement of the classification accuracy (always increasing with the rejection rate) - particularly when about 30% of samples are rejected.

In [9], a reliability measure for Naive Bayes (NB) is proposed to address text classification problems which involve extremely asymmetric misclassification costs. The NB classifier is extended by modulating the score returned by NB based on the information-theoretic Kullback-Leibler (KL) divergence. The idea is to assess the confidence of NB decisions by measuring the difficulty of reversing

the NB result for a given input. Given a classifier result which says that a document d belongs to a certain class, the authors ask the question how much extra training with d is necessary to reverse the classifier outcome. The KL divergence is used to measure the effected change to the training distribution. The paper reports the improvement of results using NB-KL compared to the baseline NB for three benchmark datasets i.e. Reuters 21578, 20 Newsgroups and TREC-AP.

A meta-classifier approach to reliable classification is investigated in [10,11]. A conformity-based classifier is trained as a meta-classifier to predict the corectness of each document classification of a base classifier. For this purpose, the p-values of the meta-predictions are computed based on non-conformity functions and used in deciding which class output of the base classifier is the most reliable. In this way, the meta-classifier decides if the class predicted by the base classifier for an individual instance is a reliable result (otherwise, the classification is rejected). The estimation of p-values for each class can be however a difficult task in this approach.

Looking over the main existing related work highlights the fact that they mostly use a simple thresholding strategy on the posterior probability or evaluation measures such as F_1. Moreover, they normally rely on the opinion of a single expert in a rejection/acceptance decision, which obviously is not as reliable as an ensemble decision. However, the most important motivation of continuing research in this area is a low performance of existing systems in obtaining a good reliability. The method proposed in this paper tries to address these issues by consensus of a set of binary experts, forming an ECOC classifier. This strategy takes advantage of significant ensemble features e.g. redundancy and diversity to increase the reliability of decisions.

3 The ECOC Classifier with a Reject Option

The proposed reliability measure for text categorization takes advantage of the computed distances of a given document from any category in an ECOC decoding step to evaluate the confidence of label assignment. ECOC is a strategy to indirectly deal with a multi-class problem by hiring complementary binary classifiers, each focusing on different partitions (dichotomies) of the problem [8]. The main advantages of ECOC, which particularly proved to be efficient for large number of classes, are as follows: (i) possibility of using strong binary classifiers such as boosting and support vector machines (SVM) algorithms which can not directly address multi-class problems, (ii) generally, it is expected to be easier to address binary problems than multi-class problems with two many classes (divide and conquer principle), (iii) introduces redundancy for the same solutions so that if a classifier makes a mistake the final true label can still be recovered using information given by the other classifiers which have contributed to the same task (error-correcting property), and (iv) in datasets with small number of samples per class or imbalanced classes such as text data, the ECOC approach can lead to denser problems by merging different classes into a superclass which can *potentially* be better addressed by a *dichotomizer*.

Given a text classification problem with N_c categories, the main idea of ECOC is to create a codeword for each category. Arranging the codewords as rows of a matrix, a code matrix M is defined: $M \in \{-1, 0, +1\}^{N_c \times L}$ and L is the code length. From learning point of view, M specifies N_c categories to train L dichotomizers, $f_1 ... f_L$. A classifier f_l is trained according to the column $M(., l)$. If $M(N, l) = +1$ then all documents of category N are positive, if $M(N, l) = -1$ then all its documents are negative and, finally, if $M(N, l) = 0$ none of the documents of category N participate in the training of f_l. It is worth noting that the original ECOC proposed in [8] has only $-1, +1$ values to which later the 0 value as an uncertainty concept has been added [12].

Let us suppose a predicted codeword $\overline{y}_d = [y_1 ... y_L]$, $y_l \in \{-1, +1\}$ is a binary string assigned to document d (each bit representing the hard output of a dichotomizer). In the decoding step, the class output that maximizes the similarity measure S between $\overline{y}_d$ and row $M(N, .)$ is selected as follows:

$$Predicted\ Category(d) = ArgMax\ S(\overline{y}_d, M(N, .)) \tag{1}$$

Concerning the similarity measures, two of the most common techniques are the Hamming decoding distance (equation 2) where classifier outputs are hard decision and margin decoding (equation 3) where the outputs are soft level. These equations actually specify the distance of a document from any category - required in the proposed approach to define reliability.

$$S_H(\overline{y}_d, M(N, .)) = 0.5 \times \sum_{l=1}^{L} (1 + y_l \cdot M(N, l)) \tag{2}$$

$$S_M(\overline{y}_d, M(N, .)) = \sum_{l=1}^{L} y_l \cdot M(N, l) \tag{3}$$

The ECOC matrix codifies the class labels in order to achieve different partitions of classes considered by each dichotomizer. The main coding strategies can be divided into problem-independent (or fixed) [13,14] and problem-dependent [15,16,17,18].

The main idea of the proposed method is that a category assigned to a document is more reliable if the distance of predicted codeword to the candidate class is shorter than the distance to the second ranked candidate. For the ECOC-based text categorization system, the proposed reliability measure for the classification of a document d is defined as follows:

$$Reliability(d) = \frac{H_d(cw_2, \overline{y}_d) - H_d(cw_1, \overline{y}_d)}{H_d(cw_2, cw_1)} \times 100 \tag{4}$$

where cw_1 and cw_2 are the closest and second closest rows of the code matrix to the output vector $\overline{y}_d$ given by ECOC classifier for each test document and H_d is the Hamming distance between two codewords. A reliability threshold can be set on *Reliability* so that testing documents with reliability smaller than the threshold can be rejected.

Finally, the recognition rate and the rejection rate computed by the proposed method for text categorization with a reject option are defined as follows:

$$Recognition\ Rate = \frac{\psi_c}{\psi_a} \tag{5}$$

$$Rejection\ Rate = \frac{\psi_r}{\psi_n} \tag{6}$$

where ψ_c is the number of correctly labeled documents, ψ_a is the number of accepted documents which obtained the reliability score above the calculated threshold, ψ_r is the number of rejected documents and ψ_n represents the total number of documents. The threshold can be adjusted based on a trade-off between recognition rate and rejection rate. For example, for applications with low tolerance on errors such as those in information security, the threshold should be set higher and the error rate can be reduced at the cost of more rejections.

4 Numerical Experiments and Results

Reuters Corpus Volume I (RCV1) [19] is a benchmark dataset widely used in text categorization and in document retrieval. It consists of over 800,000 newswire stories, collected by the Reuters news and information agency. The stories have been manually coded using three orthogonal category sets. Therefore, category codes from three sets (Topics, Industries and Regions) are assigned to stories:

- Topic codes capture the major subject of a story. The hierarchy of topics consists of a set of 104 categories organized in a four-level hierarchy.
- Industry codes are assigned on the basis of the types of business discussed in the story and they include 365 different categories.
- Region codes include both geographic locations and economic/political groupings with 366 distinct region classes.

We pre-processed documents as proposed by Lewis et al. [19] and, in addition, we separated the training set and the testing set using the same split adopted in [19]. This process includes all necessary steps such as feature generation and feature selection/reduction as mentioned in [19] which results in 47,152 features. Documents published from August 20, 1996 to August 31, 1996 (document IDs 2286 to 26150) are included in the training set while documents published from September 1, 1996 to August 19, 1997 (document IDs 26151 to 810596) are considered for testing. The result is a split of the 804,414 documents into 23,149 training documents and 781,265 test documents. After multiple-label document removal, we have 150,765 documents (4,517 training documents and 146,248 testing documents).

In applications using text categorization as the core task, the computational efficiency is crucial because of the very large number of features, classes and samples. Therefore, the need for designing a simple and fast classification system is extremely important. There are many research studies using different

kinds of classifiers such as k-nearest neighbors (kNN), SVM, artificial neural networks (ANN), bayesian methods and rocchio classifiers [2]. However, in practice, most of them are not applicable since in real-world applications, e.g. search engines and recommender systems, a just-in-time response has great importance. Among them, NB and centroid classification algorithms are extremely simple and straightforward demonstrating competitive performance on text categorization problems. Moreover, they do not need to memorize a huge amount of training data as some other classifiers (e.g. kNN) or learn too many parameters (as opposed to ANN for example).

For the experiments presented in the current paper, we used *centroid-based classifiers* as the ECOC dichotomizers. This means that the prototype vector or centroid vector (μ_i^*) is computed for each supper-class $\mathcal{T}_i^*$ as:

$$\mu_i^* = \frac{1}{\mid \mathcal{T}_i^* \mid} \sum_{d \in \mathcal{T}_i^*} d \tag{7}$$

where $\mid \mathcal{T}_i^* \mid$ denotes the cardinality of set $\mathcal{T}_i^*$, i.e. the number of documents belonging to $* = +/-$ supper-class in the i-th dichotomy and d is a training document.

In the testing step, we calculate the similarity of a document d to each centroid by the *cosine* measure,

$$S(d, \mu_i^*) = \frac{d \cdot \mu_i^*}{\mid\mid d \mid\mid \ \mid\mid \mu_i^* \mid\mid} \tag{8}$$

This similarity can be regarded as the *posterior probability* of the dichotomizers and used for the i-th bit of the predicted codeword $\bar{y}_d$.

Numerical results emphasize the performance of the proposed method in discriminating between the reliable and unreliable decisions made for documents classified by ECOC algorithm. Documents with low *reliability score* (i.e. lower than the threshold τ) are rejected whereas those with higher *reliability score* are accepted. However, finding an optimal value for τ proved to be important and should be done by considering all factors of the classifiers. Among them, the recognition and rejection rates are the most effective ones and should be very well adjusted while designing a classifier with a reject option. False Rejection (FR) occurs either when a category is *correctly* assigned to a document with low reliability or when high reliability is given to a document with *incorrectly* predicted label. The results obtained for the Reuters dataset are presented in Table 1. The proposed method correctly rejects documents falsely predicted by classifiers, i.e. True Rejection (TR) ratio, whereas the FR ratio is very low considering the large number of test documents.

It is important to note that the FR and TR ratios are directly related to the threshold τ. Higher thresholds avoid the label assignment for false predicted documents while the FR increases. Therefore, in applications with high cost for mislabeling, the proposed strategy reduces the cost at the risk of rejecting some correctly labeled documents.

Table 1. Results obtained by the proposed method using an instance threshold on the selected subset of RCV1-v2 datasets. Accuracy boost is calculated comparing ECOC algorithm with the proposed rejected option versus standard ECOC (using the same instance matrix generated by Dense-random method).

Problem	Rejected documents	TR	FR	Accuracy boost (%)
Topics	97	87	10	3.8
Industry	88	79	9	3.7
Regions	90	83	7	1.3

We have compared the results of the proposed method with those of different commonly used TC algorithms [20,21] as well as the related reliability methods [4,5]. The standard TC methods used for comparison are the big-bang (global method), the flat method, Local Classifier per Node (LCN) and ECOC classifier with various matrices (without a reject option). For all these methods, centroid-based classifiers with the same parameters have been implemented. As shown in Table 2, the proposed method boosts the accuracy of the standard TC approaches using the ECOC algorithm and outperforms the related TC reliability methods.

Table 2. The recognition rate of the proposed method using different coding strategies compared to the existing standard text categorization methods on the selected subset of RCV1-v2 datasets (recognition rates are reported in percentage)

Problem	big-bang	LCN	ECOC D-Rand	ECOC S-Rand	ECOC BCH	Proposed Method (D-Rand)	Proposed Method (S-Rand)	Proposed Method (BCH)
Topics	34.5	34.1	35.5	34.4	35.9	**37.9**	37.0	36.9
Industry	38.3	36.7	38.1	39.1	38.9	40.0	**40.5**	39.9
Regions	38.0	37.0	37.5	36.8	37.3	**38.3**	38.2	38.0

5 Conclusions and Future Work

An efficient distance-based rule is introduced to evaluate the reliability of decisions made by ECOC text classifier for a given document. The proposed approach relies on the computed distances of an incoming document from each category given by the ECOC decoding step to make a final decision about the candidate category. A document will be assigned to a class with maximum posterior probability if its reliability score is higher than a predefined threshold. This way, by double checking the candidate category, the reliability of the decision made by ECOC classifier is increased. Experiments show capability of the method to boost the recognition rate with rejecting decisions about document class which have been assigned a low reliability.

Ongoing and future work focuses on adapting the proposed ECOC-based reliability to more complicated paradigms such as multi-label [22] or hierarchical TC problems since the application of ECOC classifier on these areas has not been explored. Investigating different strategies to formulate reliability measures based on binary classifiers opinion given by ECOC ensemble is another interesting research line.

Acknowledgments. Camelia Chira acknowledges the support of Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCS Romania.

References

1. Feldman, R., Sanger, J.: The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data, pp. I-XII, 1–410. Cambridge University Press (2007)
2. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1) (2002)
3. Hotho, A., Nurnberger, A., Paass, G.: A Brief Survey of Text Mining. LDV Forum 20(1), 19–62 (2005)
4. Fumera, G., Pillai, I., Roli, F.: Classification with reject option in text categorisation systems. In: Proc. 12th International Conference on Image Analysis and Processing, pp. 582–587. IEEE Computer Society (2003)
5. Fumera, G., Pillai, I., Roli, F.: A Two-Stage Classifier with Reject Option for Text Categorisation. In: Structural, Syntactic, and Statistical Patt. Rec., pp. 771–779 (2004)
6. Theeramunkong, T., Sriphaew, K.: Discovery of Relations among Scientific Articles using Association Rule Mining. In: Proceedings of the 2007 NSTDA Annual Conference Science (Science and Technology for National Productivity and Happiness), Thailand Science Park, Pathumthani, Thailand (2007)
7. Pillai, I., Fumera, G., Roli, F.: A Classification Approach with a Reject Option for Multi-label Problems. In: Maino, G., Foresti, G.L. (eds.) ICIAP 2011, Part I. LNCS, vol. 6978, pp. 98–107. Springer, Heidelberg (2011)
8. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error correcting output codes. J. of Artificial Intelligence Research 2, 263–286 (1995)
9. Kołcz, A., Chowdhury, A.: Improved Naive Bayes for Extremely Skewed Misclassification Costs. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 561–568. Springer, Heidelberg (2005)
10. Smirnov, E.N., Nalbantov, G.I., Kaptein, A.M.: Meta-conformity approach to reliable classification. Intell. Data Anal. 13(6), 901–915 (2009)
11. Kaptein, A.M.: Meta-Classifier Approaches to Reliable Text Classification, Master Thesis, Universiteit Maastricht, The Netherlands (2005)
12. Allwein, E.L., Shapire, R.E., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research 1, 113–141 (2000)
13. Hastie, T., Tibshirani, R.: Classification by pairwise grouping. The Annals of Stat. 26(5), 451–471 (1998)
14. Lin, S., Costello, D.J.: Error Control Coding, 2nd edn. Prentice-Hall, Inc. (2004)

15. Hatami, N.: Thinned-ECOC ensemble based on sequential code shrinking. Expert Systems with Applications 39(1) (2012)
16. Pujol, O., Radeva, P., Vitria, J.: Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. IEEE Transactions on PAMI 28(6), 1001–1007 (2006)
17. Pujol, O., Escalera, S., Radeva, P.: An incremental node embedding technique for error correcting output codes. Pattern Recognition 41, 713–725 (2008)
18. Zhou, J., Peng, H., Suen, C.Y.: Data-driven decomposition for multi-class classification. Pattern Recognition 41, 67–76 (2008)
19. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
20. Kosmopoulos, A., Gaussier, E., Paliouras, G., Aseervatham, S.: The ECIR 2010 Large Scale Hierarchical Classification, Workshop report (2010)
21. Silla Jr., C.N., Freitas, A.A.: A Survey of Hierarchical Classification Across Different Application Domains. Data Mining and Knowledge Discovery 20(1) (2010)
22. Armano, G., Chira, C., Hatami, N.: Error-Correcting Output Codes for Multi-label Text Categorization. In: Proceedings of 3rd Italian Information Retrieval Workshop (IIR 2012), Bari (in press, 2012)

Spontaneous Facial Expression Recognition: Automatic Aggression Detection

Ewa Piątkowska[1] and Jerzy Martyna[2]

[1] Institute of Applied Computer Science, Jagiellonian University, ul. Reymonta 4, 30-059 Cracow, Poland
[2] Institute of Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University, ul. Prof. S. Łojasiewicza 6, 30-348 Cracow, Poland

Abstract. The study presents results of analysis of spontaneous facial expression. Their purpose was to isolate aggression from the facial expressions. Based on tracking specific points of a face, selected from a video sequence, a trajectory of the face's movement was made. Then, using the Gabor filter and Local Binary Patterns (LBP) operator, extraction and analysis of the facial features was performed, from which vectors of aggression features have been detailed. Using the support vector machine (SVM) classifier, classification of the spontaneous facial data was made in order to detect the aggression. A correct recognition rate of the method, as high as 85% as well as a high ability for generalization was obtained.

Keywords: facial expression recognition, aggression detection, classification.

1 Introduction

In recent years, the research area of hybrid artificial intelligence systems has seen a remarkably active development [1, 7, 8]. Furthermore, there has been an enormous increase in the successful use of hybrid intelligent systems in many diverse areas such as robotics, medical diagnosis, computer vision, etc.

Computer vision is a widely developed field of science, the aim of which is to study the environment using data mining methods and extracting information from them. The main topics dealt with by the computer vision include recognition of persons, their movement and their faces. It is obvious that correctly operating computer vision systems are based on the vision systems (cameras, sensors, etc.) and programs that allow following the subject in the video sequences, estimating its movements, detecting facial features, etc.

One of the research areas developed within the computer vision is examination of human behaviour. It is a strong trend consisting of the analysis, not only of reading information transmitted by a person verbally, but also monitoring several non-verbal signals. This, primarily refers to facial expressions. The outcome of so extensive research operations carried out in this field was their direct translation into various aspects of everyday life, such as the systems intended to recognize and analyze facial expressions. These systems are increasingly widely used in such fields as:

– *medicine* - to support treatment of patients with emotional disorders.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 147–158, 2012.

- *monitoring* - in processing of data from cameras in order to detect aggression with the purpose to ensure security, checking degree of concentration or distraction of drivers or pilots, detecting moments when pain (patients) or panic (gathering of people) appears, etc.,
- *business, politics* - to assess a partner during negotiations, his or her reaction to articles, proposals presented, etc.,
- *education* - within an e-learning platform, to examine progress in assimilating knowledge and concentration of a student.

Facial Expression Recognition and Analysis (FERA), in particular the Facial Action Coding System Action Unit (FACS AU) recognition and discrete emotion detection [9], has been an active topic in computer science for some time now, and many promising methods have been published [10, 16]. The first survey of the field was reported in 1992 [17] and has been continued by several others [11,24,25]. However, none of these issues can track facial actions and recognize expressions over time in a monocular video sequence.

The main purpose of the article is to analyze spontaneous facial expressions, paying special attention to the detection of aggression. Using the method introduced here, facial representation was defined by means of a set of points, which are tracked in the video sequence. By using a special filter and developed texture coding, extraction facial features was made. Then, the SVM classifier was used to detect the aggression features in the examined face. Numerical tests confirmed the correctness of the approach in detecting aggression.

There were two kinds of research employed. Their main target was to elaborate a method, that could be used for the automatic detection of aggression in the spontaneous facial expressions. The other target of the study was to find a technique, that could be used to analyze facial features and the texture of people's faces in the video motion sequences. In both the first and second cases the problem was solved, by developing an appropriate method for the detection of aggression, and choosing the right technique for analyzing facial features and the textures of a face.

The concept adopted for facial representation and the tracking of selected points is presented in chapter 2. Chapter 3 is devoted to the extraction of features to the trajectory of particular points and changes in the texture, which correspond to facial changes due to aggression. In chapter 4, the classifier that was used here to detect aggression is described. Chapter 5 presents results of the numerical research. Final conclusions are presented in chapter 6.

2 Representation of the Face and Tracking Its Selected Points

The chapter below presents the applied method of facial representation and the monitoring of its selected points in the video sequence.

The Facial Action Coding System (FACS) [9] belongs among the most important facial expression coding methods, which also include spontaneous expressions. In this method facial expressions are described by means of Action Units (AUs). The AU constitutes the observable and indivisible facial movement, caused by the tensing or relaxing of a muscle or group of muscles. The authors of the FACS method have distinguished 46 independent AUs units. However, over 7000 combinations were isolated,

where these units can occur. In addition, the FACS system also determines the intensity of a particular facial expression on a scale from A to E, where A means trace (or minimum detectable) intensity, and E means the maximum intensity.

Fig. 1. Facial expressions corresponding to human aggression

According to FACS, four AUs describe aggression, namely:

1) AU4 lowering eyebrows,
2) AU5 lifting upper eyelids,
3) AU7 tightening of eyelids,
4) AU23 tightening of mouth.

Action Units that describe facial expressions corresponding to aggression are presented in Fig. 1.

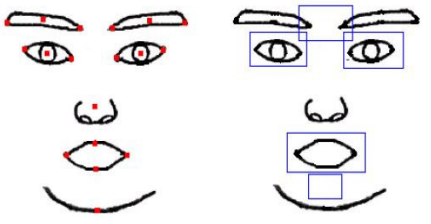
Systems of the facial expression analysis generally consist of 4 parts: face detection, face representation, feature extraction and classification. The proposed solution also retains the used 4-phase scheme.

Among the others for analyzing spontaneous expressions, was the system proposed by Bartlett [3], which was based on several methods of machine learning. It included a database, containing the facial expressions of over 100 persons. In this system the face was detected using the Viola and Jones algorithm [21], together with the GentleBoost method. Facial area was presented by means of the Gabor wavelets, of which the appropriate subset was the input set for the SVM classifier. The Bartlett system achieved an efficiency of 90.5%.

In the Zeng study [23], the data concerning the facial expressions obtained in real conditions were used. That data were marked using FACS system, and then a model given by Tao [19] was used to track the facial movements. For the classification the Support Vector Data Description method was used.

In the paper by Ioannou et al. [13] a system was created that would be able to adjust itself to the facial features specific to each user. The area containing the face was detected by means of SVM method, and then the areas of the mouth, eyes and eyebrows were located. The FAP method, defined within the MPEG-4 standard, was used to describe facial movement. In the Ioannan method an efficiency of 78% was achieved.

In the paper by Cohn et al. [6], attention was focused on the analysis of changes in facial appearance over time. The main objective of the study was smile detection, because this is the most frequent facial expression, accompanying the interpersonal communication. Changes in the facial appearance were monitored, based on corners at the corners of mouth. Strong linear dependence was observed between the amplitude of changes of the mouth corners at the mouth corners and the duration of the spontaneous expressions. For classification the Linear Discriminant Analysis (LDA) method was used, which accuracy of smile detection reaching 93% was obtained.



a) b)

Fig. 2. Facial points (a) and the related regions (b)

For the representation of the face the method was used, mentioned in the paper [18] was used, which consists of the determination of 18 points (Fig. 2a) and related regions (Fig. 2b). Selected points describe individual elements of a face, such as eyes, forehead, chin, and mouth. Additionally, along with these points the areas of a face are isolated, which can then be analyzed according to the changes in the texture.

For the object tracking in the video sequence (25 frames per second), an estimation of dynamics of the objects in time should be made, which will allow reproducing trajectory of such object. In the proposed solution, a particle filter [4] was used, also called the sequential Monte Carlo method.

Operation of a particle filter consists in the estimation of a non-linear dynamic system, in which the measuring error does not have Gaussian character. Filtering by means of this filter consists in determination of the density function of marginal distribution $p(x_t \mid y_{1:t})$, or the distribution calculated a posteriori on the basis of the observation history from the moment 1 up to t. Taking into account, that determination of the a priori and a posteriori distributions is connected with calculation of the high-dimensional integrals, dealing with both the linear and non-Gaussian systems, Monte Carlo methods are used in the particle filter.

The recursive Bayes filter is used as the particle filter, in which evaluation of the probability $p(x_t \mid y_{1:t})$ distribution is carried out through a set of the weighted samples:

$$s = \{(s^n, \pi^n) \mid n = 1, \ldots, N\} \tag{1}$$

Each sample represents a hypothesis of the object's state in time, together with the corresponding discrete π probability, where $\sum_{n=1}^{N} \pi^n = 1$. In each step (frame) N samples are taken at random with the probability

$$\pi^n = p(y_t \mid X_t = s_t^n) \tag{2}$$

The averaging state of the object is given according to the formula:

$$E[S] = \sum_{n=1}^{N} \pi^n s^n \tag{3}$$

In order to locate the objects in each frame of the video sequence, it is necessary to define representation of the object by means of some characteristic features. Here, it was assumed that color is such a characteristic feature.

Using a colour in the partial filter requires that each x_i pixel belonging to the object is assigned to the appropriate range in the histogram by means of the $h(x_i)$ function. Then, the colour variable of the object being tracked is analyzed.

Due to the high sensitivity of the colour variable distribution to the changes in illumination, the Hue Saturation Value (HSV) model was used here, in which it was assumed, that the histogram consists of $8 \times 8 \times 4$ ranges, assigning the smaller stress to the component V. In addition, it was assumed that higher weights were assigned to the colours from the area of the central object, while colours of pixels in marginal areas were considered less significant. This helped to avoid false readings connected with probability of the object's contours. The weights are assigned according to the distance from the centre of an ellipse circumscribing the object, namely:

$$k(r) = \begin{cases} 1 - r^2, & r < 1 \\ 0, & r \geq 1 \end{cases} \tag{4}$$

Then, the distribution $\{p_j^n\}_{u=1,\ldots,m}$ of the $\{p_j^n\}_{u=1,\ldots,m}$ colour variable for pixels belonging to the object is calculated according to the formula:

$$p_j^u = f \sum_{i=1}^{I} k\left(\frac{\| f - x_i \|}{a}\right) \delta(h(x_i) - u) \tag{5}$$

where m is the number of ranges in the histogram, I is the number of pixels in the object, δ is the Kronecker delta. Parameter a is responsible for adaptation of the size of the ellipse circumscribing the object and is defined as follows:

$$a = \sqrt{H_x^2 + H_y^2} \tag{6}$$

In addition, in order to fulfill the condition $\sum_{u=1}^{m} p_j^u = 1$, a normalizing parameter f equal to $1/\sum_{i=1}^{I} k(\frac{\|j - x_i\|}{a})$ was applied.

In the object-tracking problem, the object's state is evaluated for each frame of the sequence on the basis of new observations. This is why, the so-called measure of similarity between the objects with the given representation must necessarily be defined. Here, the Bhattacharyya coefficient ρ was applied in order to compare two discrete distributions $\{p_j^u\}_{u=1,\ldots,m}$ and $\{q_j^u\}_{u=1,\ldots,m}$, in the following way:

$$\rho[p, q] = \int \sqrt{p^u q^u} \tag{7}$$

Each sample (particle) is defined by vector $s = \{x, y, \dot{x}, \dot{y}, H_x, H_y, \dot{a}\}$, where (x, y) are the position of the ellipse describing the object, $(\dot{x}, \dot{y})$ describes movement, (H_x, H_y) are the lengths of the big and small semi-axes, while $\dot{a}$ is the parameter of change of the object's size over time.

The algorithm for tracking the object in the form of the distribution of the colour variable q for a given set of samples s has the form:

1) Sampling of N samples from the set S_{t-1} with the probability π_{t-1}^n.
2) Propagation of each sample through $s_t^n = As_{t-1}^n + w_{t-1}^n$, where A is the matrix of transition, which defines the deterministic model of the dynamics, w_{t-1} is a multidimensional random variable G of the distributions,
3) Observation of the colour variables: a) calculation of colour distribution p_s^n for each sample from the set S_t, b) calculation of Bhattacharyya coefficient (distance) between the object model q and the hypothesis p_s^n, c) assigning weights for each

sample with the changing values of the Bhattacharyya coefficient, which means that the most significant are the samples with the highest similarity to the model of the object q,

4) Assessment of the averaging state of the set S_t, namely $E(S_t) = \sum_{n=1}^{N} \pi_t^n s_t^n$.

3 Extraction of Features

For the extraction of features connected with the trajectory of individual points on the face and changes in expression, the Gabor filter and LBP operator were used.

In the two-dimensional area the Gabor filter is defined as the Gaussian function modulated by the sine and cosine waves:

$$\Psi_G(x, y; f_{0G}, \theta_G) = \frac{f_{0G}^2}{\pi \cdot \gamma_G \cdot \eta_G} \cdot e^A \cdot e^B \tag{8}$$

where $A = -(\frac{f_{0G}^2}{\gamma_G^2} \cdot x'^2 + \frac{f_{0G}^2}{\gamma_G^2} \cdot y'^2)$, $B = 2\pi j \cdot f_{0G} \cdot x'$, $x' = x \cdot \cos\theta_G + y \cdot \sin\theta_G$, $y' = -x \cdot \sin\theta_G + y \cdot \cos\theta_G$.

Parameters of the Gabor filter are the sharpness along the longer axis of γ_G and shorter axis of η_G, filter central, frequency of filter f_{0G} and angle of rotation θ_G of the main filter axis.

In order to analyze the texture, convolution of the input image is conducted with the Gabor filter kit. From the representation thus obtained, a histogram is calculated, which is the vector of features.

The Local Binary Patterns (LBP) method, proposed originally by Ojala [14,15] and later extended by Ahonen [2] and by Hadid [12], allows to transforming the image into a representation, thanks to the application of the special operator, which assigns the value on the basis of a value of the particular pixel P. The LBP operator is given by

$$\forall_{n \in N} t(n) = \begin{cases} 1, & n \leq P \\ 0, & n > P \end{cases} \tag{9}$$

where N is the number of pixels in the neighbourhood. Values of pixels from the neighbourhood, making a binary sequence, constitute a code, which when transformed into the decimal system is assigned to a pixel.

The classical LBP operator has analyzed the neighbourhood with dimensions 3×3, yet the relatively small size of the operator was its basic limitation. For the purposes of the features extraction, a new kind of LBP operator was adopted here as well as new dimensions of the neighbourhood for the operator, e.g., a circular neighborhood with radius R and any number of pixels P. As a result of LBP transformation carried out in this way, binary standards are coding local primitives of texture, also called the micro-patterns or textons. Examples of such micro-patterns are as follows: spot, spot/flat, line end, edge, corner.

An LBP operator with R radius and a number of pixels P can code the image by means of $2P$ different values (codes). The image after processing into the LBP representation $lbp(x, y)$ is analyzed by means of a histogram, which can be defined as follows:

$$H_i = \sum_{x,y}^{I}(lbp(x,y) = i), \quad i = 0, \ldots, n-1 \tag{10}$$

where n is a number of possible values of LBP codes. Function $I(.)$ returns 1, when expression $(.)$ is true, otherwise it returns 0. The obtained histogram describes the statistical distribution of individual standards.

4 The SVM Classifier

In the case of the separable binary classification problem, the goal of support vector machine (SVM) is to find a particular hyperplane for which the separated margin is maximized. The optimal hyperplane is defined [20] as follows

$$g(\mathbf{x}) = \mathbf{w}_0^T \mathbf{x} + \mathbf{b}_0 \tag{11}$$

where $\mathbf{w}_0$ denotes the optimum value of the weight vector, $\mathbf{b}_0$ is the optimum value of the bias.

This maximization of the margin between classes is equivalent to minimizing the Euclidean norm of vector $\mathbf{w}$. Thus, the optimal hyperplane satisfies the constraints:

$$\begin{cases} d_i(\mathbf{w}_i^T + \mathbf{b}) \geq 1, \quad i = 1, \ldots, N \\ \Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \end{cases} \tag{12}$$

In order to solve Eq. (11), we can use the Lagrangian method, which is known as a primal problem

$$J(\mathbf{w}, \mathbf{b}, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i \left[d_i(\mathbf{w}^T\mathbf{x}_i + \mathbf{b}) - \mathbf{1} \right] \tag{13}$$

where the auxiliary variable α_i is referred to as the Lagrange multiplier and its value is positive. The saddle point of the Lagrangian function $J(\mathbf{w}, \mathbf{b}, \alpha)$ provides a solution to the optimization problem. Then, we obtain the two conditions:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x}_i \tag{14}$$

$$\sum_{i=1}^{N} \alpha_i d_i = 0 \tag{15}$$

From the above conditions we can receive the dual form of our problem, namely

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T x_j \tag{16}$$

The solution to the objective function is provided here by

$$\sum_{i=1}^{N} \alpha_i d_i = 0 \tag{17}$$

$$\alpha_i \geq 0 \text{ for } i = 1, 2, \ldots, N \tag{18}$$

(a) (b)

Fig. 3. Face described as a set of 18 points (a) and the related regions (b)

The optimum values of the weight vector $\mathbf{w}_0$ and bias $\mathbf{b}_0$ can be obtained as follows

$$\mathbf{w}_0 = \sum_{i=1}^{N} \alpha_{0,i} d_i \mathbf{x}_i \tag{19}$$

$$\mathbf{b}_0 = 1 - \mathbf{w}_0^T \mathbf{x}^{(s)} \quad \text{for} \quad d^{(s)} = 1 \tag{20}$$

5 The Experimental Results

In our analysis, we have used data contained in the FEED [22] base. This base was prepared in the framework of the FG-NET project at the University of Munich. The basic assumption in creation of the base was the recording of the most natural behaviours of people. For this purpose, films were shown in order to provoke emotions. The collected data may not have been fully spontaneous, as the participants were aware of what kind of a test they were taking part in. However, the demonstrated emotions are to significant degree close to the natural ones.

The basis consists of video sequences acquired during recording of 18 persons of whom 9 were women and 9 men. The data are stored in the form of MPGE4 files with a frame-rate equal to 25 frames/second, dimensions 320×240 and 8-bit depth of colour. For each person, 3 samples were gathered, where the first two make the teaching set, while the third serves for testing. Presented emotions were labeled in accordance with the theory of six universal emotions. The base also contains sequences, in which persons keep a neutral face, i.e., they do not express emotions.

The proposed method was implemented in the environment of MATLAB using the Image Processing and Bioinformatics Toolboxes. The made out algorithm consists of the following stages:

1) matching of the model in the first model frame,
2) tracking the model points in each frame,
3) definition of the feature vector, extraction of texture and analysis of the point movement trajectory,
4) classification, or detection of aggression.

(a) (b)

Fig. 4. Change of location of mouth corners (a) and measured position of the eyes, forehead, chin and mouth regions (b)

The face was described using a set of 18 points (Fig. 3a). Location of the face model in the sequence is determined in the first frame of sequence. Coordinates of points are stored in the file. Next, the areas defined around the face model (Fig. 3b) are tracked by means of the particle filtering algorithm, described in section 2. Points of the model have been circumscribed by ellipse, the area of which is then represented by histogram of the colour variable.

At this stage, the fundamental problem is head and body movements during presentation of the emotions. They lead to corruptions in estimating of the movement of the model points. Hence, the movement of such points as the ends of the eyebrow and corners of the mouth is defined by means of the distance from the established reference points.

In the case of the eyebrow movement such a point is the inner corner of the eye, on the left and right side, respectively. On the other hand, movement of the mouth corners is measured by reference to the tip of the nose (Fig. 4a).

In addition, the detected movements of the head can bring significant information for the description of expression. Translation, or the horizontal or vertical displacement of the head is calculated on the basis of the nose's position in every frame. Rotation, however, is detected through analysis of the eyes' axes.

At the stage of extraction and analysis of features describing the emotion on the basis of the detected face points trajectory in the video sequence, we can determine the composition of a vector of features, which include:

a) Distance of the mouth corners from the nose and the distance of the eyebrow ends from the inner corner of the eyes. These values are placed in each frame of the sequence in relation to the first frame, presenting a neutral face. In addition, these distances are presented as a percentage, in order to make the movement model more adjusted.
b) Head movement - change of location of the nose center in each frame, also as a percentage.

In order to maintain a constant value of the features vector, the video sequences have been normalized according to the number of frames. As a result of this, the algorithm operates on the sequences consisting of 100 frames.

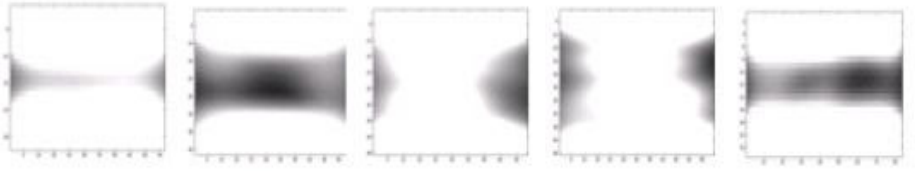


Fig. 5. Face regions after the Gabor method filtering. From left side are face regions of chin, forehead, right eye, left eye, mouth.

On the basis of the face model location in the particular frames we have assessed the positions of the eyes, forehead, chin and mouth regions (Fig. 4b), which are then analysed according to the changes in the texture. Particular regions have been subjected to the process of normalization, in the effect of which is that the areas of the eyes have been rescaled to the size of 40×35, forehead 30×30, mouth 60×25 and chin 40×20.

In the proposed approach, the texture is analyzed by means of two methods: Gabor filter and LBP operator. First, the areas are transformed to the appropriate representation, and then they are described in the form of a histogram. The histograms for each area are connected into one vector of features.

Filtering using the Gabor method meant that filter banks with 8 orientations and 9 rates were used (Fig. 5). A a result of this method, the vector of features obtained consists of 432000 elements. In the case of the LBP operator, the basic operation (3×3) version was used. The number of the histogram ranges was 256, and the size of each vector of features was 1280. The last element of the algorithm is the SVM classifier, made available at the Bioinformatics Toolbox. As the input set, various combinations of the vector of features were fed:

- geometrical features in time,
- histogram of texture obtained by means of the Gabor filters,
- histogram of texture obtained by means of the LBP operator,
- concatenation of geometrical features and histograms of the texture (for both methods).

The system was trained using 10 video sequences, demonstrating aggression and 10 sequences expressing other emotions. The set of tests contained 4 examples, 2 positive cases (aggression) and 2 negative cases (no aggression). In addition, the ability of the method for generalization was checked, using recordings of the persons, who have not been taken into account in the process of learning.

Effectiveness in detecting the expression is measured by the correctness recognition coefficient of the received results, namely

$$R = \frac{\text{number of correctly recognized examples}}{\text{number of all examples in the test}} \times 100\% \tag{21}$$

Presented results are in conformity with the tested configurations of the vector of features. For the analysis of the face points movements trajectory, a vector was defined, with dimensions of 1000 features, containing the information on the change in the face

Table 1. The overall results of the classification in different configurations of features

Vector of features in time	Number of features	Correctness recognition coefficient
Geometrical features	1000	78%
Coding the texture with Gabor filter	432 000	81%
Coding the texture with LBP method	1280	72%
Geometrical features in time plus Gabor filter	433 000	80%
Geometrical features in time plus LBP method	2 280	85%

geometry in the determined time slice (frames). The correctness recognition coefficient obtained here is equal to 78%.

The texture of the face was acquired from the most representative frame of the sequence, where the presented emotion is in the phase of culmination. When coding the texture with Gabor filters the system achieves a correctness recognition coefficient of 81%, while when using the LBP method 72%. Combining the two categories, or the methods describing both the geometry (in the dynamic take), and the face texture, the tested examples are classified with a correctness recognition coefficient of 82.5%. The overall results of the classification in different configurations of the vector of features are presented in Table 1.

The ability of the system for generalization was also tested, which here means here the detection of aggression in examples that were not included in the test set (new persons). In this case the vector of features was used which consisted of the geometric-dynamic features as well as the LBP histograms. Two samples (negative and positive) were presented from four different persons. The system correctly classified all of them.

6 Conclusions

The conducted experiment demonstrated, that automatic detection of aggression in people, from a given video sequence, containing spontaneous facial reactions was possible. The best results for the analysis of a human face texture were obtained using the Gabor filter (81% of correctness recognition coefficient). Results of the classification for the texture analysis prove, that a relatively large amount of information is coded in the face's texture, and in particular in the analyzed regions of the eyes, forehead, mouth and chin. It is for this reason that the texture of the face was combined during the experiment with the dynamics and geometry. Here, the task of classification becomes much more difficult. Using the Gabor filter gave much worse results than the LBP operator.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Recognition with Local Binary Patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)

3. Bartlett, M., Littlewort, G., Lainscsek, C., Fasel, I., Movellan, J.: Machine Learning Methods for Fully Automatic Recognition of Facial Expressions and Facial Actions. In: IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 592–597 (2004)
4. Brzozowska, K., Dawidowicz, A.L.: Partial Filter Method. Applied Mathematics 10, 69–107 (2009)
5. Cohen, I., Sebe, N., Garg, A., Chen, L., Huang, T.: Facial Expression Recognition from Video Sequences: Temporal and Static Modeling. Computer Vision and Image Understanding 91(1-2), 160–187 (2003)
6. Cohn, J.F., Schmidt, K.L.: The Timing of Facial Motion in Posed and Spontaneous Smiles. Int. Journal of Wavelets, Multiresolution and Information Processing 2, 1–12 (2004)
7. Corchado, E., Abraham, A., Carvalho, A.: Hybrid Intelligent Algorithms and Applications. Information Sciences 180(14), 2633–2634 (2010)
8. Corchado, E., Graña, M., Wozniak, M.: New Trends and Applications on Hybrid Artificial Intelligence Systems. Neurocomputing 75(1), 61–63 (2012)
9. Ekman, P., Friesen, W.: Facial Action Coding Systems: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto (1978)
10. Essa, I., Pentland, A.: Coding, Analysis, Interpretation, and Recognition of Facial Expressions. IEEE Trans. on PAMI 19, 757–763 (1997)
11. Fasel, B., Luettin, J.: Automatic Facial Expression Analysis: A Survey. Pattern Recognition 36(1), 259–275 (2003)
12. Hadid, A., Pietikainen, M., Ahonen, T.: A Discriminative Feature Space for Detecting and Recognizing Faces. In: Proc. Computer Vision and Pattern Recognition, pp. 797–804 (2004)
13. Ioannou, S., Raoouzaiou, A., Tzouvaras, V., Mailis, T., Kaollias, K.: Emotion Recognition Through Facial Expression Analysis Based on a Neurofuzzy Method. Neural Networks 18, 423–435 (2005)
14. Ojala, T., Pietikainen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Featured Distribution. Pattern Recognition 29(1), 51–59 (1996)
15. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution Gray-scale and Rotation Invariant Texture with Local Binary Patterns. IEEE Trans. on Pattern Analysis and Machine Intelligence 7(7), 971–987 (2002)
16. Pantic, M., Rothkrantz, L.: Automatic Analysis of Facial Expressions: The State of the Art. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(12), 1424–1445 (2000)
17. Samal, A., Iyengar, P.A.: Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey. Pattern Recognition 25(1), 65–77 (1992)
18. Sohail, A.S.M., Bhattacharya, P.: Detection of Facial Feature Points Using Anthropometric Face Model. In: Signal Processing for Image Enhancement and Multimedia Processing, pp. 189–200. Springer, Heidelberg (2008)
19. Tao, H., Huang, T.S.: Explanation-based Facial Motion Tracking Using a Piecewise Bezier Volume Deformation Mode. In: IEEE CVPR 1999, pp. 611–617 (1999)
20. Vapnik, V.N.: Statistical Learning Theory. John Wiley, New York (1998)
21. Viola, P., Jones, M.J.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: IEEE CVPR 2001, pp. 511–518 (2001)
22. Wallhoff, F.: Facial Expressions and Emotion Database, Technische Univesität München (2006), http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html
23. Zeng, Z., Pantic, M., Roisman, G.I., Wen, Z., Hu, Y., Huang, T.S.: Spontaneous Emotional Facial Expression Detection. Journal of Multimedia 1(5), 1–8 (2006)
24. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(1), 39–58 (2009)
25. Zhao, W., Challappa, R., Phillips, P., Rosenfeld, A.: Face Recognition: a Literature Survey. ACM Computing Surveys 35(4), 399–458 (2003)

A Memetic Approach to Project Scheduling that Maximizes the Effectiveness of the Human Resources Assigned to Project Activities

Virginia Yannibelli and Analía Amandi

ISISTAN Research Institute, UNCPBA, Argentina and also CONICET, Argentina
Campus Universitario, Tandil (7000), Argentina
{vyannibe,amandi}@exa.unicen.edu.ar

Abstract. In this paper, we address a project scheduling problem that considers a priority optimization objective for project managers. This objective involves assigning the most effective set of human resources to each project activity. To solve the problem, we propose a memetic algorithm. This is a hybrid algorithm that combines an evolutionary algorithm and a local search algorithm. To evaluate the performance of the memetic algorithm, we report the computational experiments carried out on six different instance sets. Then, we compare the performance of the memetic algorithm with that of the evolutionary algorithm previously proposed in literature for solving the addressed problem. The obtained results show that the memetic algorithm outperforms the previous evolutionary algorithm.

Keywords: project scheduling, human resource assignment, multi-skilled resources, effectiveness of human resources, memetic algorithms.

1 Introduction

The scheduling of a project involves defining feasible start times and feasible human resource assignments for project activities. Moreover, to define human resource assignments, it is necessary to have knowledge about the effectiveness of the available resources in relation to different project activities. This is because the development and the results of an activity depend on the effectiveness of the resources assigned to it [17, 26].

In the literature, many different kinds of project scheduling problems have been described and addressed until now. However, to the best of our knowledge, only few works have considered human resources with different levels of effectiveness [5, 15, 14, 27], a central aspect in real project scheduling problems. These works state different assumptions about the effectiveness of the human resources.

In [5, 15, 14], the authors assume that each human resource only has one or several skills, and an effectiveness level in relation to each skill. Then, the effectiveness of a human resource in a given activity is determined only on the basis of the effectiveness level of the resource in relation to one of the skills required for that activity.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 159–173, 2012.

Thus, only the skills of a human resource are considered as determining factors of their effectiveness. However, other contextual factors that also determine the effectiveness of a human resource in a given activity are not considered in the mentioned works. Such factors involve the attributes of the activity to which the resource is assigned, the other resources with whom the resource in question must work, as well as the experiences and attributes of the resource [2, 17, 26].

In contrast with the above-mentioned works, in [27] the authors consider that the effectiveness of a human resource depends on various factors inherent to its work context (i.e., the activity to which the resource is assigned, the skill to which the resource is assigned within the activity, the set of human resources that has been assigned to the activity, and the attributes of the resource). This is a really significant aspect of the work [27] above mentioned. This is because, in real projects, the human resources usually have different effectiveness levels in relation to different work contexts [2, 17, 26] and, therefore, the effectiveness of a human resource needs to be considered in relation to its work context. To the best of our knowledge, the influence of the work context on the effectiveness of the human resources has not been considered in other works that address project scheduling problems. For this reason, we consider that the above-mentioned work [27] states novel and valuable assumptions about the effectiveness of the human resources in the context of project scheduling problems.

In this paper, we address the project scheduling problem introduced in [27]. This problem considers a priority optimization objective for managers at the early stage of scheduling. This objective involves assigning the most effective set of human resources to each project activity. In this respect, as was previously mentioned, the addressed problem considers that the effectiveness of a human resource depends on various factors inherent to its work context.

To solve the problem, we propose a memetic algorithm. This is a hybrid algorithm that combines an evolutionary algorithm and a local search algorithm [11, 28, 29]. Specifically, the memetic algorithm integrates a local search algorithm within the framework of an evolutionary algorithm. The hybridization or fusion of these algorithms is meant to improve the performance of the traditional evolutionary algorithms [11, 28, 29, 30]. In particular, the incorporation of a local search stage into the evolutionary framework has the aim of fine-tuning the solutions obtained in each cycle of the evolutionary algorithm. Thus, the evolutionary-based search is augmented by the addition of one stage of local search [11].

We propose a memetic algorithm because of the following reasons. The problem addressed here can be seen as a special case of the RCPSP (Resource Constrained Project Scheduling Problem) [7] and, therefore, the problem is a NP-Hard problem. In this sense, the hybridization of evolutionary algorithms with other search and optimization techniques (e.g., local search) has been proven to be more effective than the classical evolutionary algorithms in the resolution of a wide variety of NP-Hard problems [11, 28, 29, 30] and, in particular, in the resolution of project scheduling problems [20]. Thus, we consider that a memetic algorithm could outperform the evolutionary algorithm previously proposed in [27] for solving the addressed problem.

The remainder of the paper is organized as follows. In Section 2, we give a brief review of published works that address project scheduling problems in which the effectiveness of human resources is considered. In Section 3, we describe the problem addressed in this paper. In Section 4, we present the proposed memetic algorithm. In Section 5, we present the computational experiments carried out to evaluate the performance of the memetic algorithm and an analysis of the results obtained. Finally, in Section 6 we present the conclusions of the present work.

2 Related Works

Different works in the literature have considered the effectiveness of human resources in the context of project scheduling problems. These works state different assumptions about the effectiveness of the human resources. In this respect, few works have considered human resources with different levels of effectiveness [5, 15, 14, 27], a central aspect in real project scheduling problems. In this section, we focus the attention on analyzing the way in which the effectiveness of human resources is considered in related works.

In [3, 4, 6, 23, 9], the authors address the multi-skill project scheduling problem. In this problem, each project activity requires specific skills and a given number of human resources (employees) for each required skill. Each available employee masters one or several skills, and all the employees that master a given skill have the same effectiveness level in relation to the skill (homogeneous levels of effectiveness in relation to each skill).

In [5], the authors consider the multi-skill project scheduling problem with hierarchical levels of skills. In this problem, given a skill, for each employee that masters the skill, an effectiveness level is defined in relation to the skill. Thus, the employees that master a given skill have different levels of effectiveness in relation to the skill. Then, each project activity requires one or several skills, a minimum effectiveness level for each skill, and a number of resources for each pair skill-level.

In [22], the authors address the multi-skill project scheduling problem with the aim of minimizing the total staffing cost, and consider a specific workload capacity (i.e., weeks in a project) and a specific salary for each employee. In [10], the objective of the multi-skill project scheduling problem is to minimize the maximal lateness of the project, and consider a specific workload capacity (i.e., duration of work per day, etc.) for each employee. Both works consider homogeneous levels of effectiveness in relation to each skill.

The works mentioned in the preceding three paragraphs consider that all sets of employees that can be assigned to a given activity have the same effectiveness on the development of the activity. Specifically, with respect to effectiveness, such sets are merely treated as unary resources with homogeneous levels of effectiveness.

In [15], the authors address the multi-skill project scheduling problem with three different optimization objectives (i.e., time, quality and cost). In this work, most activities require only one employee with a particular skill, and each available employee masters different skills. In addition, the employees that master a given skill

have different levels of effectiveness in relation to the skill. Then, the effectiveness of an employee in a given activity is defined by considering only the effectiveness level of the employee in relation to the skill required for the activity.

In [24, 25, 1, 12], the authors address the skilled workforce project scheduling problem considering different optimization objectives. In this problem, each project activity requires only one worker with a particular skill, and each available worker has different skills. In [24, 25], given a skill, for each worker that masters the skill, an efficiency level is defined (heterogeneous efficiencies in relation to each skill). In [1, 12], workers with homogeneous efficiencies in relation to each skill are considered. The four works consider workers with homogeneous levels of effectiveness in relation to each skill.

In [18, 14], the authors address the problem of scheduling multiple projects taking into account different optimization objectives. In [18], the authors consider human resources with homogeneous levels of effectiveness and heterogeneous efficiencies in relation to each skill. They also consider a specific workload capacity and cost per time unit for each resource. In [14], the authors consider human resources with different levels of effectiveness and heterogeneous efficiencies in relation to each skill.

Unlike the above-mentioned works, in [27] the authors consider that the effectiveness of a human resource depends on various factors inherent to its work context. Then, it is possible to define different effectiveness levels in relation to different work contexts for each human resource. This is a very relevant aspect of [27] because, in real project scheduling problems, the human resources have different effectiveness levels in relation to different work contexts [2, 17, 26] and, thus, the effectiveness of a human resource needs to be considered in relation to its work context. To the best of our knowledge, the influence of the work context on the effectiveness of the human resources has not been considered in other works that address project scheduling problems. Because of this, we consider that the above-mentioned work [27] states novel and valuable assumptions about the effectiveness of the human resources in the context of project scheduling problems.

3 Problem Description

In this paper, we address the project scheduling problem introduced in [27]. This problem is described below.

A project contains a set A of N activities, $A = \{1, \ldots, N\}$. The duration, precedence relations and resource requirements of each activity are known.

The duration of each activity j is notated as d_j. Moreover, it is considered that pre-emption of activities is not allowed (i.e., the d_j periods of time must be consecutive).

Among some project activities, there are precedence relations due to technological requirements (i.e., each of the activities consumes products generated by other activities). Thus, the precedence relations establish that each activity j cannot start until all its immediate predecessors, given by the set P_j, have completely finished.

Project activities require human resources – employees – skilled in different knowledge areas. Specifically, each activity requires one or several skills as well as a given number of employees for each skill.

Companies and organizations have a qualified workforce to develop their projects. This workforce is made up of a number of employees, and each employee masters one or several skills.

Set SK represents the K skills required to develop the project, $SK = \{1,\ldots, K\}$, and set AR_k represents the available employees with skill k. Then, the term $r_{j,k}$ represents the number of employees with skill k required for activity j of the project. The values of the terms $r_{j,k}$ are known for each project activity.

It is considered that an employee cannot take over more than one skill within a given activity. In addition, an employee cannot be assigned more than one activity at the same time.

Based on the previous assumptions, an employee can be assigned different activities but not at the same time, can take over different skills required for an activity but not simultaneously, and can belong to different possible sets of employees for each activity.

As a result, it is possible to define different work contexts for each available employee. It is considered that the work context of an employee r, denoted as $C_{r,j,k,g}$, is made up of four main components. The first component refers to the activity j which r is assigned (i.e., the complexity of j, its domain, etc.). The second component refers to the skill k which r is assigned within activity j (i.e., the tasks associated to k within j). The third component is the set of employees g that has been assigned j and that includes r (i.e., r must work in collaboration with the other employees assigned to j). The fourth component refers to the attributes of r (i.e., his or her experience level in relation to different tasks and domains, the kind of labor relation between r and the other employees of g, his or her educational level in relation to different knowledge areas, his or her level with respect to different skills, etc.). It is considered that the attributes of r could be quantified from available information about r (e.g., curriculum vitae of r, results of evaluations made to r, information about the participation of r in already executed projects, etc.).

The four components described above are considered the main factors that determine the effectiveness level of an employee. For this reason, it is assumed that the effectiveness of an employee depends on all the components of his or her work context. Then, for each employee, it is possible to consider different effectiveness levels in relation to different work contexts.

The effectiveness level of an employee r, in relation to a possible context $C_{r,j,k,g}$ for r, is notated as $e_{rCr,j,k,g}$. The term $e_{rCr,j,k,g}$ represents how well r can handle, within activity j, the tasks associated to skill k, considering that r must work in collaboration with the other employees of set g. The mentioned term $e_{rCr,j,k,g}$ takes a real value over the range [0, 1]. The values of the terms $e_{rCr,j,k,g}$ inherent to each employee available for the project are known. It is considered that these values could be obtained from available information about the participation of the employees in already executed projects.

The problem of scheduling a project entails defining feasible start times (i.e., the precedence relations between the activities must not be violated) and feasible human resource assignments (i.e., the human resource requirements must be met) for project activities in such a way that the optimization objective is reached. In this sense, a priority objective is considered for project managers at the early stage of the project schedule design. The objective is that the most effective set of employees be assigned each project activity. This objective is modeled by Formulas (1) and (2).

Formula (1) maximizes the effectiveness of the sets of employees assigned to the N activities of a given project. In Formula (1), set S contains all the feasible schedules for the project in question. The term $e(s)$ represents the effectiveness level of the sets of employees assigned to project activities by schedule s. Then, $R(j,s)$ is the set of employees assigned to activity j by schedule s, and the term $e_{R(j,s)}$ represents the effectiveness level corresponding to $R(j,s)$.

Formula (2) estimates the effectiveness level of the set of employees $R(j,s)$. This effectiveness level is estimated calculating the mean effectiveness level of the employees belonging to $R(j,s)$. The mean effectiveness level is used because of the reasons presented below. It is considered that the sets of employees are assigned to project activities with the following properties. First, in the activities considered here, the effectiveness level of a set of employees depends on the effectiveness level of each employee belonging to the set. Second, the higher the sum of the effectiveness levels of those employees, the higher the effectiveness of the set. In the case of project activities with the properties mentioned, it is considered that the mean effectiveness level of the employees of a set is a good predictor of the effectiveness of the set. This is because the mean effectiveness level of the employees of a set is directly proportional to the sum of the effectiveness levels of those employees. Specifically, the higher the sum of the effectiveness levels of the employees, the higher the mean effectiveness level. Thus, if the mean effectiveness level is used as a predictor, the higher the sum of the effectiveness levels of the employees, the higher the effectiveness of the set.

For a more detailed discussion of Formula (1) and Formula (2), we refer to [27].

$$\max_{\forall s \in S} \left(e(s) = \sum_{j=1}^{N} e_{R(j,s)} \right) \tag{1}$$

$$e_{R(j,s)} = \frac{\sum_{r=1}^{|R(j,s)|} eC_{r,j,k(r,j,s),R(j,s)}}{|R(j,s)|} \tag{2}$$

4 Memetic Algorithm

To solve the problem, we propose a memetic algorithm. This is a hybrid algorithm that combines an evolutionary algorithm and a local search algorithm [11, 28, 29]. Specifically, the memetic algorithm incorporates a local search stage into the

framework of an evolutionary algorithm. The incorporation of a local search stage into the evolutionary framework has the aim of fine-tuning the solutions obtained in each cycle of the evolutionary algorithm. Thus, the evolutionary-based search is augmented by the addition of one stage of local search [11].

The general behavior of the memetic algorithm is shown in Fig. 1 and is described as follows. Considering a given project, the algorithm starts the evolution from an initial population of solutions in which each solution codifies a feasible project schedule. Then, each solution of the population is decoded (i.e., the related schedule is built), and evaluated according to the optimization objective of the problem by a fitness function. As explained earlier, the objective is to maximize the effectiveness of the sets of employees assigned to project activities. In relation to this objective, the fitness function evaluates the assignments of each solution based on knowledge about the effectiveness of the employees involved in the solution. Then, a selection process is used to select a number of solutions from the current population according to a selection strategy. The selected solutions are paired, and a crossover process is applied to each pair of solutions to generate new feasible ones. Then, a mutation process is applied to the generated solutions by the crossover. Then, a standard local search improvement algorithm [11] is applied to the generated solutions by the mutation. This local search algorithm is aimed at fine-tuning solutions. Finally, a replacement strategy known as deterministic crowding [11] is used to create a new population from the solutions in the current population and the new generated solutions.

This process is repeated until a predetermined number of repetitions or iterations is reached.

```
BEGIN
    CREATE initial population;
    EVALUATE each candidate solution;
    REPEAT UNTIL ( number of iterations is reached ) DO
        SELECT parents;
        RECOMBINE pairs of parents to produce offspring;
        MUTATE offspring;
        EVALUATE offspring;
        IMPROVE offspring via Local Search;
        CREATE new population;
    OD
END
```

Fig. 1. General behavior of the memetic algorithm

4.1 Representation of Solutions

Each solution in the memetic algorithm population represents or encodes a feasible project schedule. In this respect, we used an appropriate representation to project schedules. This representation was introduced in [27].

Each solution is represented by two lists having as many positions as activities in the project. The first list is a standard activity list [19]. This list is a feasible precedence list of the activities involved in the project (i.e., each activity j can appear on the list in any position higher than the positions of all its predecessors). The activity list describes the order in which activities shall be added to the schedule.

The second list is an assigned resources list [27]. This list contains information about the employees assigned to each activity of the project. Specifically, position j on this list details the employees of every skill k assigned to activity j.

In order to build the schedule related to the representation, we considered the serial schedule generation method [19]. In this method, each activity j is scheduled at the earliest possible time.

4.2 Initial Population

In order to generate each solution of the initial population, we used a two-stage process introduced in [27]. The first stage defines a feasible activity list. This stage begins with an empty activity list. The next activity for the list is randomly taken from the activities not yet inserted on the list while all its predecessors have already been inserted in it.

The second stage defines a feasible assigned resources list. This stage assigns employees to each project activity. Each activity j requires $r_{j,k}$ employees with skill k ($k = 1,\ldots, K$). For each activity j, $r_{j,k}$ employees with skill k ($k = 1,\ldots, K$) are randomly selected from the group of available employees with skill k, AR_k, and the selected employees are assigned to activity j (i.e., are assigned to position j of the assigned resources list).

This process guarantees that the precedence relationships between the activities are not violated and that the human resource requirements for each activity are met during the construction of each solution.

4.3 Fitness Function

The fitness function evaluates a given solution in relation to the predefined optimization objective. In this case, the objective is to maximize the effectiveness level of the sets of employees assigned to the project activities.

Given a solution to a project p, the fitness function decodes the schedule s related to the solution by using the serial method mentioned in Section 4.1. Then, the function calculates the value of the term $e(s)$ corresponding to s (Formulas (1) and (2)). This value determines the fitness level of the solution. The term $e(s)$ takes a real value over $[0, \ldots, N]$.

To calculate the term *e(s)*, the function utilizes the values of the terms $e_{rCr,j,k,g}$ inherent to *s* (Formula 2). As was mentioned in Section 3, the values of the terms $e_{rCr,j,k,g}$ inherent to each available employee *r* are known.

4.4 Selection, Crossover and Mutation

In relation to the selection process, we applied the 2-tournament selection scheme, one of the most applied in the literature [11, 13].

The crossover and mutation operators were defined on the basis of the representation used for the solutions. Thus, the crossover operator contains a feasible crossover operation for activity lists and a feasible crossover operation for assigned resources lists. For activity lists, we considered the two-point crossover developed by Hartmann [16]. For assigned resources lists, we considered the standard uniform crossover [27]. The crossover operator is applied with a probability of P_c.

The mutation operator contains a feasible mutation operation for activity lists and a feasible mutation operation for assigned resources lists. For activity lists, we consider the mutation operation introduced in [27]. This operation is an adaptation of the procedure proposed by Boctor [8] in his simulated annealing algorithm to generate a neighbour. The behaviour of the operation is as follows. For each activity on a given activity list, a new position is randomly chosen. In particular, the new position must be higher than the position corresponding to any of the activity's predecessors, and lower than the position corresponding to any of the activity's successors. The activity is inserted in the new position with a probability of P_m. For assigned resources lists, we consider the mutation operation introduced in [27]. The behaviour of this operation is as follows. For each activity on a given assigned resources list, the operation defines a new resource assignment with a probability of P_m. In case a new resource assignment needs to be defined for a given activity, the operation considers the second stage of the mechanism used to create the random solutions of the initial population (Section 4.2).

4.5 Local Search Algorithm

In order to fine-tune the solutions obtained after the mutation process, a standard local search algorithm [11] is applied to each of these solutions.

This local search algorithm is an iterative algorithm that starts from a given solution *x* and then searches the solutions in the neighbourhood of *x* to find a solution *x'* that performs better than *x*. If in the neighbourhood of the current solution *x* a better solution *x'* is found, it replaces the current solution and the local search is continued from *x'*; if no better solution is found, the current solution *x* is considered a local optimal solution and the algorithm terminates in *x*.

The described local search algorithm has two main parameters: the pivot rule and the move operator. The pivot rule determines which neighbouring solution replaces the current one (e.g., the first found highest fitness solution, the best solution of the neighbourhood, etc.). The move operator generates neighbouring solutions of the

current solution. In this respect, different move operators could generate different neighbouring solutions (i.e., different neighbourhoods) [11].

The two above-mentioned parameters affect the performance of the local search algorithm, both in terms of time taken, and in the quality of solution found [11].

5 Computational Experiments

In this section, we describe the computational experiments developed to evaluate the performance of the memetic algorithm and we analyze the obtained results. Then, we compare the performance of the memetic algorithm with that of the evolutionary algorithm previously proposed in [27] for solving the addressed problem.

In order to develop the experiments, we selected test instances of the standard sets j30, j60, and j120 from PSPLIB [21], and then we extended the content of the selected instances with the aim of adapting these instances to the characteristics of the proposed algorithm. Table 1 shows the characteristics of the instance sets defined to develop the experiments.

Table 1. Characteristics of instance sets

Instance set	Activities per instance	Possible sets of employees per activity	Instances
j30_5	30	1 to 5	40
j30_10	30	1 to 10	40
j60_5	60	1 to 5	40
j60_10	60	1 to 10	40
j120_5	120	1 to 5	40
j120_10	120	1 to 10	40

For each instance selected from PSPLIB, we defined all the terms $e_{rCr,j,k,g}$ inherent to each employee r of the instance - these terms are defined considering each of the possible work contexts for r in the instance - and a random value over [0, 1] for each of the above-mentioned terms.

Then, for each extended instance, we designed an optimal solution with the aim of using it as a reference. Specifically, for each extended instance, we designed a feasible solution s. Solution s was designed by the mechanism described in Section 4.2. Then, we defined all existing terms $e_{rCr,j,k,g}$ in the solution s. We assumed a value equal to 1 (maximal value) for each of the terms, and we added them to the content of the instance, together with the assumed values. When adding the assumed values to the content of the instance, the level $e(s)$ of the solution s is equal to N (N is the number of activities of the instance). Thus, we defined a feasible solution s with a level $e(s)$ equal to N for the instance. Considering that N is the maximal level for the solutions of an instance with N activities, the defined solution s can be considered as an optimal solution.

The memetic algorithm has been tested 20 times on each of the extended instances. Table 2 gives the parameter values used for these experiments. The parameters were fixed thanks to preliminary experiments that showed that these values led to the best and most stable results.

Table 2. Parameter values of the memetic algorithm

Parameter	Value
Crossover probability P_c	0.8
Mutation Probability P_m	0.05
Population size	90
Number of generations	300
Parameters of the local search algorithm	
Pivot rule	The current solution must be replaced by the first found highest fitness neighbor
Move operator	Insert mutation [11]

Table 3 reports the results obtained by the experiments. Column 2 reports the average percentage deviation from the optimal solution (Av. Dev. (%)) for each instance set. Column 3 reports the percentage of instances for which the value of the optimal solution is achieved at least once among the 20 generated solutions (Optimal (%)).

Table 3. Results obtained by the computational experiments

Instance set	Av. Dev. (%)	Optimal (%)
j30_5	0	100
j30_10	0	100
j60_5	0	100
j60_10	0.1	100
j120_5	0.75	100
j120_10	0.91	100

The results obtained by the algorithm for j30_5, j30_10 and j60_5 indicate that the algorithm has found an optimal solution in each of the 20 runs carried out on each instance of the sets.

The Av. Dev. (%) obtained by the memetic algorithm for j60_10, j120_5 and j120_10 is greater than 0%. Considering that the instances of j60_10 and the instances of j120_5 and j120_10 have known optimal solutions with a level *e(s)* equal to 60 and 120 respectively, we analyzed the meaning of the average deviation obtained for each one of these sets. In the case of j60_10, an average deviation equal to 0.1% indicates that the average value of the solutions obtained by the algorithm is 59.94. In the case of j120_5 and j120_10, average deviations equal to 0.75% and 0.91% indicate that the average value of the solutions obtained by the algorithm is 119.1 and 118.908 respectively. Therefore, we may state that the algorithm has obtained very high quality solutions for the instances of j60_10, j120_5 and j120_10.

Moreover, the Optimal (%) obtained by the algorithm for j60_10, j120_5 and j120_10 is 100%. These results indicate that the algorithm has found an optimal solution in at least one of the 20 runs carried out on each instance of the sets.

5.1 Comparison with a Competing Algorithm

In this section, we compare the performance of the memetic algorithm with that of the evolutionary algorithm previously proposed in [27] for solving the addressed problem. For simplicity, we will refer to the evolutionary algorithm proposed in [27] as algorithm E. Unlike the memetic algorithm, the algorithm E is not a hybrid algorithm. The framework of the algorithm E is a standard evolutionary framework and, therefore, only includes the standard evolutionary stages (i.e, selection of parents, crossover, mutation and selection of individuals for the next population).

To develop the above-mentioned comparison, the algorithm E was run 20 times on each of the instances of the 6 sets. To perform these runs, the algorithm parameters were set with the values recommended in [27]. Table 4 shows the results obtained by the algorithm E on the 6 instance sets.

Table 4. Results obtained by the algorithm E

Instance set	Av. Dev. (%)	Optimal (%)
j30_5	0	100
j30_10	0	100
j60_5	0.42	100
j60_10	0.59	100
j120_5	1.1	100
j120_10	1.29	100

The results reported in Table 4 and Table 3 indicate that the algorithm E and the memetic algorithm have obtained the same effectiveness level (i.e., an optimal effectiveness level) on the first two instance sets (i.e., the less complex sets). However, the effectiveness level obtained by the memetic algorithm on the last four instance sets (i.e., the more complex sets) is higher than the effectiveness level obtained by the algorithm E on these sets. The memetic algorithm has outperformed the algorithm E on the four more complex instance sets. The main reason for this is that, in contrast with the algorithm E, the memetic algorithm incorporates a local search algorithm into the evolutionary framework and this local search algorithm fine-tunes the solutions obtained in each evolutionary cycle. Thus, the memetic algorithm can reach more effective solutions than the algorithm E on the more complex instance sets. Based on the above-mentioned, the memetic algorithm may be considered to obtain higher-quality solutions than the algorithm E.

6 Conclusions

We have addressed the project scheduling problem introduced in [27]. In this problem, it is considered a priority optimization objective for managers at the early stage of scheduling. This objective is to maximize the effectiveness level of the sets of employees assigned to the project activities. In the problem, it is considered that the effectiveness level of an employee depends on different factors inherent to his or her work context. Therefore, it is possible to define different effectiveness levels in relation to different work contexts for each employee. To the best of our knowledge, the influence of the work context on effectiveness of the employees has not been considered in works previous to [27].

To solve the problem, we have proposed a memetic algorithm. This is a hybrid algorithm that combines an evolutionary algorithm and a local search algorithm. Specifically, the memetic algorithm incorporates a local search stage into the framework of an evolutionary algorithm with the aim of fine-tuning the solutions obtained in each cycle of the evolutionary algorithm. Thus, the evolutionary-based search is augmented by the addition of one stage of local search.

To evaluate the performance of the proposed memetic algorithm, we have presented the computational experiments carried out on six different instance sets. Besides, we have compared the performance of the memetic algorithm with that of the evolutionary algorithm previously proposed in [27] for solving the addressed problem.

Based on the results obtained by the memetic algorithm for each one of the six sets, we may state that this algorithm has reached an optimal level of effectiveness on the first three sets, and that the algorithm has reached a high level of effectiveness on the remaining three sets. Furthermore, as a result of the comparative analysis conducted, we may state that the memetic algorithm has been shown to be more effective than the evolutionary algorithm proposed in [27] on the used instance sets. Thus, we consider that the memetic algorithm may be used to obtain higher-quality solutions than the evolutionary algorithm proposed in [27].

In future works, we will include other relevant optimization objectives into the addressed problem (e.g., the minimization of the project makespan and the minimization of the project cost) and we will adapt the fitness function of the proposed memetic algorithm. On the other hand, in future works, we will propose new feasible crossover and mutation processes for the used representation of solutions.

Considering that the proposed algorithm uses knowledge about the effectiveness of the employees in relation to different work contexts, future research should be conducted in order to develop approaches that automatically estimate the effectiveness levels of the employees from available information about the participation of the employees in already executed projects. These approaches will complement our algorithm.

References

1. Aickelin, U., Burke, E., Li, J.: An Evolutionary Squeaky Wheel Optimization Approach to Personnel Scheduling. IEEE Transactions on Evolutionary Computation 13(2), 433–443 (2009)
2. Barrick, M.R., Stewart, G.L., Neubert, M.J., Mount, M.K.: Relating member ability and personality to work-team processes and team effectiveness. Journal of Applied Psychology 83, 377–391 (1998)
3. Bellenguez, O.: A reactive approach for the multi-skill Project Scheduling Problem. In: 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008), pp. 1–4. Université de Montréal, Montréal (2008)
4. Bellenguez, O., Néron, E.: Methods for the multi-skill project scheduling problem. In: 9th International Workshop on Project Management and Scheduling (PMS 2004), pp. 66–69. Université Nancy, Nancy (2004)
5. Bellenguez, O., Néron, E.: Lower Bounds for the Multi-skill Project Scheduling Problem with Hierarchical Levels of Skills. In: Burke, E., Trick, M. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 229–243. Springer, Heidelberg (2005)
6. Bellenguez, O., Néron, E.: A branch-and-bound method for solving multi-skill project scheduling problem. RAIRO - Operations Research 41(2), 155–170 (2007)
7. Blazewicz, J., Lenstra, J., Rinnooy Kan, A.: Scheduling Subject to Resource Constraints: Classification and Complexity. Discrete Applied Mathematics 5, 11–24 (1983)
8. Boctor, F.F.: An adaptation of the simulated annealing algorithm for solving resource constrained project scheduling problems. International Journal of Production Research 34, 2335–2351 (1996)
9. Néron, E.: Lower Bounds for the Multi-Skill Project Scheduling Problem. In: Eighth International Workshop on Project Management and Scheduling, pp. 274–277. University of Valencia, Valencia (2002)
10. Drezet, L.E., Billaut, J.C.: A project scheduling problem with labour constraints and time-dependent activities requirements. International Journal of Production Economics 112, 217–225 (2008)
11. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing, 2nd edn. Springer, Heidelberg (2007)
12. Focacci, F., Laborie, P., Nuijten, W.: Solving scheduling problems with setup times and alternative resources. In: Fifth International Conference on Artificial Intelligence Planning and Scheduling, Breckenbridge, CO, USA, pp. 92–101 (2000)
13. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Inc. (2007)
14. Gutjahr, W.J., Katzensteiner, S., Reiter, P., Stummer, C., Denk, M.: Competence-driven project portfolio selection, scheduling and staff assignment. Central European Journal of Operations Research 16(3), 281–306 (2008)
15. Hanne, T., Nickel, S.: A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects. European Journal of Operational Research 167, 663–678 (2005)
16. Hartmann, S.A.: Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. Naval Research Logistics 45, 733–750 (1998)
17. Heerkens, G.R.: Project Management. McGraw-Hill (2002)
18. Heimerl, C., Kolisch, R.: Scheduling and staffing multiple projects with a multi-skilled workforce. OR Spectrum 32(4), 343–368 (2010)

19. Kolisch, R., Hartmann, S.: Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. In: Weglarz, J. (ed.) Project Scheduling: Recent Models, Algorithms and Applications, pp. 147–178. Kluwer Academic (1999)
20. Kolisch, R., Hartmann, S.: Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. European Journal of Operational Research 174, 23–37 (2006)
21. Kolisch, R., Sprecher, A.: PSPLIB - A project scheduling library. European Journal of Operational Research 96, 205–216 (1997)
22. Li, H., Womer, K.: Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. Journal of Scheduling 12, 281–298 (2009)
23. Néron, E., Bellenguez, O., Heurtebise, M.: Decomposition method for solving multi-skill project scheduling problem. In: Tenth International Workshop on Project Management and Scheduling, pp. 265–269. Wydawnictwo Nakom, Poznan (2006)
24. Valls, V., Gomez-Cabrero, D., Pérez, M.A., Quintanilla, S.: Project Scheduling Optimization in Service Centre Management. Tijdschrift voor Economie en Management 52(3), 341–366 (2007)
25. Valls, V., Pérez, A., Quintanilla, S.: Skilled workforce scheduling in service centers. European Journal of Operational Research 193(3), 791–804 (2009)
26. Wysocki, R.K.: Effective Project Management, 3rd edn. Wiley Publishing (2003)
27. Yannibelli, V., Amandi, A.: A knowledge-based evolutionary assistant to software development project scheduling. Expert Systems with Applications 38(7), 8403–8413 (2011)
28. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
29. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
30. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)

Hunting for Fraudsters in Random Forests

R.M. Konijn and W. Kowalczyk

Department of Computer Science, Faculty of Sciences, VU University Amsterdam
{r.m.konijn,w.j.kowalczyk}@vu.nl

Abstract. In this paper we present a hybrid method for identifying suspicious behavior in transactional data by combining techniques from outlier detection and subgroup discovery. Most existing outlier detection approaches focus on the identification of single outliers without providing a description of these outliers. Moreover, these methods find single outliers instead of groups of outlying records. However, when searching for fraud, it is important to analyze data not on the level of single records but on a higher, group level, such as sets of records of customers, shops, etc. Our method is able to analyze data on such a higher level and additionally it provides descriptions of groups of found outliers.

The method involves three steps: scoring of individual records with help of a newly proposed outlier measure which is calculated with help of random forests, identification of unusual groups of records with help of subgroup discovery techniques, and finally, identify the most deviating entities such as shops, hospitals.

Keywords: subgroup discovery, outlier detection, fraud detection, random forests.

1 Introduction

The main motivation for this paper comes from a real life problem where we are interested in detecting fraud in health insurance data. After trying out several standard outlier detection approaches, we found that from the business point of view single outliers are not that interesting, and it is much more valuable to know if groups of records of for example a pharmacy or dentist as a whole can be labeled as anomalous. Furthermore, especially distance-based outlier detection methods lack any form of interpretability. When working with a domain expert, who is for example an expert in medicine or dental treatments, it is much more efficient to provide easy to understand descriptions of outliers rather than to present single cases iteratively. Our goal is thus to construct an outlier detection system that provides easy to understand descriptions of groups of outlying records, and can label care givers like pharmacies and dentists as being fraudulent or not. Ideally the output of our approach should be like: "Dentist number 142 is suspicious, because the group of patients of this dentist with characteristics A and B significantly differs from others".

The paper is organized as follows: We start with related work on outlier detection, random forests, and methods that provide feedback on outliers. Next we describe background knowledge of subgroup discovery, which is needed to understand our approach. In the third section we describe our approach in three steps. Section four contains experimental results on a real life health insurance dataset and on the publicly available abalone dataset.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 174–185, 2012.

2 Related Work

Our approach combines several techniques from the field of machine learning: distance-based methods for detecting outliers, a powerful regression method, called Random Forests, and subgroup discovery. It can be considered an unsupervised learning problem, so outliers are not labeled beforehand. We will discuss related approaches in this section first, and then provide some background information about subgroup discovery.

Approaches to outlier detection that involve measuring the distance between single records can be split into the following three categories: depth-based methods, distance-based methods, and density-based methods. *Depth-based* methods measure the distance from a point to the center of the data – therefore the term *depth*. Points that have the biggest distance to the center are considered outliers [13]. The main disadvantage of depth-based methods is their inability of handling clusters in data – they assume that data form a single cluster. *Distance-based* methods require a distance measure to determine the distance between two instances. The main idea is that the distance between outlying instances and their neighbors is bigger than the distance between normal instances and their neighbors, [7]. *Density-based* methods measure outlierness of a point by analyzing the distances between the point and points in its local neighborhood. Examples of density-based measures are the Local Outlier Factor (LOF), [4], the Connectivity-based Outlier Factor (COF), [14], or the Multi-granularity Deviation Factor (MDEF), [11].

The concept of *random forests* was introduced by Leo Breiman in 2001 [3], and since then it became one of the most prominent technique for solving classification and regression problems [15]. The key idea behind this technique is a construction of many (hundreds) of *de-correlated* classification or regression trees and then aggregating their predictions. This leads to models with very good accuracy. Moreover, forests determine a proximity measure on data points: the more frequently two points fall into the same leaf the higher their proximity. Random forests were successfully used for intrusion detection, [18]. The authors first label the data by a number of classes, including a class that correspond to intrusion. Next, they build a random forest for the corresponding classification problem and use the induced proximity matrix to find outliers. Outliers are defined as points having a relatively low proximity to the points of the same class. Another approach to outlier detection, [9], uses randomized trees called *I-trees*. Similarly to random forests, a large collection of trees is generated automatically from data; trees are grown until each leaf shrinks to a single point. Outlierness of a point is measured by the depth (the distance to the root) of a leaf that contains this point; the smaller the depth the higher the outlierness of the point.

There has been research in the area of finding the so-called *intentional knowledge* of outliers, [7]. The intentional knowledge is defined as a set of dimensions that is small enough for an instance to be considered an outlier. Removal of any of these dimensions, and recalculating the score afterwards, would result in the instance not being considered an outlier anymore. A method described in [17] first labels records as outliers using their smartsifter algorithm, and then creates the so-called stochastic decision list: a set of rules in the form $constraints \rightarrow outlier$ with a probability higher than a predefined threshold. The focus in this article is on the on-line part of the outlier detection process, and the main purpose of the decision lists is classification rather than description.

Yet another approach to finding outlying groups of records is presented in [20] and [19], where knowledge of already found outliers is used for searching for new outliers. The method can only be used if some cases are labeled as outliers. First, with the use of an unsupervised learning method, instances are ranked by a scoring function that is based on the Multi-granularity Deviation Factor (MDEF), [10], where a high score indicates a high probability of being an outlier. From this set, the records with the lowest score are put in a set together with the known outliers. A Support Vector Machine model is then trained on this set in an iterative way, after which the fraction of the data classified as positive by the SVM is selected as the outlying set.

Subgroups can also be thought of as multi dimensional cubes containing data points. In [1] the authors propose an evolutionary search algorithm to find multi-dimensional cubes with a very low density, where large cubes with few points have a low density. These cubes are the subgroups that describe the outliers.

2.1 Background: Subgroup Discovery

In its general form, *subgroup discovery* [16], is defined as follows: Given a population of individuals and a property of those individuals we are interested in: find population subgroups that are statistically 'most interesting'. For example, subgroups that are as large as possible and have the most unusual statistical characteristics with respect to the property of interest. A special type of subgroup discovery is the problem of mining *contrast sets* [2], where one searches for subgroups of records that can be defined by a conjunction of several attribute-value combinations. Yet another approach to subgroup discovery, called *exceptional model mining*, is presented in [8]. This time one measures the interestingness of a group of records by the *exceptionality* of a model that is fitted to this group.

We will describe subgroup discovery here in more detail. A subgroup discovery approach consists of a description of a subgroup, which are in our case 'rectangular descriptions': conjunctions of the form <attribute, condition, value>, for example $age \leq 18$. An interestingness measure is used to quantify the interestingness of such subgroups. In order to find interesting subgroups a search strategy has to be used, because the number of all possible conjunctions of attribute value pairs is usually too big to enumerate.

We can either use an outlier score as a continuous target variable, or we can use being an outlier or not as a binary target variable. In this paper we will use being an outlier as a binary property. We will explain an interesting measure for this case. Interestingness measures for continuous variables can be found in [12].

Interestingness Measure for Binary Target Variable. Subgroup discovery interestingness measures usually operate on the 2x2 contingency table of the subgroup and the target variable. When we have two binary properties, named S of a record in the database belonging to a subgroup, and T indicating whether the target variable is zero or one, the weighted relative accuracy (WRAcc) is defined as:

$$WRAcc = P(S)(P(T|S) - P(T)) = P(TS) - P(T)P(S), \tag{1}$$

where $P(S)$ is the fraction of data belonging to subgroup S, $P(T)$ is the fraction of data for which the target variable is true, and $P(TS)$ is the fraction within the data for which the subgroup and the target variable are true. Other interestingness measures are for example accuracy, relative risk, and the odds ratio [6].

Subgroup Discovery Search Process. Subgroup discovery approaches try to find subgroups for which the interestingness measure is as high as possible. Usually a heuristic search algorithm like beam search or evolutionary search algorithm is used to find interesting subgroups.

3 Our Approach

The method we propose in this paper addresses the multi-level nature of transactional data. We will demonstrate our approach on the health care domain. In this case we have a database D with records, where each record is an aggregation of costs spent on (dental) treatments by a patient in a certain time period. In this section we will use this domain to explain our approach.

The detection process is split into three phases:

1. Finding outliers on individual (patient) level. Here we search for patients with overall spending or claim pattern deviating from others. To model spending patterns and to measure similarity between individuals we develop an ensemble of regression trees (a random forest) together with the underlying proximity measure.
2. Finding descriptions of deviating groups: minimal sets of attributes that contribute most to their interestingness. Here we deploy methods from subgroup discovery to compare each group against all remaining group and to rank all groups by their interestingness.
3. Finding outliers on a group (e.g., all patients of a dentist) level. Here we use an interestingness measure commonly used for subgroup discovery, described in more detail below.

3.1 Step 1: Finding Outliers on Individual (Patient) level

Methods for detecting outliers in data operate on vectors of fixed lengths. However, data stored by insurance companies is more complex: for a single patient there might be several records available (e.g., one record per each treatment received by the patient). Additionally, records related to patient's age, gender, marital status, etc. are also stored. Therefore, before applying any method for detecting outliers, an aggregation of available records, on patients' level, should take place. This can be done in many ways; usually several trial and error attempts will provide a workable aggregation scheme.

In our experiments we decided to represent each patient by a binary vector that represents a set of treatments received by the patient within the last 12 months. More precisely, assuming that the number of possible treatments is k (in our case $k = 141$), each patient is represented by a binary vector of length k, such that the presence of 1 at the i-th position indicates that the patient received the i-th treatment within the last

year. In addition to 'treatment vectors', we assign to each patient a 'target' variable: the total cost of related treatments. Moreover, a number of 'personal data', such as age and gender are also attached to each patient.

Most outlier detection methods require a distance measure. However, common distance measures, such as Euclidean distance, Hamming distance, Jaccard coefficient, etc. do not address aspects that are related to fraud. Usually, in the context of fraud, variables that are strongly related to costs are more important than remaining variables. Therefore, we propose a new distance measure, which takes into account both the cost-related variables and the deviations in predicted costs. We achieve this with help of a random forest - a big collection of trees - that is used to model the relation between input variables ('treatment vectors') and the 'target variable' - total costs. Once a random forest is trained, we will use it for measuring the distance between individuals. This distance measure, combined with the with prediction errors made by the forest, will lead to our outlierness measure: a local residual-based outlier score. Now we will explain our method in more detail.

Random Forests,[3,15], are big collections of classification or regression trees that are build on bootstrapped samples of all records, i.e., samples of N records drawn with replacement from the whole set of records of size N. There are good heuristics available for setting parameters like the number of trees, and generally the method is not very sensitive to the choice of parameters. In our case, we will use random forests both for measuring the distance (or proximity) between patients and for estimating their costs. The proximity of two records, with respect to a random forest, is defined as the percentage of leaves that contain both records, counted over all trees from the forest. In this way, the similarity between two records depends mainly on predictors that contribute to similar costs. Therefore, the distance measure that is induced by a random forest is sensitive only to attributes that affect the target variable - cost. This is exactly what we need when clustering data in the context of fraud detection.

This approach can only be applied to a transactional data set for which costs are aggregated. The transition of the explanatory variables from a numeric vector to a binary vector does not make sense for data sets with only explanatory variables like age or gender, and also the choice of the target variable is not clear. In that case other outlier scores can be used, which we show in our experimental results section.

Using the generated random forest we can define a new outlierness measure: the local residual outlier score (LRBOS) that can be applied to any record in the database.

The Local Residual Score is based on the proximity matrix of the random forest algorithm, and the out of bag predictions of the observations [15]. Proximity between any two observations is defined as a fraction of trees in the random forest for which these two observations belong to the same leaf. To obtain a distance measure, we define the distance between two observations x, y as $distance(x, y) = 1 - proximity(x, y)$.

Let $S_p(r)$ be the set of nearest neighbors of a point p within radius r, $r \in [0, 1]$, according to the distance measure defined by the proximity, including the point p itself. Let res_p denote the residual of the predicted value at p defined as $y - oob(p)$, where $oob(p)$ is value of the out of bag prediction of p and y is the target variable. Next, we define the average r-density of p and the standard deviation of its r-density by $\mu_p(r)$ and $\sigma_p(r)$. These are calculated as the mean of $res_x, x \in S_p(r)$ and the standard deviation

of $res_x, x \in S_p(r)$. The final Local Residual Based Outlier Score (LRBOS) is defined as the z-score of the local residual:

$$LRBOS_r(p) = \frac{res_p - \mu_p(r)}{\sigma_p(r)} \tag{2}$$

It can be noticed that a global outlier score is a special case of the Local Residual Based Outlier Score with $r = 1$. To calculate a score to identify local outliers, r should be smaller than one.

3.2 Step 2: Finding Descriptions of Deviating Groups

In the second step of our approach we try to describe regions of the data set that contain relatively many outliers. In subgroup discovery terminology: we try to find subgroups with a high interestingness measure for the rule: $subgroup \rightarrow outlier$

We will use a modified version of the Weighted Relative Accuracy to evaluate the rule interestingness for different subgroups. We use a modified version of the 'standard' Weighted Relative Accuracy because users should be able to alter the interestingness measure according to their needs. Within our approach the target group is fixed: we define being an outlier as a binary property by using a threshold on the outlier score. A cut-off point determining whether an observation is considered an outlier or not can be set by a domain expert (by observing some of the outliers), or can be set by observing the distribution of the outlier score.

We obtain descriptions of outliers by finding subgroup descriptions with a high value for the 'standard' Weighted Relative Accuracy for the rule $subgroup \rightarrow outlier$. We use beam search to find the most interesting subgroups. For the search process we only set restrictions on the beam width and the depth of the search, not on the value of the interestingness measure or the coverage of the subgroups.

After observing the descriptions of outliers, the user may conclude that the interestingness measure is not specific or not general enough. Descriptions may, for example, contain too many false positives according to the user. We describe here a modified interestingness measure for the case of a binary target variable.

Modified Score. Users can decide to put more weight on accuracy or generality, according to their needs. We explain here how scores can be altered by two parameters λ_1 and λ_2. The Weighted Relative Accuracy is defined as:

$$WRAcc = p(S)(p(T|S) - P(T)) = p(TS) - p(T)p(S), \tag{3}$$

where T stands for target, and S for subgroup, so it is an interestingness measure for the rule $subgroup \rightarrow target$. In order for the user to trade off generality against accuracy, we introduce two weights λ_1 and λ_2 into the WRAcc measure that can be adjusted by the user.

$$ModifiedWRAcc = \lambda_1 P(TS) - \lambda_2 P(T)P(S), \tag{4}$$

where T stands for the target variable, S for the subgroup, λ_1 weights the importance of the coverage, and λ_2 weights the importance of the accuracy. A very high value for λ_2/λ_1 corresponds to measuring accuracy only, a very low value corresponds to measuring coverage only. The following heuristic can be used when setting λ_2/λ_1: First start with $\lambda_2/\lambda_1 = 1$, which corresponds to the weighted relative accuracy interestingness measure. When a higher accuracy is needed, the factor λ_2/λ_1 can be doubled until the subgroups are accurate enough. A similar heuristic can be used when a higher coverage is needed, only then halving λ_2/λ_1 each time.

3.3 Step 3: Finding Outliers on a Group Level

After performing the subgroup discovery step, the last step is the identification of possible fraudsters. Remember that each record in the database belongs to a single entity, where an entity can for example be a shop, or in case of health insurance data a hospital, pharmacy, etc. We identify 'suspicious' entities by measuring, for the most interesting subgroups, the (modified) weighted relative accuracy of the rule:

$$subgroup \rightarrow entity$$

The measure is calculated for each entity for the top n most interesting subgroups discovered in the subgroup discovery process, n is a parameter set by the user. In this way we identify entities that are suspicious wrt different groups of outliers.

4 Results and Analysis

In this section we apply our method to two datasets. The first one is a real life dataset of health insurance claims that were claimed by prosthodontist: dentists that are specialized in dental prosthesis. For the sake of reproducibility we demonstrate our approach on the abalone dataset from the UCI Machine Learning repository, [5]. Here we try to identify the most suspicious entity, where the entities are defined as male, female, and infant Abalones.

4.1 Results on Prosthodontist Data

This dataset consists of health insurance claims made by about 500 prosthodontists. We apply our approach to identify suspicious claim behavior.

Step 1: Finding Outliers on Individual (Patient) Level. Our dataset consists of records from 500 prosthodontists and about 50,000 patients. Each patient belongs to a single prosthodontist. Our target variable Y is the total claim amount of a patient during a time period of one year. To predict this variable, we have 161 explanatory variables at our disposal. These are: age, gender, 14 variables describing costs spent on other dental care (other than prosthesis treatments), and 145 binary variables indicating if a prosthesis treatment of a given type was claimed at least once during the last year, or not.

Constructing the Random Forest. We construct a random forest with 500 regression trees. Already after 100 generated trees the average out of bag error stops decreasing, but we want to make sure we have enough trees to estimate the proximities between patients. At each step we select 50 splitting candidates at random. This parameter calculated by the number of attributes divided by 3 is a standard choice for regression based random forest, and turned out to work well in our case. We use a minimum leaf size of 30. A lower leaf size has a somewhat higher accuracy (a MSE of 125.1 vs 126.8, where the mean of the target variable is 457 and the standard deviation of the target variable is 555), but since we are mainly interested in the proximity of points we use a higher leaf size so points end up on the same leaf more frequently.

Next, we obtain the proximity matrix from the 500 fitted regression trees. We calculate the local residual based outlier score with parameter $r = 1$. This corresponds to a global outlier score, comparing all patients in the dataset with each other.

Step 2: Finding Deviating Subgroups. After visual inspection of the distribution of the outlier scores, we decided to treat instances with an outlier score bigger than 2.5 as outliers, which in our case means that about 2 percent of our data is declared anomalous. This will be our target variable in the subgroup discovery process. Our explanatory variables will be the same 161 variables that were used before, however, the 145 binary variables are now replaced by continuous variables indicating the amount spent on a treatment. We discover the most interesting subgroups by applying a beam search strategy. It turned out that all interesting subgroups were found at depth two; increasing the depth search did not lead to the discovery of more interesting subgroups. Because of the small depth size we could use full enumeration of the search space to find the most interesting subgroups.

After investigating the subgroups found it turned out that the standard WRAcc score leads to too general subgroups. Therefore we applied the modified WRAcc score as interestingness measure: $\lambda_1 p(TS) - \lambda_2 p(T)p(S)$. After some trial and error we found the best setup with $\lambda_2/\lambda_1 = 2$, that gives more weight to subgroup accuracy than to subgroup size.

Step 3: Identifying Possible Fraudsters. We calculate the interestingness of the rule $subgroup \rightarrow entity$ for the top 100 subgroups, and measure the interestingness for all entities (prosthodontists). As an interestingness measure we use again the modified WRAcc score with $\lambda_2/\lambda_1 = 2$. We plot the results for the most interesting subgroup found during the subgroup discovery process in Figure 1. This is the subgroup with the highest modified WRAcc score for the rule $subgroup \rightarrow outlier$. The most interesting subgroup is described by 'Costs spent on upper prosthesis $\geq$ 1050 Euro'. The distributions of the 'Costs spent on upper prosthesis' variable are given in Figure 1. The most interesting rule found in the identification step is the subgroup with the highest WRAcc score for the rule $subgroup \rightarrow entity$. This is the rule 'Costs on Mesostructure: Two magnets/buttons $\geq$ 1500 Euro $\rightarrow$ ProsthodontistId = 223', for which the distributions are given in Figure 2.

The figures show that these prosthodontists have different claim behavior. The distribution of costs for upper prosthesis for the first prosthodontist has a fatter tail when

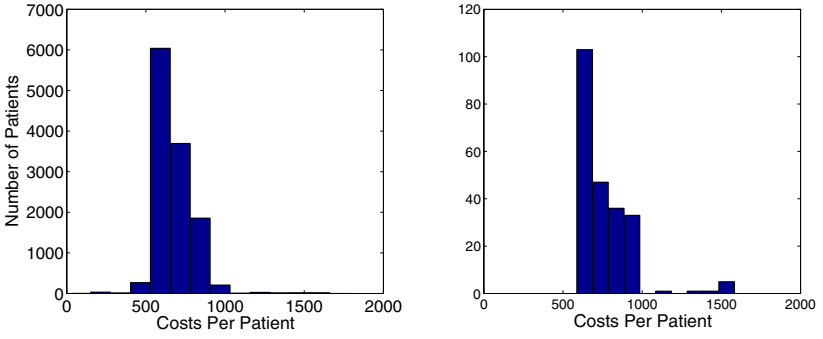


Fig. 1. Distribution of costs on treatment 'Upper prosthesis'. Only costs for patients that have received this treatment are given. On the left the distribution of the costs per patient of all prosthodontists are given, except for the suspicious prosthodontist for which the distribution is given on the right. It can be seen that the distribution of the suspicious prosthodontist has a fatter tail than the distribution of the other prosthodontists. This is exactly the outlier description that is provided by the subgroup 'Upper prosthesis $\geq$ 1050 Euro'.

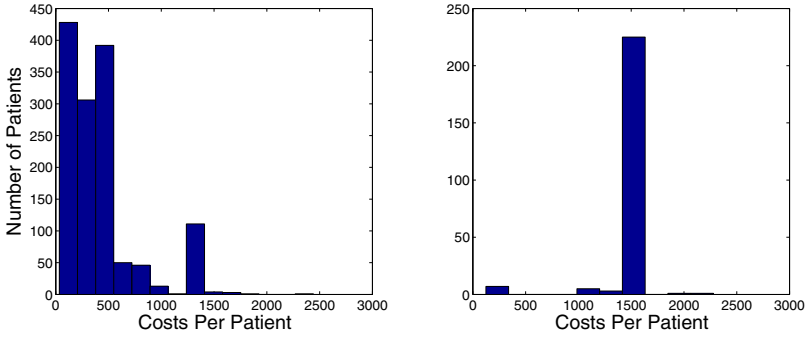


Fig. 2. Distribution of costs on treatment 'Mesostructure: Two magnets/buttons'. Only costs for patients that have received this treatment are given. On the left the distribution of the costs per patient of all prosthodontists are given, except for the suspicious prosthodontist for which the distribution is given on the right. It can be seen that the distribution of the suspicious prosthodontist is totally different from the others, and that the description 'Two magnets/buttons $\geq$ 1500 Euro' describes this difference in distributions.

compared to the costs of other prosthodontists. The distribution of costs of magnets/buttons of the second prosthodontist is totally different than the distribution of the costs of magnets/buttons of all other prosthodontists. Our approach measures the outlierness by using all attributes in the random forest, so the outliers described in both figures are not only outliers in the dimension that is plotted in the figures, but they also can not be 'explained' by other variables.

4.2 Results on Abalone Data

For illustrative purpose and reproducibility we show our results on the publicly available dataset of abalone shellfish from the UCI Machine Learning Repository. To demonstrate our approach, we use the 7 continuous variables from the data set: length, diameter, height, whole weight, shucked weight, viscera weight and shell weight. We will use the labels male, female and infant as entity labels later.

Step 1: Finding Outliers on Individual Level. For the abalone data set it makes no sense to use the regression tree approach to calculate the distance measure, because we have no target variable. However, our approach can still be used by plugging in a different kind of outlier score in Step 1. In this step we use the Local Outlier Factor [4] with parameter k equal to 100. After standardizing the columns of our dataset we calculate the outlier score.

Step 2: Finding Descriptions of Deviating Groups. After analyzing the distribution of the LOF scores we label outliers as those observations with a LOF Score of 4 or higher, which means there are 32 outliers in our data set. To automatically generate descriptions for regions containing many outliers, we perform subgroup discovery. As an interestingness measure we use the Weighted Relative Accuracy.

Step 3: Finding Outliers on a Group Level. Since we don't have a natural labeling of individual abalone in the data set, we use the gender of the abalone as an artificial label. From the top 20 subgroups found we investigate which of the three entities: Male, Female, or Infant, is the most 'suspicious'. It is found that male Abalone is the most

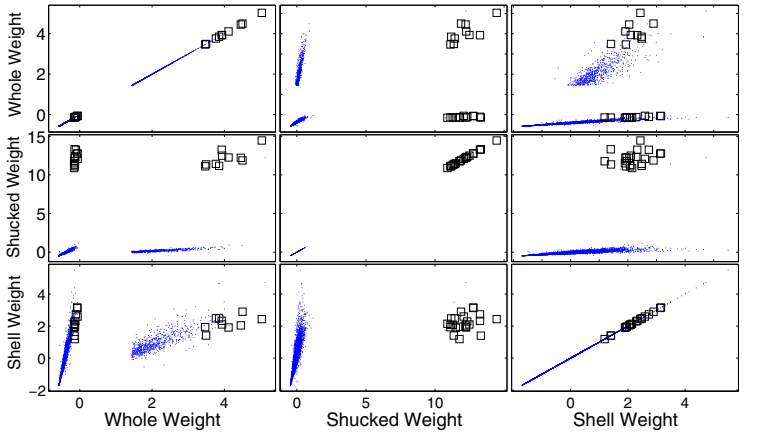


Fig. 3. Results on the abalone data. Abalone that are described by the subgroup are marked with black empty squares. Points outside the subgroup are marked with blue dots. It can be seen that black empty squares are indeed the outliers in these dimensions.

suspicious entity for subgroup ranked number 6 (according to the WRAcc measure). The description of the subgroup is:

Whole Weight $\geq$ -0.15848 $\wedge$ Shucked Weight $\geq$ 10.75 $\wedge$ Shell Weight $\leq$ 3.4925. The results corresponding to these dimensions are given in Figure 3. In total, 14 out of 32 outliers are described by this subgroup of size 23.

We can conclude that male abalone are the most 'suspicious' for this subgroup. Within the subgroup, 17 Abalone belongs to the entity Male, 6 of the Abalone belong to the entity Female, and 0 belong to the entity Infant. This causes the highest value for the WRAcc for the Male entity, because about 80 percent of this subgroup are of the Male entity, where in the total data set this is about 36 percent.

5 Conclusions and Further Research

In the paper we presented a new method for detecting and describing suspicious claim behavior in insurance data. The method makes an extensive use of Random Forests for measuring the outlierness of single records and Subgroup Discovery algorithms for detecting most deviating groups of outliers that can be described in terms of conjunctions of very few constraints. We demonstrated working of the system on a real-life dataset from an insurance company, generating statements like: "Dentist X is suspicious, because the group of his patients, exhibits characteristics A and B which significantly differ from others". We have illustrated our method, by finding two suspicious prosthodontists: one with a simple pattern of overcharging a specific treatment, and another one which was charging for another treatment in a completely different way than others. For the sake of reproducibility of our results, we have also applied our method to a publicly available Abalone dataset.

In future work we will compare this work with other methods related to subgroup discovery and exceptional model mining [8]. Future work will also focus on the inhomogeneity of patients across care givers. Some pharmacies or dentists are specialized in certain treatments, and are having different distributional characteristics because of this. This inhomogeneity causes that standard benchmarking techniques are not applicable anymore. Ideally we would like to develop a method that can automatically account for this inhomogeneity in the data by comparing only patients with the same characteristics and treatments with each other. Instead of using an outlier score which is based on costs, we would like to measure more general deviations in the underlying probability distributions.

References

1. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. SIGMOD Rec. 30(2), 37–46 (2001)
2. Bay, S.D., Pazzani, M.J.: Detecting group differences: Mining contrast sets. Data Mining and Knowledge Discovery 5(3), 213–246 (2001)
3. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001), doi:10.1023/A:1010933404324
4. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. SIGMOD Rec. 29(2), 93–104 (2000)

5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
6. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Comput. Surv. 38 (September 2006)
7. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: Proceedings of the 24th Ann. International Conference on Very Large Data Bases (VLDB), pp. 392–403 (1998)
8. Leman, D., Feelders, A., Knobbe, A.: Exceptional Model Mining. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 1–16. Springer, Heidelberg (2008)
9. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation forest. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE Computer Society, Washington, DC (2008)
10. Papadimitriou, S., Faloutsos, C.: Cross-Outlier Detection. In: Hadzilacos, T., Manolopoulos, Y., Roddick, J., Theodoridis, Y. (eds.) SSTD 2003. LNCS, vol. 2750, pp. 199–213. Springer, Heidelberg (2003)
11. Papadimitriou, S., Kitagawa, H., Gibbons, P.B., Faloutsos, C.: Loci: Fast outlier detection using the local correlation integral, p. 315. IEEE Computer Society, Los Alamitos (2003)
12. Pieters, B., Knobbe, A., Dzeroski, S.: Subgroup discovery in ranked data, with an application to gene set enrichment. In: Proceedings Preference Learning Workshop (2010)
13. Rousseeuw, P.J., Driessen, K.V.: A fast algorithm for the minimum covariance determinant estimator. Technometrics 41, 212–223 (1999)
14. Tang, J., Chen, Z., Fu, A., Cheung, D.: A robust outlier detection scheme for large data sets. In: 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (2002)
15. Hastie, T., Robert Tibshirani, J.F.: The Elements of statistical learning: Data mining, inference, and prediction. Springer, Heidelberg (2001)
16. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)
17. Yamanishi, K., Takeuchi, J.-I.: Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner. In: KDD 2001: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 389–394. ACM, New York (2001)
18. Zhang, J., Zulkernine, M.: Anomaly based network intrusion detection with unsupervised outlier detection. In: IEEE International Conference on Communications, ICC 2006, vol. 5, pp. 2388–2393 (June 2006)
19. Zhu, C., Kitagawa, H., Papadimitriou, S., Faloutsos, C.: Example-based outlier detection with relevance feedback. DBSJ Letters 3(2) (2004)
20. Zhu, C., Kitagawa, H., Papadimitriou, S., Faloutsos, C.: OBE: Outlier by Example. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 222–234. Springer, Heidelberg (2004)

Neural Networks Ensembles Approach for Simulation of Solar Arrays Degradation Process

Vladimir Bukhtoyarov[1], Eugene Semenkin[2], and Andrey Shabalov[2]

[1] Information Seurity Department, Siberian State Aerospace University, Krasnoyarsk, Russia
vladber@list.ru
[2] Department of System Analysis and Operations Research, Siberian State Aerospace University, Krasnoyarsk, Russia
eugenesemenkin@yandex.ru, shabalov-andrey@mail.ru

Abstract. Neural networks ensembles are powerful tools for solving modeling and time series forecasting problems. This approach is based on cooperative usage of neural networks for problem solving. The two major stages of the neural networks ensemble construction are: design and training of the component networks and combining of the component networks predictions to produce the ensemble output. In this paper developed evolutionary approach for neural networks ensembles automatic design is reviewed briefly. This approach is based on the operators of the well-known evolutionary algorithms and requires fewer parameters to be tuned providing more flexible and adaptive solutions. Results of the neural networks ensemble approach applying for modeling of spacecrafts arrays degradation are discussed.

Keywords: Evolutionary algorithms, neural networks, ensembles, simulation, solar arrays.

1 Introduction

Neural network ensemble is an approach based on joint usage of many trained neural networks for solving one problem [1]. It has been shown that generalization ability of neural network system can be improved by cooperation of a number of component neural networks which forms an ensemble. In general we need to execute two major steps to create an ensemble of neural networks. The first step is to design and train a number of component networks. The second step is to find a proper way to combine the component neural networks solutions.

As to design and training of the component promising approaches developed in different works are based on the genetic algorithms (GA) [5]. However in these approaches a great number of GA parameters need to be tuned. In many cases, seeking of proper settings for GA is time-consuming and requires large computational resources. An intuitive choice of GA parameters may reduce effectiveness. In this paper, a probability-based approach is proposed to avoid these difficulties. We call this method Probability Generator for Networks Structures (PGNS). There are few of parameters to be set in PGNS, so it is easier to understand and to utilize this method

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 186–195, 2012.

for different problems. As to combining of the component neural networks predictions, the most popular approaches are based on the plurality voting or the majority voting for classification problems and the simple or weighted averaging for regression problems [1], [4]. There are different ways to evaluate weights of the component networks [4], [6]. For example Jimenez uses weights determined by confidence of the component networks. Zhou, Wu and Tang utilize the genetic algorithm to find proper weights for each member of an ensemble [3]. Other diversity measure based and evolutionary based techniques are also available [7-9].

We develop another approach which utilizes basic operators of genetic programming to construct automatically formula which shows how to evaluate an ensemble prediction using component networks predictions. It involves different operations and math functions to combine the component networks predictions. It also automatically chooses component networks which are important for obtaining an efficient solution. This approach seems to be more flexible and adaptive than simple or weighted averaging or voting. So efficiency and accuracy of solutions obtained can increase significantly. We named this approach Genetic Programming based ENsembling (GPEN). Promising results were obtained by applying our approach for solving problem of designing the model for spacecraft solar arrays degradation prediction.

We applied GPEN approach for simulation of solar arrays degradation process. It was shown that the large solar flares have a significant impact on the current generated by photovoltaic arrays and on spacecraft systems functioning altogether [14]. That is why it is very important to obtain precise information about current state of spacecraft solar arrays but often obtaining of such information is not possible or very expensive. One of ways out is to simulate process of solar arrays degradation involving relatively cheap measurements of integral fluence of protons and electrons. In Section 3 results of simulation are discussed.

2 Comprehensive Evolutionary Approach for Neural Networks Ensembles Automatic Design

2.1 Probability Based Generator for Neural Networks Structures

A great number of widely-used methods for design of neural networks are based on genetic algorithms. Such methods utilize the GA operators to build neural network structure. Apparently the genetic algorithm collects and processes statistical information about neural networks structures during the run. Namely, it collects statistical information about the number of hidden layers, the number of neurons on each layer and the type of activation functions in each neuron of the network.

However GA does not process this statistical information in an explicit form. Instead, the information is collected and processed implicitly by the GA operators. The proposed herein PGNS (Probability based Generator of neural Networks Structures) method collects and processes the statistical information in an explicit form. This method allows us to avoid tuning many GA parameters. It is based on the replacement of GA operators by operators similar to those of EDA-like probability-based genetic

algorithm [10], [11]. We propose a neural network design paradigm based on estimated probability of the presence of certain kind neurons on the neural network hidden layers. By including of these estimated probabilities into our approach we can process the information about an optimal structure of the neural network in terms of mathematical statistics. This helps us to generate "good" neural networks. "Good" neural networks means that they are simple and have good generalization ability. For details see [12].

Experiments showed that the PGNS effectiveness is not lower than the effectiveness of the conventional method based on genetic algorithm. The advantage of PGNS method is that there are fewer parameters to be tuned. This is critical for solving complex problems, when tuning of many parameters is time consuming and requires large computational resources.

2.2 Genetic Programming Based ENsembling

The second major step of neural networks ensemble design procedure is the selection of networks which will be jointly used for solving the problem and finding out the way how to combine these networks predictions. Proposed herein GPEN method allows us to automatically select proper networks from a preliminary pool which is formed by PGNS method described above. Furthermore, the GPEN method allows us to automatically construct the ensemble prediction using the component networks predictions and different operations and mathematical functions.

We suppose that it is possible to find more effective solutions by forming more complex mixtures of the component networks predictions compared with simple or weighted average or voting. To find such mixtures we use the genetic programming approach. The GPEN method automatically constructs a program using the genetic programming operators [13]. This program shows how to combine the component networks predictions in order to get a good ensemble prediction. The GPEN also selects those networks which provide predictions to be taken into consideration by including them into the set of input variables of the program.

The GPEN constructs solutions with elements of terminal and functional sets being similar to the genetic programming approach. The main difference between the genetic programming and the GPEN is that GPEN terminal set includes predictions of the component neural networks instead of independent variables. The functional set consists of different operations and functions which specify the dependency between the ensemble prediction and the component networks predictions.

The GPEN algorithm is similar to the original genetic programming approach. It includes the following steps:

1. Identify the initial generation.
2. Execute the following steps until the termination criterion is satisfied:
   (a) Calculate fitness for each individual.
   (b) Select individuals for crossover by applying the selection operator.
   (c) Create the new generation by execution of the following steps:
      (i) Create the new individual by applying the crossover operator.
      (ii) Modify the new individual with the mutation operator.

(iii) Copy some initial individuals in the new generation.

3. The best found solution is declared as problem solution.

This basic scheme is often modified by introducing different methods for tuning of coefficients. For details see [12].

The ensemble prediction generated by the GPEN is a function of the component neural networks predictions:

$$o = f(o_1, o_2, ..., o_n).$$

Here o is the ensemble prediction, o_i is $i-th$ component the neural network prediction, n is the number of networks in the ensemble (or the preliminary pool).

Proposed approach has several advantages. The main are:

1. The GPEN method allows generating formulas for calculating the ensemble prediction more flexibly. These formulas define ways how to calculate ensemble predictions due to individual predictions of the neural networks which forms the ensemble.
2. The GPEN method allows executing "fine" tuning of neural networks interaction in the ensemble due to the inclusion of evolutionary algorithms for setting numerical coefficients in the method.
3. The GPEN automatically chooses the input variables of ensemble model (outputs of component networks), which are probably the most important. Although such selection procedure is absent in explicit form. If maximum number of the neural networks in the ensemble or maximum number of hidden units in all the networks are limited then special limitation procedures may be introduced in explicit form by modifying fitness function in the GPEN approach. For example, such procedures may be useful while designing neural networks systems with limited memory.

It is important that the proposed comprehensive evolutionary approach for neural networks ensembles design (it consists of PGNS and GPEN) doesn't rule out the possibility of the use of any of the specialized methods for neural networks training such as boosting, bagging and other similar methods. Similarly in the case of constructing an iterative procedure based on the proposed approach the objective function can be modified when designing and training component neural networks for example in accordance with the scheme of the negative correlation learning or other similar methods.

3 Numerical Experiments

It was shown that neural networks ensemble approach proposed above makes it possible to obtain precise simulation results for different kinds of modelling problems [12]. One of the benefits of this approach is that it generates ensemble almost automatically. Because of it this approach can be widely used by specialist in different fields and it is not necessary for them to be specialist if evolutionary computation or neural networks. For the purpose of solving spacecraft solar arrays degradation

problem we adapted our program system "*IT-PEGAS*" and now it is easy to simulate solar arrays degradation processes for users of this application.

As for the data set of solar arrays degradation problem it contains a number of observations of the spacecraft silicon-based solar arrays parameters and solar activity characteristics. Spacecraft is operating in the geostationary Earth orbit since 3/12/2000. The main objective of the study is to design a model for prediction of the electrical characteristics of the spacecraft solar arrays.

Concerning this fact we took into account the following components of space radiation to design the model for prediction of radiation-induced degradation of the solar arrays which operate in the geostationary orbit:

- Integral fluence of protons with energies < 3 MeV.
- Integral fluence of protons with energies < 10 MeV.
- Integral fluence of protons with energies < 100 MeV.
- Integral fluence of electrons with energies < 0.6 MeV.
- Integral fluence of electrons with energies < 2 MeV.

We also used the following characteristics of the spacecraft while designing the prediction model: duration of the spacecraft operation (a number of days) and coefficient of the spacecraft illumination intensity. The duration of the spacecraft operation was taken into consideration because it characterized the damages of the spacecraft from impacts of meteorites and ultraviolet radiation.

The performance of the spacecraft solar arrays was observed by measurements of the current generated by them.

The dataset was divided into two sets, a training set which includes data from the 1^{st} to the 169^{th} observation day and a test set with data from 170^{th} observation day. The training set is used for building the model. The test set is used only for forecasting to test the performance of the model.

To design the model for prediction of solar arrays degradation we used neural network ensemble approach described above which was implemented in program system called "IT-PEGAS". At the first stage we used PGNS method to form the preliminary pool of neural networks. Capacity of the preliminary pool was set to 20. As for candidate neural networks structures we decided to use quite simple multilayer perceptrons and set developed program system parameters by the following constraints: the maximum number of hidden layers was equal to 2 and the maximum number of units on the hidden layer was equal to 5. There was no a priori information which could help to choose the parameters listed above and we used constraints similar to those used in other applications of our neural networks ensemble approach.

While training candidate neural networks we used 7% relative error termination criterion that is why mean error of candidate neural networks in preliminary pool was close to 7% and was equal to 6.85%. Error of the "best" single neural network calculated using test sample was equal to 6.70%. For the model which output was calculated as mean of the outputs of the all 20 neural networks in preliminary pool error was equal to 5.54%. The average number of hidden units in preliminary pool neural networks is 6.

After the completion of preliminary pool the GPEN method was applied to form the ensemble of neural networks and for designing the formula for ensemble prediction calculation. We didn't involve any possible constraints while designing the

formula using the GPEN method, it means we didn't use the limitations concerning the number of networks in the ensemble and the total number of hidden (processing) units of ensemble members. We used this way for designing the prediction model because there were no constraints on computational complexity of the model. The obtained ensemble and formula were considered only in terms of precise computational procedure. We obtained the prediction model (consists of neural networks ensemble and formula which shows how to evaluate solution of the ensemble) with the following characteristics:

Ensemble consists of three neural networks. There are 4 hidden units in the first neural network and 6 hidden units in neural networks number 2 and 3.

The formula for calculating the ensemble solution (o – output) is shown below:

$o = (((((a \cdot x_2 \cdot x_3^2) \cdot (3 \cdot x_3+b)) \cdot (3 \cdot x_3+b+x_3+b^2))/((9+x_3 \cdot (x_3-x_1))+(9.381+((((x_1-x_2) \cdot 4.354)/(a \cdot 4.509))+x_3-x_1))))+x_2)$.

Here $a = x_1-x_2$, $b = 7.525*a$, x_1, x_2, x_3 – are the outputs of the neural networks forming the ensemble.

The error for test sample calculated is equal to 4.29%. The error was calculated according the formula (1).

We also obtain results for the GASEN approach. The error for test sample calculated using ensemble designed by the GASEN approach is equal to 5.23%. Relative improvement for our approach is about 20% (1% absolute improvement). For some problems such reduction of the modeling error is not significant but for highly precise space industry such improvements can bring a great economy of resources thanks to results obtained by more precise simulators.

So there was a significant improvement of the performance in comparison with the average single network performance and performance of the model which output is the mean of all 20 networks from the preliminary pool. It is notable that the ensemble which consists of only three neural networks is 23% more efficient than the model which consists of 20 neural networks.

To provide more objective comparison we obtained results for different algorithms widely-used for modelling and forecasting problems. We included in our test collection following algorithm and models:

1. Single neural network model. We use genetic algorithm to generate the structure of neural network model and to adopt model's weights. Multilayer perceptron was chosen as a basic topology of the neural network model. We carried out experiments with two different neural network models with two and three hidden layers. Maximum number of neurons on each layer was preset to 5. We didn't use neural networks with more hidden layers because the efficiency of such models was not improved in comparison with quite simple 2- and 3-layers solutions during test runs.
2. Symbolic model designed by applying of Hybrid Genetic Programming method. Hybrid Genetic Programming method is based on Koza's genetic programming method, but it includes additional steps which execute local search operations during modelling designing process.

Models mentioned below were obtained by executing algorithms implemented in the program system developed by authors. This program system allows to solve real-world problems and to carry out numerical experiments with test data sets.

While carrying out experiments on test data sets we use ANOVA technique to find out if there are significant difference between different algorithms and models. During designing models for solar arrays degradation process we try to improve the efficiency of the models as much as possible during some runs. The results are presented in Table 1. Figures 1-4 contain the measured values of current generated by spacecraft solar arrays along with the current values predicted by different models.

Table 1. Comparative results of simulation for solar arrays degradation problem

	Neural Network Ensemble Model (proposed GPEN method)	Neural Network Ensemble Model (GASEN)	2-layer perceptron model	3-layer perceptron model	Symbolic model
Evaluated error for the best model built by corresponding method, %	4.3	5.2	5.8	5.7	6.3
Variation of error obtained during test runs	0.027	0.024	0.327	0.419	0.381

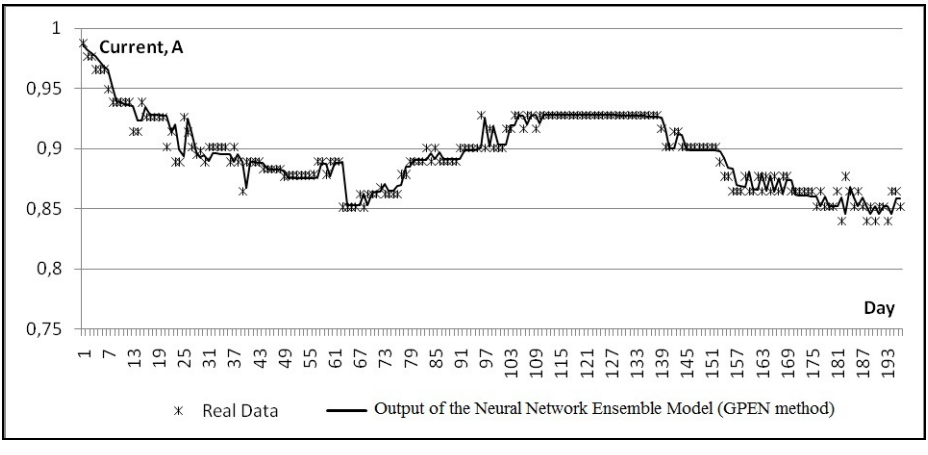


Fig. 1. Real data and predicted solar arrays current. Neural network ensemble model (GPEN method)

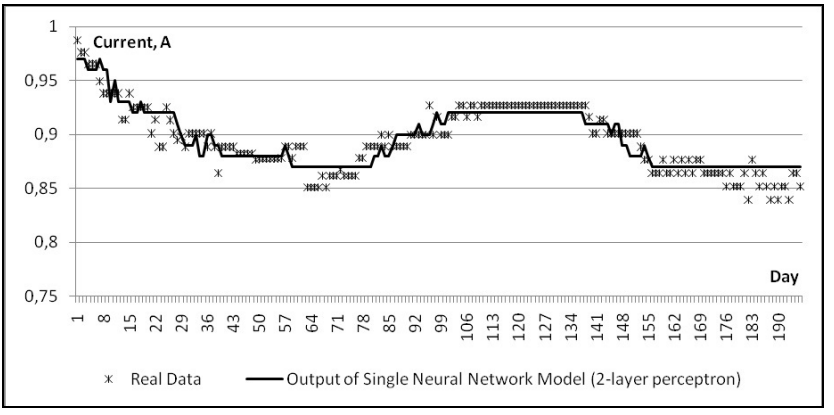


Fig. 2. Real data and predicted solar arrays current. Single 2-layer perceptron model

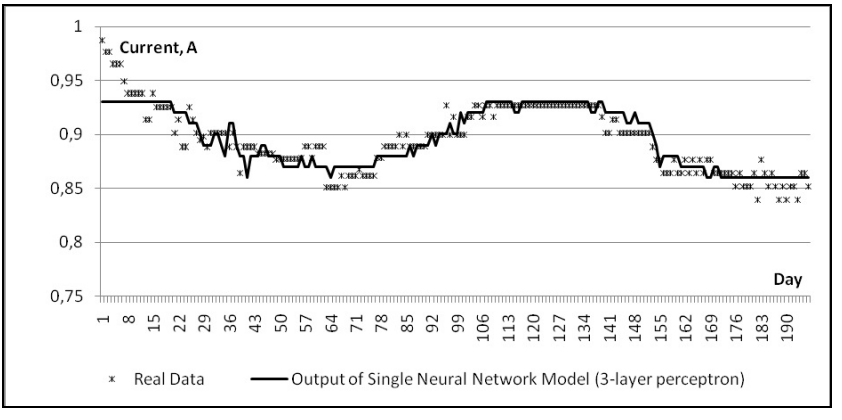


Fig. 3. Real data and predicted solar arrays current. Single 3-layer perceptron model

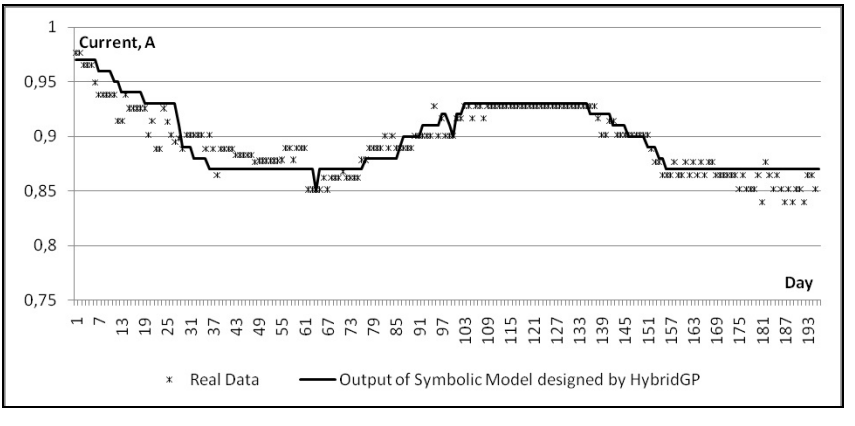


Fig. 4. Real data and predicted solar arrays current. Symbolic model developed by HybridGP

You can see that neural network ensemble model allows obtaining more precise results for our particular problem than any other model used in our study. We understand that the evaluated measurements efficiency of the algorithms and methods may depend on specific implementation. That is why we suppose that the proposed comprehensive approach (PGNS+GPEN) allows obtaining models which are "not less effective" than models developed using other modeling algorithms included in our study. Model designed for simulation solar arrays degradation process is accurate but we are looking forward to obtain more efficient results in future by tuning different parameters of the method which implements comprehensive approach for neural networks ensemble design proposed above. As for computational sources we used personal computer with 2.4 GHz dual core processor during the models development and simulations. Time needed to design one "good enough" component network is about 3 minutes, so time consumption is about 1 hour for one neural network ensemble.

4 Conclusions

In this paper we described briefly an approach (GPEN) which utilizes basic operators of genetic programming to construct automatically formula which shows how to evaluate an ensemble prediction using component networks predictions. The GPEN method provides high flexibility and adaptiveness of solutions. As contrasted to the widely-used voting and averaging method, the GPEN allows us to obtain different forms of the component neural network cooperation. Typically, the formulas for ensemble prediction evaluation gained with GPEN are more complicated than formulas resulted from averaging or voting. However, this complexity is compensated by more accurate solutions and higher generalization ability.

As for modelling of spacecraft solar arrays degradation fairly effective prediction model was obtained. The model is the ensemble which consists of three neural networks. These networks are quite simple multilayer perceptrons with only two hidden layers. Despite the simplicity of ensemble participants the model is quite precise: obtained relative error is not more that 4.3%. Now we are looking for more close interaction with engineers which will help us improve the prediction models and obtain more precise results.

We also are going to apply our approach for a wider range of real-world problems.

References

1. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence (12), 993–1001 (1990)
2. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley (2004)
3. Zhou, Z.H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial Intelligence 137(1-2), 239–263 (2002)
4. Perrone, M.P., Cooper, L.N.: When networks disagree: ensemble method for neural networks. In: Mammone, R.J. (ed.) Artificial Neural Networks for Speech and Vision, pp. 126–142. Chapman & Hall, New York (1993)

5. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading (1990)
6. Jimenez, D.: Dynamically weighted ensemble neural networks for classification. In: Proceedings of IJCNN 1998, Anchorage, AK, USA, vol. 1, pp. 753–756 (1998)
7. Sylvester, J., Chawla, N.: Evolutionary ensemble creation and thinning. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 5148–5155 (2006)
8. Dos Santos, E., Sabourin, R., Maupin, P.: Single and multi-objective genetic algorithms for the selection of ensemble of classifiers. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 3070–3077 (2006)
9. Santana, L., Silva, L., Canuto, A., Pintro, F., Vale, K.: A comparative analysis of genetic algorithm and ant colony optimization to select attributes for an heterogeneous ensemble of classifiers. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
10. Semenkin, E.S., Sopov, E.A.: Probabilities-based evolutionary algorithms of complex systems optimization. In: Proceedings of Intelligent Systems (AIS 2005) and Intelligent (CAD 2005), vol. 1, pp. 77–79. FIZMATLIT, Moscow (2005)
11. Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E.: Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms. Springer, Secaucus (2006)
12. Bukhtoyarov, V., Semenkina, O.: Comprehensive evolutionary approach for neural network ensemble automatic design. In: Proceedings of the IEEE World Congress on Computational Intelligence, Barcelona, Spain, pp. 1640–1645 (2010)
13. Koza, J.R.: The genetic programming paradigm: genetically breeding populations of computer programs to solve problems. MIT Press, Cambridge (1992)
14. Grigorieva, G.M., Kagan, M.B., Letin, V.A., Nadorov, V.P., Evenov, G.D., Hartov, V.V.: Analysis of Geostationary Spacecraft Solar Arrays Degradation from Solar Proton Flares. In: Space Power: Proceedings of the Sixth European Conference, Porto, pp. 725–730 (2002)

Using Genetic Algorithms to Improve Prediction of Execution Times of ML Tasks

Rattan Priya[1], Bruno Feres de Souza[2], André L.D. Rossi[2], and André C.P.L.F. de Carvalho[2]

[1] Computer Science Engineering
Indira Gandhi Institute of Technology, GGSIPU
New Delhi, India
bhasin.rattanpriya@gmail.com
[2] Computer Science Department
Institute of Mathematics and Computer Sciences, University of São Paulo
São Carlos-SP, Brazil
{bferes,alrossi,andre}@icmc.usp.br

Abstract. Experimental procedures associated with Machine Learning (ML) techniques are usually computationally demanding. An important step for a conscientious allocation of ML tasks into resources is predicting their execution times. Previously, empirical comparisons using a Meta-learning framework indicated that Support Vector Machines (SVM) are suited for this problem; however, their performance is affected by the choice of parameter values and input features. In this paper, we tackle the issue by applying Genetic Algorithm (GA) to perform joint Feature Subset Selection (FSS) and Parameters Optimization (PO). At first, a GA is used for FSS+PO in SVMs with two kernel functions, independently. Later, besides FSS+PO an additional term is evolved to weight predictions of both models to build a combined regressor. An empirical investigation conducted for predicting execution times of 6 ML algorithms over 78 publicly available datasets unveils a higher accuracy when compared with the previous results.

Keywords: Predicting execution times, Machine Learning, Meta-learning, Genetic Algorithms.

1 Introduction

Several Machine Learning (ML) techniques can be employed to extract valuable information from the massive amount of data that has been recently generated by a number of real-world applications. Data analysis in this context may involve very complex and time-consuming experimental procedures. For instance, the user may have to choose which approach is more appropriate for his data, trying to match the assumptions embedded in each algorithm to the specific requirements of the problem at hand. Conducting this kind of investigation typically relies on demanding empirical processes or expensive expert advice [4], being the former less biased but more computational intensive. The situation aggravates

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 196–207, 2012.

when one needs to perform an exploratory analysis to assess the behavior of a pool of algorithms on a batch of datasets.

To couple with the computational demand of ML applications, Distributed Heterogeneous Computing (DHC) systems [3] can be used. In a DHC environment, a suite of various resources with different capabilities, interconnected using high speed network, is coordinated to support the execution of a multitude of parallel and distributed tasks. A scheduling algorithm can allocate tasks into machines, thus optimizing the performance or cost-effectiveness of the system. In this scenario, an accurate prediction of the execution time of these tasks provides essential information for the scheduler, reducing the response time for the workload. Knowing the execution time and the resources usage of the tasks can be useful in two ways. First, users do not have to wait unexpectedly long for running their experiments. When these demanding tasks can be identified in advance, they can be scheduled without any resource contention. Second, accurate predictions may enhance the scheduling and thus minimize the overall execution time.

In our previous study [18] we explicitly casted the issue of predicting execution time as a Meta-learning problem [4], where a learning algorithm is used to relate characteristics of data and computational resources to the execution time of ML tasks. Among the meta-regressors compared there, Support Vector Machines (SVM) [21] yielded the most accurate predictions. Despite of those promising results, regression models were constructed using default SVM parametrization and full set of independent variables. However, one can argue that SVMs are sensitive to the choice of their parameter values as well as to the input features used to represent data [11], suggesting that improvements in the earlier work may be possible. In general, the first issue has been faced by either trial-and-error or Parameter Optimization (PO) techniques able to find out proper values for SVM's parameters [6]. The second issue has been extensively coupled with the use of Feature Subset Selection (FSS) techniques [12], which are able to readily discard irrelevant and redundant information that can adversely affect generalization performance of the model. Although both problems can be tackled independently, it is expected that better results are achieved with a joint FSS+PO process, since the choice of parameters is influenced by the feature subset employed and vice versa [11]. In the context of SVM for regression, FSS+PO combination has been commonly investigated using Genetic Algorithms (GA) [16,10,2], mainly because of their ability to search through a large solution space minimizing the risk of getting trapped in local minima [17].

In this paper we present a hybrid intelligent system [8,9] that employs a GA approach to simultaneously perform FSS+PO of SVMs aiming to improve predictions of execution times of ML tasks. Two scenarios were considered here. In the first one, SVMs with two common kernel functions were independently evolved and compared with our previous published results [18], achieving better performance. Motivated by the observation that predictions from different kernels are different, we hypothesized that a combined regressor would present more accurate results. Thus, in the second scenario, GA was employed to

perform FSS+PO of two different SVMs and to combine them to construct a single regressor.

The text is organized as following. Section 2 presents an approach for predicting execution time of ML tasks using Meta-learning. Section 3 provides an overview of SVMs for regression problems, defining the kernel functions used here and their parameters. In Section 4, the proposed GA for FSS+PO and model combination is described. In Section 5, the experimental setup used to evaluate the optimized models is described as well as the results achieved. Finally, in Section 6 we draw the concluding remarks and discuss future research directions.

2 Meta-Learning for Predicting Execution Times

In [18] we developed a Meta-learning based approach to predict the execution time of ML tasks. In that context, ML tasks were defined as the application of a ML algorithm over a dataset representing a real world problem. The solution proposed is illustrated in Figure 1. It consists of two main parts: a) the generation of the meta-dataset by extracting the characteristics of the datasets and computational resources and; b) the induction of a meta-learning model by applying a regression algorithm to the meta-dataset. For the first part, a learning algorithm is trained for different datasets, inducing different models, under a specific machine. The information about the time that a particular algorithm take to execute each dataset and the current status of the machine during the process is stored. Then, the characterization component extract the relevant features from the datasets and the machine status to create a meta-dataset. Each line of this meta-dataset is a meta-example and each column is an meta-attribute. The last column is a special meta-attribute, namely target, that corresponds to the execution times. For the second part, the meta-dataset is used to induce a meta-regressor model whose intent is to discover the relationship between the meta-attributes values and the execution times. Although the same approach is used in [18], in this paper we aim at improving previous results by performing SVM parameter optimization and feature subset selection (See Section 4). Therefore, our main focus here is to build more accurate regression models.

The characteristics that constitute the meta-attributes of the meta-dataset are extracted from the datasets and from the current status of the computational resources. The first set of meta-attributes corresponds to static measures whereas the second correspond to dynamic measures. In this study, the following widely employed measures were used to extract the datasets characteristics [4]: a) Log of number of instances; b) Log of number of attributes; c) Log of number of classes; d) Proportion of continuous attributes; e) Normalized class entropy; f) Average absolute correlation of all pairs of attributes; g) Average absolute correlation between target and attributes. The other measures are related to the current state of the computational environment, as follows: a) Current free memory available in the machine; b) Current CPU idle in the machine.

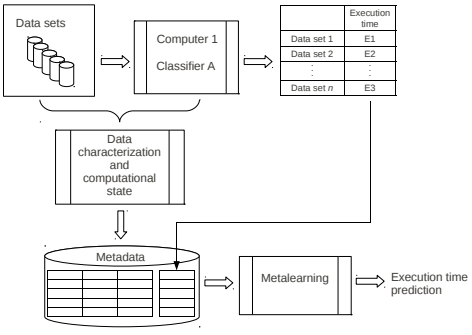


Fig. 1. Meta-learning approach for predicting execution time of ML algorithms

In order to evaluate the meta-regressors using the meta-dataset generated, we use the Leave-one-out Cross Validation (LOOCV) method. Other methods, such as the 10-fold cross validation, could be employed, but the LOOCV generates more reliable estimates, mainly when there are a small number of examples, which is the case here. The accuracy of the predictions was assessed by the Mean Absolute Deviation (MAD) measure. It is defined as the sum of the absolute differences between actual and predicted execution times divided by the number of test items in the LOOCV.

3 Support Vector Machines

Let $S = \{(\overrightarrow{x_1}, y_1), ..., (\overrightarrow{x_n}, y_n)\}$, where $\overrightarrow{x_i} \in \mathbb{R}^m$ and $y_i \in \mathbb{R}$, be a training dataset. Each $\overrightarrow{x_i}$ is a m-dimensional input vector and each y_i corresponds to the output associated to $\overrightarrow{x_i}$. The task of a regression algorithm is to learn a mapping $f : \overrightarrow{x} \mapsto y$ using data from S, such that it is general enough to predict y for new $\overrightarrow{x}$. Standard formulations of SVMs [21] handle this by approximating a linear function in the form of $f(\overrightarrow{x}) = (\overrightarrow{w} \cdot \overrightarrow{x}) + b$, with weight vector $\overrightarrow{w} \in \mathbb{R}^m$ and bias term $b \in \mathbb{R}$, both chosen according to the Structural Risk Minimization principle (SRM). SRM states that in order to minimize the probability of error on unseen data, $f(\overrightarrow{x_i})$ should have minimum deviation from y_i for all $\overrightarrow{x_i} \in S$ and $f(\overrightarrow{x})$ should remain as flat as possible, at the same time.

Deviations between predicted and actual output values are measured using the so-called ϵ-insensitive loss function $|y - f(\overrightarrow{x})|_\epsilon = \max\{0, |y - f(\overrightarrow{x})| - \epsilon\}$, which allows errors in prediction to be neglected with a tolerance range of $y_i \pm \epsilon$. If errors occur, loss is simply denoted by the distances between $f(\overrightarrow{x_i})$ and the ϵ tube, which are denoted by the slack variables ξ_i and ξ_i^*. By its turn, flatness of $f(\overrightarrow{x})$ is related to each component w_j of its weight vector and it is measured by the norm $\|\overrightarrow{w}\|^2$. So, in order to implement SRM, one needs to minimize both $\|\overrightarrow{w}\|$ and slack variables ξ_i and ξ_i^*. Formally, this goal can be translated into

a quadratic programming problem. Eq. 1 exhibits the primal formulation of ν-SVM [19], an extended version of the SVM initially proposed. It introduces a new parameter $\nu \in (0, 1]$, which is a constant that controls variation of ϵ.

$$\begin{aligned} \text{min: } & \tfrac{1}{2}\|\overrightarrow{w}\|^2 + C \cdot (\nu\epsilon + \tfrac{1}{n}\sum_{i=1}^{n} \xi_i + \xi_i^*) \\ \text{s.t.: } & y_i - (\overrightarrow{w} \cdot \overrightarrow{x_i}) - b \leq \epsilon + \xi_i \\ & (\overrightarrow{w} \cdot \overrightarrow{x_i}) + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{1}$$

where $C > 0$ is a regularization parameter used to penalize prediction errors during training. To make use of general optimization solvers and to explore nonlinear capabilities of SVMs, a dual formulation of Eq. 1 is usually preferred. Thus, with the introduction of Lagrange multipliers α and α^*, and the proper manipulation of the partial derivatives of the resulting Lagrange function, the ν-SVM problem can be stated as in Eq. 2.

$$\begin{aligned} \text{min: } & \tfrac{1}{2}\sum_{i,i=1}^{n} (\alpha_i^* - \alpha_i) \cdot (\alpha_j^* - \alpha_j) \cdot (\overrightarrow{x_i} \cdot \overrightarrow{x_j}) - \sum_{i=1}^{n} y_i \cdot (\alpha_i^* - \alpha_i) \\ \text{s.t.: } & \sum_{i,j=1}^{n} (\alpha_i^* - \alpha_i) = 0 \\ & 0 \leq \sum_{i,j=1}^{n} (\alpha_i^* + \alpha_i) \leq C \cdot \nu \\ & 0 \leq \alpha_i^*, \alpha_i \leq \tfrac{C}{n} \end{aligned} \tag{2}$$

From Eq. 2, on can note that parameters C and ν have a joint influence in the behavior of the model generated. The former imposes a trade-off between empirical risk and the model complexity. Very complex models tend to perform poorly on new data, due to overfitting during the training phase. On the other hand, models with too low complexity do not predict well even the training data, characterizing underfitting. The latter parameter controls the fraction of the so-called Support Vectors ($\overrightarrow{x_i}$ with $(\alpha_i - \alpha_i^*) \neq 0$) in the solution and also the fraction of outliers in training data [5]. Hence, to achieve the best generalization performance of SVMs, both parameters need to be carefully chosen.

At the optimum of Eq. 2, regression function can be expressed as in Eq. 3

$$f(\overrightarrow{x}) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \cdot (\overrightarrow{x_i} \cdot \overrightarrow{x})) + b \tag{3}$$

Predictions using the function in Eq. 3 can only deal with linear decision boundaries and thus may have restrict practical use. To overcome this limitation, one can replace dot product $(\overrightarrow{x_i} \cdot \overrightarrow{x_j})$ by a kernel function $K(\overrightarrow{x_i}, \overrightarrow{x_j})$ in the dual formulation of ν-SVM. Kernels implicitly map data from original input space to a higher dimensional feature space, allowing nonlinear relationships between inputs and outputs of data in S to be better modeled. Common kernels are able to construct polynomial, radial basis function and neural networks based regressors. In this study, we focus on kernels of the two latter types (namely

RBF and sigmoid), mainly because of their general purpose characteristics, low computational burden under some parameter settings and encouraging performance in a number of applications. Their definitions are depicted in Eqs. 4 and 5, respectively, for input data $\overrightarrow{u}$ and $\overrightarrow{v}$:

$$K(\overrightarrow{u}, \overrightarrow{v}) = exp(-\gamma \cdot |\overrightarrow{u} - \overrightarrow{v}|^2) \tag{4}$$

$$K(\overrightarrow{u}, \overrightarrow{v}) = tanh(-a \cdot (\overrightarrow{u} \cdot \overrightarrow{v}) + r \tag{5}$$

where γ is the width of the Gaussian function, a is a scaling factor of the data and r is a scalar controlling the threshold of the mapping. A proper choice of such kernel parameters are need for SVM to generalize well [14,15]. Choosing proper values for SVM's hyperparameters (C, ν and kernel parameters) is sometimes called Parameter Optimization.

4 Genetic Algorithms for SVM

Genetic Algorithms (GA) is a search and optimization technique based on genetics and natural selection [17]. GA deals with optimization problems by evolving a population of solutions with increasing quality. A traditional GA starts by randomly generating an initial population of individuals, named chromosomes, which encode feasible solutions to the problem at hand. During the evolutionary process, the whole population is evaluated using an appropriate fitness function. Next, the chromosomes produce offspring by the application of a crossover operator. This operator generates a new solution by exchanging features of multiple individuals (typically one pair). Then, a mutation operator is applied to the new generated individuals to randomly change parts of some of them to ensure that any point on the search space can be reached. Finally, a selection operation creates a new population by choosing individuals from the current population. The fittest individuals have a higher chance to be selected for the next population, thus leading the method towards good solutions. The advantages of this technique are it can handle a large search space, it is applicable to complex objective function and it can avoid trapping in local optimum solution.

Recently, GAs have been successfully employed for simultaneous feature selection and parameters optimization of SVM regressors in several contexts [16,10,2]. In the present paper, they are applied to improve the prediction of execution times of ML tasks. The main aspects of the approach investigated here are discussed next.

Solution Representation: GAs work on a suitable representation of the search space. To deal with the FSS+PO problem for SVMs, chromosomes present a hybrid binary/floating codification. The binary section is used to perform FSS, where each position is associated to a meta-feature in the Meta-learning problem (see Section 2). One or zero at a given position indicates the presence or absence of the corresponding meta-feature, respectively. The floating section encodes the parameters of the SVM to be tuned (see Section 3). If RBF Kernel is considered, each position of the chromosome corresponds to parameters C, ν

and γ. Instead, if sigmoid kernel is employed, the chromosome has four positions, which correspond to the parameters C, ν a and r. Finally, if a combined version of SVMs is optimized, then an extra weighting parameter β is considered, besides ordinary parameters for RBF and sigmoid kernels.

Initial Population Generation: Chromosomes in the first population are generated at random. Despite this kind of initialization seems to be naive, it aims at covering more regions at the solution space, without any bias for specific combinations of parameter. During the execution of the algorithm, less fitted chromosomes will gradually be discarded and focus in more promising regions can take place.

Selection Procedure: This process usually chooses the parents for reproduction, usually, emphasizing good solutions in the population. Here, the tournament selection method was employed [17], which randomly samples a number of individuals and selects the fittest one and it is applied every time a parent needs to be selected. This approach imposes an evolutionary pressure to the optimization process, allowing a faster, yet adequate, convergence.

Crossover: This genetic operator is used to guide the evolutionary process through potentially better solutions. This is performed by interchanging genetic material of two chromosomes in order to create two individuals that can benefit from their parents' fitness. Since we have a hybrid chromosome representation, each one requires a specific crossover operator. For the binary section, a uniform crossover is employed [17]. For its floating section, the BLX-α operator is used (see [17] and reference therein).

Mutation: The mutation operator is responsible for diversifying search directions over the solution space, minimizing the risks of premature convergence to local minima. Here, mutation was carried out by a gene based adaptive strategy [20] for the binary section of the chromosome and by a non-uniform operator [17] for the floating section.

Replacement: Replacement schemes determine how a new population is generated. The experiments performed here used a generational GA, where the parents are replaced by offspring at each generation. In order to assure the individual with best fitness survive, we adopted an elitism approach. Thus, the best combination of feature subset and SVM parameters is never replaced.

Fitness Function: One of the most critical and time consuming aspect of a GA is its fitness function. Here, the quality of a chromosome is defined by the MAD values obtained after the application of a 5 Fold Cross Validation scheme. For this, information regarding the selected meta-features and parameters of SVMs are firstly decoded and then data is split into five training and test sets for assessing performance of a SVM built with the current parameter settings. If two SVMs (one with each kernel) are simultaneously evolved, then the seven floating parameters are considered (see Solution representation). In this case, the additional parameter β is used to weight the predictions of both SVMs such that $P = \beta \cdot P_r + (1-\beta) \cdot P_s$, where P is the prediction of the new combined regressor,

P_r is the prediction of the SVM with RBF kernel and P_s is the prediction of the SVM with sigmoid kernel.

Termination Condition: In order to produce results in acceptable time, GA should finish its execution after a certain number of iterations. Here, it exits the searching process after a maximum number of iterations is reached.

Random Immigrant: It is a method that helps to keep diversity in the population, minimizing the risk of premature convergence [7]. It works by replacing the individuals whose fitnesses are under the mean by new randomly generated individuals. Random immigrant is invoked when the best individual does not change for a certain number of generations (here named re-start frequency).

5 Experiments

The GA used to perform FSS+PO for SVM is evaluated in this section. The main goal of the proposed approach is to provide accurate predictions of execution times of ML tasks. For such, regression models using SVMs with RBF and sigmoid kernel functions are genetically optimized. Three prediction schemes are considered. In the first one, regression using full set of meta-features and default values for the hyperparameters of SVMs takes place. Such naive regressors are called RBF and SIG, depending on which kernel funcion is used. In the second one, GA jointly evolves parameter values and feature subsets for SVMs with both kernels, independently. The resulting optimized regressors are called RBF_{opt} and SIG_{opt}. In the third one, a strategy named $(\text{RBF+SIG})_{opt}$ is assessed. It combines two SVMs with different kernels by evolving an additional parameter β, which weights the importance of each regressor for the final prediction (see details in Section 4).

5.1 Experimental Settings

In order to perform the experiments we used 78 classification datasets that were obtained from the UCI Machine Learning Repository. They were employed to train six classification algorithms available in WEKA, namely: K-Nearest Neighbor (K-NN_c), Support Vector Machine (SVM_c), Decision Tree (J48), Rules System (JRip), Bagging and Naive Bayes (NB), all with default parameter values. They represent distinct ML paradigms and have different inductive bias. Thus, they are expected to be representative of commonly used ML algorithms. Classification experiments were conducted using an Intel Core2duo E8400 3GHz computer with 4GB of RAM running Linux Ubuntu 10.4. Execution time of each classifier is the sum of five runs, and this value in seconds is used as the output values to be predicted. This setup was also employed in [18].

RBF and SIG regressors are induced using the default parameter values established by LIBSVM library. On the other hand, GA performs FSS+PO for RBF_{opt} and SIG_{opt}. In the case of $(\text{RBF+SIG})_{opt}$ approach, it also optimizes the weighting parameter β, which is used to combine the predictions of RBF_{opt}

and SIG_{opt}. The parameter settings of the GA used here are presented in Table 1. All values are either common found in literature or empirically defined. GA performs a search for the SVM parameters over $C = [0.001, 2]$ and $\nu = [0.2, 0.8]$. In the optimization process, the kernel parameters can take the values $\gamma = [0.001, 1]$ for the RBF kernel, and $a = [0.001, 1]$ and $r = [-3, -0.001]$ for the sigmoid kernel. For the simultaneous optimization of both kernels, GA searches for the β value over the range $[0.25, 0.75]$. Details about datasets' characteristics and algorithms' default parameter values can be retrieved from Supplementary material in www.icmc.usp.br/~bferes.

Table 1. Parameter settings of the GA

	Parameter	Value
GA	Number of chromosomes	20[1]
	Number of generations	50
	Selective pressure	2
Random immigrant	Percentage of individuals to be replaced	10%
	Restart frequency	10%
Crossover	Crossover probability	90%
	Exploitation and exploration relationship	0.5
Mutation	Mutation probability	5%
	Degree of dependency on the number of iterations	5
	Binary mutation range	[0.01,0.2]
	Binary mutation update value	0.005

5.2 Experimental Results

The quality of the estimation of the five versions of SVMs using the available meta-dataset for the six classifiers is shown in Table 2. It presents the MAD values of LOOCV for every regressor and the dMAD, which is obtained by predicting the test target using the median target of the training sets. For the genetically optimized models, MADs are averaged over 10 repetitions and standard deviations are presented in parenthesis, since GA is a stochastic method. The best value for each classifier is highlighted. Lower MAD indicates better performance of the regressor. As a general remark, its worthy noting that MADs for all methods are unevenly distributed throughout the classifiers. For instance, J48 and NB have lower values, indicating that their execution times are easier to predict. Conversely, K-NN_c and Bagging have the highest error rates. This can be partially explained by the presence of outliers in data, as suggested in [18].

As can be noted, RBF regressor is able to consistently generate more accurate predictions than dMAD. On the other hand, SIG performed worse than dMAD for all classifiers, except for JRip. Possible explanations for unsuitable performance of the latter regressor compared to the former rely on the intrinsic behaviour of its kernel or the hyperparameter values used in the experiments.

[1] For $(\text{RBF+SIG})_{opt}$ the number of chromosomes used was 30.

Table 2. MAD of the regression models for each classifier

	RBF_{opt}	SIG_{opt}	$(RBF+SIG)_{opt}$	RBF	SIG	dMAD
K-NN_c	3.947 (0.077)	**3.495** (0.364)	3.581 (0.227)	4.905	7.311	6.815
SMO	**2.860** (0.037)	3.023 (0.092)	2.923 (0.073)	3.180	4.679	3.786
J48	0.680 (0.015)	0.672 (0.041)	**0.644** (0.035)	0.823	1.221	1.125
JRip	2.284 (0.101)	**2.149** (0.203)	2.263 (0.102)	2.684	3.168	3.515
Bagging	3.527 (0.052)	**3.438** (0.141)	3.522 (0.118)	3.773	7.854	4.466
NB	**0.374** (0.017)	0.390 (0.011)	0.378 (0.011)	0.481	0.682	0.666

Moreover, the set of meta-features considered to describe the regression problem may not be the most adequate one. Both issues are simultaneously handled by GA in RBF_{opt} and SIG_{opt} methods. In Table 2 one can see that improvements were achieved by those regressors for all classifiers, supporting that joint FSS+PO is important for the prediction task at hand. In order to provide more evidence that the observed differences between regressors are significant, we applied a t-test with 95% confidence level as suggested by [1]. The null hypotesis to be verified is the equality between their errors. Since various statistical comparisons are being performed, Bonferroni correction is employed. As general remarks, we can state that[2]: a) comparing RBF and RBF_{opt}, the latter outperforms the former for K-NN_c, J48 and NB, and; b) comparing SIG and SIG_{opt}, the latter is better than the former for all classifiers. Confronting the optimized SVMs, one can see in Table 2 that SIG_{opt} has smaller MAD values than the RBF_{opt} for four out of six classifiers. However, the standard deviation of the former is usually higher than the latter. Applying the same statistical test as before points out that these regressors have comparable performance. This result can be explained by the fact that under a set of parameters sigmoid kernel behaves similarly to RBF kernel [15].

Despite the similar MADs, predictions of the two models do not coincide for every meta-example in our meta-dataset, as can be seen in Figure 2. It shows the relative difference between absolute deviation of values predicted by RBF_{opt} and SIG_{opt} for all 78 meta-examples. Interestingly, for all classifiers, the disagreement between the regressors are not strongly biased for any side, which means that one model do not outperform the other for every case. This observation motivated us to try combining predictions of both SVMs in the so-called $(RBF+SIG)_{opt}$. Results are still better than RBF and SIG, but there is no significant differences when compared with the other individually optimized SVMs. This can be understood by inspecting the fitness function of the GA when combined prediction is considered (see Section 4). Since it performs weight combination of regressors, when their estimates are either under or over the actual execution time, the new outcome will range between the best and the worst predictions. Nevertheless, considering plain MAD values in Table 2, the combined regressor is never worse than one of its competitors. Furthermore, it presents the lowest average error for J48.

[2] For GA based methods, predictions from the repetition with MAD closer to the average of 10 repetitions are used in the comparisons.

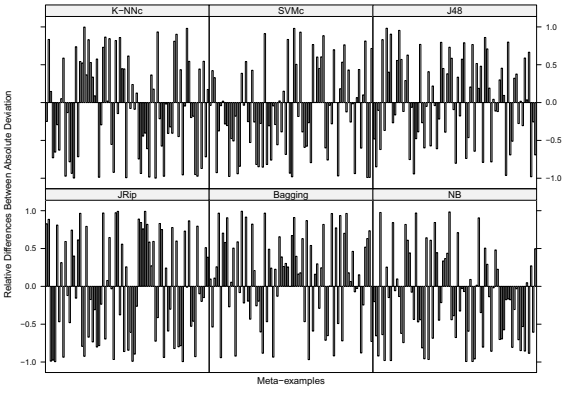


Fig. 2. Relative difference between absolute deviations of RBF_{opt} and SIG_{opt} for all 78 meta-examples

6 Conclusions

Recently, SVMs have been used as regressors for predicting execution times of ML tasks in a Meta-learning framework [18]. Although convincing outcomes were presented, they were based on full set of meta-features and default SVM parametrization. This scenario may be sub-optimal what motivated us to revisit that work in order to assess if the application of FSS and PO techniques would improve previous results. For such, we employed a GA to simultaneously tackle both tasks. Considering RBF_{opt} and SIG_{opt} methods, statistically significant improvements over their non-optimized counterparts were achieved for most ML algorithms. We also introduced a new regression approach that combines the outputs of SVMs with two kernels to make single predictions. $(\text{RBF} + \text{SIG})_{opt}$ performed similar to the other genetically evolved methods, presenting always intermediate MAD values.

As future works, we plan to investigate the meta-features selected by the GA methods as well as the most promising combinations of SVM's hyperparameters. We also intend to compare our approach to other well known optimization and search methods, like Particle Swarm Optimization [13], for FSS+PO of SVMs.

Acknowledgments. The authors would like to thank the financial support of funding Brazilian agencies FAPESP, CAPES, and CNPq, and the Department of Science and Technology, Government of India.

References

1. Bensusan, H., Kalousis, A.: Estimating the Predictive Accuracy of a Classifier. In: Flach, P., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 25–36. Springer, Heidelberg (2001)

2. Braga, P.L., Oliveira, A.L.I., Meira, S.R.L.: A GA-based feature selection and parameters optimization for support vector regression applied to software effort estimation. In: Proceedings of ACM-SAC, pp. 1788–1792. ACM (2008)
3. Braun, T.D., Siegel, H.J., Beck, N., Bölöni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B., Hensgen, D., Freund, R.F.: A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. J. Parallel Distrib. Comput. 61, 810–837 (2001)
4. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. Springer, Heidelberg (2009)
5. Chalimourda, A., Schölkopf, B., Smola, A.J.: Experimentally optimal nu in support vector regression for different noise models and parameter settings. Neural Netw. 17, 127–141 (2004)
6. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning 46(1-3), 131–159 (2002)
7. Congdon, C.B.: A comparison of genetic algorithm and other machine learning systems on a complex classification task from common disease research. Phd thesis, University of Michigan (1995)
8. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. Information Sciences 180, 2633–2634 (2010)
9. Corchado, E., Graña, M., Wozniak, M.: Editorial: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
10. Fei, S.-W., Sun, Y.: Forecasting dissolved gases content in power transformer oil based on support vector machine with genetic algorithm. Electric Power Systems Research 78(3), 507–514 (2008)
11. Fröhlich, H., Chapelle, O., Schölkopf, B.: Feature selection for support vector machines using genetic algorithms. International Journal on Artificial Intelligence Tools 13(4), 791–800 (2004)
12. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
13. Huang, C.-L., Dun, J.-F.: A distributed pso-svm hybrid system with feature selection and parameter optimization. Appl. Soft Comput. 8, 1381–1391 (2008)
14. Keerthi, S.S., Lin, C.-J.: Asymptotic behaviors of support vector machines with gaussian kernel. Neural Comput. 15, 1667–1689 (2003)
15. Lin, H.T., Lin, C.J.: A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Tech. rep., Department of Computer Science, National Taiwan University (2003)
16. Mejía-Guevara, I., Kuri-Morales, Á.: Evolutionary Feature and Parameter Selection in Support Vector Regression. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 399–408. Springer, Heidelberg (2007)
17. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs (2nd, extended edn.). Springer-Verlag New York, Inc., New York (1994)
18. Priya, R., de Souza, B.F., Rossi, A.L.D., de Carvalho, A.C.P.L.F.: Predicting execution time of machine learning tasks using metalearning. In: World Congress on Information and Communication Technologies 2011, pp. 1197–1203. IEEE Computer Society (2011)
19. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Comput. 12, 1207–1245 (2000)
20. Uyar, S., Sariel, S., Eryigit, G.: A Gene Based Adaptive Mutation Strategy for Genetic Algorithms. In: Deb, K., et al. (eds.) GECCO 2004, Part II. LNCS, vol. 3103, pp. 271–281. Springer, Heidelberg (2004)
21. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc., New York (1995)

Hybrid Artificial Intelligence Approaches on Vehicle Routing Problem in Logistics Distribution

Dragan Simić[1] and Svetlana Simić[2]

[1] University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
dsimic@eunet.rs
[2] University of Novi Sad, Faculty of Medicine, Hajduk Veljkova 1-9, 21000 Novi Sad, Serbia
drdragansimic@gmail.com

Abstract. Biological intelligence for modelling and optimization on vehicle routing problem of logistics distribution and supply chain management systems are presented in this paper. Logistics distribution is adaptive, dynamic, and open self-organizing system, which is maintained by flows of information, materials, goods, funds, and energy. The aim of this research is to summarize different individual bio-inspired methods, evolutionary computing, genetic algorithm, ant colony optimization, artificial immune systems, and to obtain power extension of these hybrid approaches. In general, these bio-inspired hybrid approaches are more competitive than the classical problem-solving methodology including improvement heuristics methods or individual bio-inspired methods and their solutions in logistics distribution and supply chain management applications.

Keywords: Logistics distribution, vehicle routing problem, bio-inspired models, genetic algorithms, artificial immune systems, hybrid artificial intelligence.

1 Introduction

The vehicle routing problem (VRP) has been one of the elementary problems in logistics ever since it was brought forward in [1]. Because of its wide use and high economic value it always attracts much attention. The VRP is a problem that has been studied for many years in subjects such as mathematics and computer science, because it is easy to describe but difficult to resolve. There are many solutions for some of those problems, but the VRP has several variations. Those variations depend on some time, spatial and non spatial restrictions.

Artificial intelligence techniques have demonstrated a capability to solve real-word problems in sciences, business, technology, and commerce. The integration of different learning techniques and their adaptation, which overcomes individual constraints and achieves synergetic effects through hybridisation or fusion, has in recent years contributed to a large number of new intelligent system designs [2]. In

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 208–220, 2012.

(1) bioinformatics and medical applications the theoretical and practical results to generate grammatical structures of a specific language; (2) fuzzy combiner harnesses the support values from classifiers to provide final response on their structure and the combination methods of decision templates; (3) the new supervised update rule utilizes data labels to achieve a better clustering result are mentioned in [3].

In study [4] the fundamentals and essential development issues of logic-driven constructs of fuzzy neural networks referred to as logic-oriented neural networks, constitute an interesting conceptual and computational framework that greatly benefits from the establishment of highly synergistic links between the technology of fuzzy sets and neural networks. Also, the principles of information granulation, logic computing and underlying optimization including those biologically inspired techniques, such as particle swarm optimization and genetic algorithms are shown. A case study which involves a set of techniques in classification tasks and study of nonparametric procedures useful to analyze the behaviour with respect to a set of algorithms is proposed and developed in [5]. The review [6] outlines some current approaches of fuzzy models which is implemented in the terms of potential benefits gained in logistics domain in order to mitigate the uncertainty and risk of the business turbulent environment and global world financial crises. On the other hand, new mechanical engineering technology had a direct impact on development of those constructions of cargo motor vehicles and trailers which would be able, by their dimensions, to use to the fullest their potential capacity and simplify loading [7].

This paper is interested in biologically inspired computing, a branch of natural computing which develops algorithms inspired by nature to solve highly complex problems, in particular the problems that cannot be addressed in a satisfactory way by a traditional approach. Under this paradigm, algorithmic models of processes observed in nature are developed and implemented on computers to explore solution spaces. There is a growing interest in bio-inspired algorithms, and the representatives of this trend have been applied to a large variety of problems, including optimization.

Presently, globalization and informatization are developing direction in the logistics industry, and logistics distribution becomes more important in the supply chain. In the process of logistics distribution, because of the quantity of customers and the complexity of urban communication lines, the problems of how to form a perfect line and how to make collection of the distribution and the distribution line effective are not only the characteristics of distributing transportation but a task of great difficulty as well. So, effective vehicle routing program design and an attempt of reduction of the number of vehicles and the travelling distance are very practical questions. Practical application of VRP include logistic distribution, drawing up a bus route, mail and newspaper delivery, time schedule arrangement for aviation and railway and the collection of industrial waste [8].

The rest of the paper is organized as follows. Section 2 introduces a brief review of VRP and extensions. Section 3 presents some classical problem-solving methodologies. Then, Sections 4 introduces the main bio-inspired algorithms, Evolutionary Algorithms (EA), Ant Colony Optimization (ACO), and Artificial Immune Systems (AIS), respectively, and describes their application to vehicle

routing problems. Section 5 is devoted to hybrids that combine ideas from different bio-inspired algorithms. Concluding remarks and future work follow in Section 6.

2 The Vehicle Routing Problem and Extensions

The aim of vehicle routing is to determine optimal collection or delivery routes for a fleet of vehicles in a transportation network [9]. The VRP is non-deterministic polynomial-time hard (*NP-hard*) problem which generalizes classical travelling salesman problem (TSP) by requiring both, an assignment of vertices to vehicles and a sequencing of these vertices within each vehicle route to obtain a solution.

2.1 Travelling Salesman Problem

Given a graph with a number of vertices and the cost associated with each arc, the TSP looks for the least cost tour that visits each vertex one time only. The sequence in which the salesman visits different cities is called a *tour*. A tour should be such that every city on the list is visited once and only once, except for the return to the city of origin. The goal is to find a tour that minimizes the total distance the salesman travels, among all the tours that satisfy this criterion.

For n cities to visit, let X_{ij} be the variable that has value 1 if the salesman goes from city i to city j and value 0 if the salesman does not go from city i to city j. Let d_{ij} be the distance from city i to city j. The travelling salesman problem (TSP) is stated as follows [10]. Minimize the linear objective function:

$$\textit{Min} \qquad z = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}$$

$$\textit{Subject to} \qquad \begin{cases} \sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1, \\ \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1. \end{cases}$$

In vehicle routing problems, side constraints are introduced to limit the number of vertices that can be visited with a single tour, thus leading to solutions that cover all vertices with different routes that start from end of a particular vertex which is called the depot. These problems are used to model various real-world applications.

2.2 Vehicle Routing Problem

The VRP generalizes the TSP, a canonical combinatorial optimization problem that has been widely studied in the literature [11]. The VRP can be described as in Fig. 1. Suppose there are M vehicles each of which has a capacity Q, and N customers who must be served from a certain depot. The goods each customer asks for and the distance between them are known in advance. The vehicles start from the depot,

supply the customers and go back to the depot. It is required that the route of the vehicles should be arranged appropriately so that the least number of vehicles is used and the shortest distance is covered, and at the same time the following conditions must be met: (1) The total demand of any vehicle route must not exceed the capacity of the vehicle; (2) Any given customer is served by one, and only one, vehicle; (3) Customer delivery can not be split up.

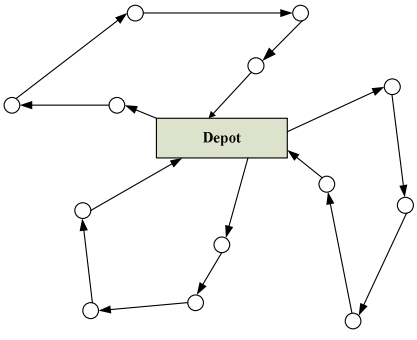


Fig. 1. Vehicle Routing Problem – General View

The vehicle routing problem can be formally stated in following way. Let $G = (V, A)$ be a graph where $V = \{1, 2, \ldots, n\}$ is the vertex set and A is the arc set. Vertex 1 is the depot for a fleet of identical vehicles of capacity Q that collects a demand q_i at each vertex (customer) $i \in V - \{1\}$. A non negative cost c_{ij} is also associated with every arc $(i, j) \in A$, $i \neq j$. This cost is often interpreted as a distance or a travel time, depending on the context. Unless otherwise stated, the following is assumed; that problem is symmetrical, that is, $c_{ij} = c_{ji}$ for ever arc (i, j). The problem then determines a set of the most cost effective vehicle routes so that: (1) each vertex, apart from the depot, is visited exactly once by exactly one vehicle to serve its demand; (2) all vehicle routes start and end at the depot; (3) the total demand on each route does not exceed the vehicle capacity.

Sometimes, a fixed cost is associated with each vehicle. The objective is then aimed at minimizing a weighted sum of fix costs and travel costs. A maximum distance or time constraint can also be considered, in addition to the capacity constraint, for each vehicle route.

A large number of variants are depending on the objective function to be optimized and the types of constraints to be satisfied. These variants are reviewed in the following.

2.3 Vehicle Routing Problem with Time Windows

Vehicle routing problem with time windows (VRPTW) in extension of VRP where time interval or time window $[a_i, b_i]$ constrains the beginning of service at each vertex $i \in V - \{1\}$. There is also a window $[a_1, b_2]$ at the depot to constrain the start time and end time of each vehicle route. The upper bound b_i at vertex i can be either hard or

soft constraint. If the vehicle arrives late there are penalties incurred in the objective. A waiting time is introduced in the route schedule when the vehicle arrives before the lower bound a_i. In most papers, a hierarchical objective is considered. The first objective is to minimize the number of vehicles and to minimize, for the same number of vehicles, the distance, travel time or scheduling time (travel time + waiting time). The VRP with time Deadlines (VRPD) is special case of VRPTW where only a time upper bound is associated with each vertex.

2.4 Vehicle Routing Problem with Back-Hauls

In this special situation of VRP, vehicle routing problem with back-hauls, the set of vertices is partitioned into two subsets: line-haul vertices for which the demand is delivered from the depot and back-haul for which the demand is picked up brought back to the depot. It has been quickly recognized that substantial cost saving can be achieved by allowing empty vehicles to pick up inbound products when they return to the depot. In the grocery industry, for example, line-hauls could be supermarkets where goods are delivered while the back-hauls could be grocery suppliers. In the classical VRPB, there is a strict precedence relationship between the line-haul and backhaul vertices, that is, all line-hauls must be visited before the backhauls. Without this constraint, goods could be picked up while other goods would not have yet been delivered, thus potentially leading to rearrangement of goods inside the vehicle. In other variants, this constraint is relaxed to allow a vehicle to pick up goods if the capacity utilization is sufficiently low. Finally, in the mixed VRPB, line-haul and back-haul vertices can be freely mixed as long as the capacity constraint is satisfied. When time windows are associated with the vertices, the VRPBTW is obtained.

2.5 Other Vehicle Routing Problems

Without being exhaustive, it could be possible to mention following similar problems:

- *multiple-depot* problems where each vehicle can start or end its route from (to) a different depot
- *mixed fleet and size* problems where the fleet size must be determined based on different types of vehicles with different characteristics (e.g., capacity)
- *periodic vehicle routing* problems where routes are determined over a horizon that spans a number of periods and where each vertex must be visited at a given frequency within this horizon
- *location-routing* problems where strategic decisions about the location of different facilities are taken concurrently with the determination of vehicle routes
- *inventory routing* where each vertex has an inventory and delivery routs are determined to replenish the inventory so as to avoid inventory shortage

There are also a number of vehicle routing problems where arcs must be visited instead of vertices [12]. Arc routing problems are useful to model applications where the customers are located along the streets of the transportation network, as in postman delivery routes.

3 Classical Problem-Solving Methodologies

In this section, some classical methodologies for solving vehicle routing problems are briefly reviewed. These methodologies can be divided into two main classes: exact methods and heuristics. Exact methods are divided into (1) tree search methods: branch-and-bound, (2) dynamic programming and (3) integer programming-based methods. In the case of heuristics there is a distinction between construction heuristics, improvement heuristics and meta-heuristics.

3.1 Construction Heuristics

In this category, there are pure construction heuristics and two-phase construction heuristics. Pure construction heuristics are:

- *Insertion heuristics* is based on iteration, where a vertex is selected among all unvisited vertices and a feasible, most cost effective, insertion place between two consecutive vertices is looked for. This process is repeated until all vertices are visited.
- *Saving heuristics* is a problem-solving approach based on: start from a configuration where each vertex is visited by an individual route connected to the depot, two routes are selected at each iteration and merged together. The merging process is guided by the savings measure. These savings are thus calculated for every pair of vertices and sorted in non increasing order. Pairs of routes are then merged together, by using the sorted list savings to guide the merging process until it is not possible to merge two routes without violating the side constraints.

3.2 Improvement Heuristics

Once an initial solution has been constructed, it can be further improved by local search heuristics. In this case, a class of modifications to the current solution is defined to generate a neighbourhood of solutions. The best solution in this neighbourhood becomes the new current solution if better than standing current solution. The procedure is repeated until there is no better solution in the neighbourhood of the current solution. Some basic modifications, based either on nodes or arcs in a solution, are following:

- *vertex move*: a vertex is removed from the solution and inserted at another place
- *vertex swap*: two vertices exchange their position in the solution
- *arc exchange*: r arcs are removed from the solution and are replaced by r new arcs to produce a new valid solution. The most widely used exchanges of this type involve $r = 2, 3$ arcs and are called 2-opt and 3-opt [13].

Also, these modifications can be applied to each route individually, to modify the sequence of vertices in the route, or they can be applied to several routes at once to modify the vertex assignment.

3.3 Metaheuristics

The emergence of metaheuristics for solving difficult combinatorial optimization problems, including VRP, is one of the most notable achievements in operations research in the last two decades. Metaheuristics offer global search strategies for exploring the solution spaces that are then adapted to a particular class of problems. Some metaheuristics are high-level abstraction of the optimization processes observed in nature and will be describe in the following sections, starting with EAs.

4 Evolutionary Algorithms

Evolutionary algorithms (EA) are stochastic search methods that operate on a population of solutions by simulating, at high level of abstraction, the evolution of species observed in nature. Some results obtained with Evolution Strategies (ES) an evolutionary approach developed in the 60's and 70's will be mentioned [14]. Like genetic algorithms, ES evolves a population of solutions through mutation and recombination (crossover). One distinctive feature is the concurrent evolution of strategy parameters which typically are the properties of the mutation operators, like mutation step size.

In [15] ES to the VRPTW is first applied. The mutation operator is based on local search heuristics and the mutation step size in the number of modifications to the current solutions. An additional strategy parameter associated with mutation operator specifies the objective to be favoured, either minimization of the total distance or minimization of the number of vehicles.

4.1 Genetic Algorithms

Genetic algorithms (GA) [16] were the first evolutionary algorithms to be applied to combinatorial optimization problems. In a GA, a population of solutions evolves over a number of generations through the application of operators like selection, crossover and mutation, which mimic the corresponding genetic process observed in nature.

4.2 Ant Colony Optimization

In their search for food sources, ants initially look around in a random manner. When they find one, they come back to the nest, laying down an aromatic substance on the ground, known as pheromone. The amount of pheromone is related to the quality of the food source, as determined by the quantity of food and its distance from the nest. The ants that come next will thus search in a less random fashion, as they will be attracted by the pheromone trails. More ants will be attracted to these paths with more pheromone, which in turn will lead to more and more pheromone laid down on these paths. Ultimately, all ants will be attracted to the best path.

Ant colony optimization (ACO) is motivated from the way real ants find good paths to food sources, in particular the indirect communication scheme through

pheromone trails that lead to the optimization, a phenomenon called stigmergy. Different ant-based metaphors are reported in the literature, starting from the original ant system (AS) [17] to more recent variants, like ant colony system (ACS). Although ant-based systems have first been tested on the TSP, many other combinatorial problems have since been addressed. The basic AS algorithm for solving the TSP is described in [18].

Another approach is the Max-Min ant system (MMAS) described in [19]. Its main characteristic is the exclusive reinforcement of the best tour found at the current iteration or the best tour found since the start of the algorithm. To prevent convergence to suboptimal tours due to the accumulation of pheromone on the same edges, MMAS then forces the amount of pheromone on every edge to lie within an interval τ_{min} and τ_{max} as the parameters.

Ant-based systems have also been applied to the VRPTW. In [20], the proposed multiple ant colony system (MACS) defines two ant colonies: ACS-VEI that minimizes the number of vehicles (main objective) and ACS-TIME that minimizes the total travel time (secondary objective). Both colonies use the standard of the ACS framework, except that during the construction of a solution each ant has the choice to go either to a vertex that does not violate the capacity and time window constraints or to return to the depot. ACS-VEI searches for a solution that maximizes the number of visited vertices with a fixed number of vehicles. When a feasible solution that includes all vertices is obtained, ACS-TIME then constructs solutions with that particular number of vehicles, while trying to minimize the total travel time. Also, ACS-VEI is restarted with one vehicle less the before. This is repeated until the number of vehicles cannot be reduced anymore. This approach is very similar to the one used by the EA and presented in [21].

4.3 Artificial Immune Systems

Artificial immune systems (AIS) is a recent bio-inspired algorithm motivated from the mechanisms used by our organism to defined itself against foreign micro-organisms, like viruses and bacteria [22]. In the natural immune system, molecular patterns at the surface of micro-organism, known as antigens, lead to the proliferation of lymphocytes that are able to produce antigen-specific antibodies. This phenomenon is known as clonal selection (clonal expansion). Since each lymphocyte has distinct antigen specificity and the organism needs to react to different antigens, the number of lymphocytes that can differentiate into effector cells able to produce antibodies for each specific antigen is limited. Thus, it is primordial for the organism to develop a good repertoire of lymphocytes with different specificities. Furthermore, the differentiation process is not of the ”*all-or-nothing*” type, so that antibodies typically show a variable degree of antigen affinity or specificity.

In AIS developed for optimization purposes, the antigen affinity corresponds to the solution value. Then, different mechanisms are observed to generate good solutions. For example, artificial clonal selection involves the selection of solutions with the best objective values (high affinity), the reproduction of these solutions to obtain a number of copies or clones and mutation of these clones to generate improved

solutions. As opposed to classical genetic algorithms, where mutation is a random process which is applied at low rates, the mutation operator in AIS is controlled and its rate is inversely proportional to the solution value. It is also typically applied at high rates to increase population diversity. These AIS principles are often integrated within GAs to allow a better management of the population of solutions.

This is the case for example in [23], where an additional operator motivated from AIS is applied after crossover to allow the proliferation of high quality solutions in the population. This approach is applied for solving an electricity distribution network problem. The problem is modelled as a multiple depot vehicle routing problem, where the depots correspond to high/medium voltage stations and the other vertices to medium/low voltage stations. In [24], an immune-based operator is used within a GA to solve the VRPTW. Basically, the operator is able to recognize good patterns in solutions, such as sub-paths than link close vertices. It then uses this information to improve solutions.

5 Hybrid Vehicle Routing Problem Models and Applications

The hybridization of intelligent techniques, draws from different areas of computational intelligence, and has become prevalent because of the growing awareness that they outperform individual computational intelligence techniques. In a hybrid intelligence system, a synergy combination of multiple techniques is used to build an efficient solution to deal with a particular problem [25]

An increasing number of bio-inspired algorithms does not rely on a single search strategy but rather combine various algorithmic ideas. These approaches are referred to as hybrids. The motivation behind hybrids is usually to exploit the strengths of a number of individual search strategies in order to obtain a more powerful problem-solving approach. For example, a decoder system can be viewed as the hybrid of a GA and a construction heuristic. The integration of local search-based mutation operators within GAs, known as memetic algorithms [26], is another type of hybrid. Likewise, some AIS concepts have been integrated within GAs to make them more robust.

In [27], an algorithm called AGES combines guided local search (GLS) [28] and ES in an iterative two-stage procedure to address both the VRP and VRPTW. GLS introduce modifications into the objective function through penalties when the search gets trapped in a local optimum. Here, the penalty counter of one arc in the solution is incremented by one, each time a local optimum is reached. The arc is selected on the basis of its length (a long arc is more likely to be penalized) and current penalty counter (an arc with a high penalty counter is less likely to be penalized again). In the first stage, it controls the objective functions of an ES. In this ES, the ruin and recreate principle is at the core of the mutation operator. That is, a number of vertices are removed from the current solution and reinserted at least cost. Refinements to this algorithm are proposed in [29] with excellent results reported on classical VRP and VRPTW benchmark instances. On the VRP data sets of [30], in particular, this approach has produced the best average solution values when compared with state-of-the art methods.

A multiple vendor transportation problem with time and cost criteria and proposed AIS algorithms strengthened by a fuzzy logic controller (FLC) to solve the multicriteria problem is discussed in [31]. AIS works as an ES algorithm to find out the Pareto optimal front, whereas FLC is implemented to change the hyper mutation rate adaptively on the basis of the fitness values at each iteration.

In [32], a two-stage vehicle routing problem of distribution centres and customers is considered. To solve the problem for selecting some potential places as distribution centres in order to supply demands of all customers with minimum opening cost plus shipping cost, a GA and an AIS algorithm are developed, and the results showed that the AIS algorithm exhibits robust performance and improvements in large size problems in comparison to situations when only GA is used.

Some other bio-inspired methods are also used in the approaches of hybridization of bio-inspired methods, in combination with previously mentioned methods (FLC, GA, AIS) and with methods such as local search [33], tabu search [34], and simulated annealing [35].

It can be seen, and noticed, that new innovative informational/computational paradigms, such as chaotic systems, quantum informatics, and DNA computing, provide valuable inspiration to create new heuristics for complex optimization problems including a host of NP-hard problems. Thus, the extensions of current bio inspired methods based on these new paradigms are expected to achieve dramatic improvement on computational performance. For example, chaotic sequencing and local search operations have been successfully applied for helping evolutionary algorithms avoiding premature convergence effectively [36]. Also, quantum-inspired evolutionary algorithms are regarded to the complex interaction between quantum computing and evolutionary algorithms [37] and have been applied in some logistics distribution and supply chain management optimization problems in very recent research [38]. To improve presented experimental results one innovative research area that uses fuser as part of multiple classifier system, which is discussed in designing fusers on the basis on discriminants – evolutionary and neural networks models [39], could form part of future research in hybrid system for vehicle routing problem.

It is important to note that swarm-based methods and AIS are not yet mature and thus are expected to gain more research interests. With the increasing importance and complexity of vehicle routing problem in logistics distribution systems, researches are facing the challenges to promote the performance, reliability, and scalability of supply chain management problem solving methods.

6 Conclusion and Future Work

Today's logistics distribution and SCM systems have to deal with ever-changing markets and intrinsic structural complexity emerging from virtually infinite number of interacting entities. Therefore, the community requires effective artificial intelligence methods and tools for modelling and optimizing large scale complex logistics distribution and SCM systems. The paper has reviewed the recent development of main bio-inspired methods in VRP to solve logistics distribution applications. Typical illustration is addressed for evolutionary algorithms, genetic algorithms, swarm-based intelligent algorithms including ant colony, and other bio inspired methods like artificial immune systems. It is clear that evolutionary algorithms and their hybrids have shown the best performance up to now. Over the last decade, bio inspired

methods have experienced a rapid growth and have successfully been applied to the design and optimization of highly complex systems such as logistics distribution systems. Finally, hybrid systems for vehicle routing problems are not yet mature fields of research and these approaches certainly show great promise for the future work.

Acknowledgment. The authors acknowledge the support for research project TR 36030, funded by the Ministry of Science and Technological Development of Serbia.

References

1. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Management Science 6(1), 80–91 (1959)
2. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. Neurocomputing 72(13-15), 2729–2730 (2009)
3. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
4. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
5. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
6. Simić, D., Simić, S.: A review: Approach of fuzzy models applications in logistics. In: Burduk, R., Kurzyński, M., Woźniak, M., Żołnierek, A. (eds.) Computer Recognition Systems 4. AISC, vol. 95, pp. 717–726. Springer, Heidelberg (2011)
7. Kudumovic, D., Mujevic, M., Sukic, C.: New trends and ideas in the application of multi-modal transportation systems. Technics Technologies Education Management 6(2), 241–246 (2011)
8. Huanglan, C.C.: The mode of vehicle routing problem and the of brain power heuristic algorithm. Computer Development and Application 16, 2–5 (2003)
9. Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research 59, 345–358 (1992)
10. Rao, V.B.: Neural networks and fuzzy logic. M&T Books, IDG Books Worldwide (1995)
11. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B.: The traveling salesman problem: A guided tour of combinatorial optimization. Wiley (1985)
12. Eielst, H.A., Laporte, G.: A historical perspective on arc routing. In: Dror, M. (ed.) Arc Routing: Theory, Solutions and Applications. Springer, Berlin (2000)
13. Lin, S.: Computer solutions of travelling salesman problem. Bell System Technical Journal 44, 2245–2269 (1965)
14. Beyer, H.-G., Schwelef, H.-P.: Evolution strategies: A comprehensive introduction. Natural Computing 1(1), 3–52 (2002)
15. Homberger, J., Gehring, H.: Two evolutionary metaheuristics for the vehicle routingproblem with time windows. INFOR 37, 297–318 (1999)
16. Holland, J.H.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)
17. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: Optimization by a colony cooperation agents. IEEE Transactions on Systems, Man and Cybernetics, Part 2 26(1), 1–13 (1996)

18. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computing 1(1), 53–66 (1997)
19. Stutzle, T., Hoos, H.H.: MAX-MIN ant system. Future Generation Computer System 16(8), 889–914 (2000)
20. Gambardella, L.M., Taillard, E.D., Agazzi, G.: MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 63–76. McGraw-Hill, London (1999)
21. Berger, J., Barkaoui, M., Braysys, O.: A route-direction hybrid genetic approach for vehicle routing problem with time windows. Information System and Operational Research 41, 179–194 (2003)
22. de Castro, L.N., Timmis, J.: Artificial immune systems: A new computational intelligence approach. Springer, London (2002)
23. Keko, H., Skok, M., Skrelec, D.: Solving the distribution network routing problem with artificial immune systems. In: Proceedings of the IEEE Mediterranean Electrotechnical Conference, pp. 959–962 (2004)
24. Ma, J., Zou, H., Gao, L.-Q., Li, D.: Immune genetic algorithm for vehicle routing problem with time windows. In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, pp. 3465–3469 (2006)
25. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid Intelligent Algorithms and Applications. Information Science 180(14), 2633–2634 (2010)
26. Moscato, P., Cotta, C.: A gentle introduction to memetic algorithms. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 105–144. Kluwer, Boston (2003)
27. Mester, D., Braysys, O.: Active guided evolution strategies for large scale vehicle routing problems with time windows. Computer & Operations Research 32, 1593–1614 (2005)
28. Voudouris, C., Tsanh, E.P.K.: Guided local search. Technical Report CSM-247. Department of Computer Sciences. University of Essex, Colchester, UK (1995)
29. Mester, D., Braysy, O., Dullaert, W.: A multi-parametric evolution strategies algorithm for vehicle routing problems. Expert Systems with Applications 34, 2964–2975 (2007)
30. Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P. (eds.) Combinatorial Optimization, pp. 315–338. Wiley (1979)
31. Prakash, A., Deshmukh, S.G.: A multi-criteria customer allocation problem in supply chain environment: an artificial immune system with fuzzy logic controller based approach. Expert Systems with Applications 38(4), 3199–3208 (2011)
32. Hajiaghaei-Keshteli, M.: The allocation of customers to potential distribution centers in supply chain networks: GA and AIA approaches. Applied Soft Computing Journal 11(2), 2069–2078 (2011)
33. Wang, Y.J.: Improving particle swarm optimization performance with local search for high-dimensional function optimization. Optimization Methods and Software 25(5), 781–795 (2010)
34. Meeran, S., Morshed, M.S.: A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study. Journal of Intelligent Manufacturing, doi:10.1007/s10845-011-0520-x
35. Wang, X., Gao, X.Z., Ovaska, S.J.: A hybrid artificial immune optimization method. International Journal of Computational Intelligence Systems 2(3), 249–256 (2009)

36. Guo, D., Wang, J., Huang, J., Han, R., Song, M.: Chaotic-NSGA-II: an effective algorithm to solve multi-objective optimization problems. In: Proceedings of the International Conference on Intelligent Computing and Integrated Systems (ICISS 2010), pp. 20–23 (2010)
37. Zhang, G.: Quantum-inspired evolutionary algorithms: a survey and empirical study. Journal of Heuristics 17(3), 303–351 (2010)
38. Tao, F., Zhang, L., Zhang, Z.H., Nee, A.Y.C.: A quantum multi-agent evolutionary algorithm for selection of partners in a virtual enterprise. Manufacturing Technology 59(1), 485–488 (2010)
39. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants – Evolutionary and Neural Methods of Training. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 590–597. Springer, Heidelberg (2010)

Fuzzy C-means Clustering with Bilateral Filtering for Medical Image Segmentation

Yuchen Liu, Kai Xiao, Alei Liang, and Haibing Guan

Shanghai Key Laboratory of Scalable Computing and Systems, School of Software, Shanghai Jiao Tong University, Shanghai, China
{liuyuchen,showkey,liangalei,hbguan}@sjtu.edu.cn

Abstract. Fuzzy c-means (FCM) is a widely used unsupervised pattern recognition method for medical image segmentation. The conventional FCM algorithm and some existing variants are either sensitive to noise or prone to loss of details. This paper presents a modified FCM algorithm that incorporates bilateral filtering for medical image segmentation. The experimental results and quantitative analyses suggest that, compared to the conventional FCM, the proposed method improves clustering performance with higher standard of noise-resistance and detail-preservation.

Keywords: Fuzzy c-means, bilateral filtering, image segmentation, noise-resistance.

1 Introduction

Fuzzy c-means (FCM) clustering [1-3] is an unsupervised pattern recognition technique that has been successfully applied to clustering, classifier designs, and feature extraction in fields including medical imaging, geology, target recognition, and image segmentation. The FCM algorithm classifies images by grouping points with similar features into clusters. The algorithm iteratively minimizes a cost function which is dependent on the pixels to the cluster centers in the feature domain.

Medical image pixels are highly correlated, i.e. the pixels in the immediate neighbors possess similar feature data. In other words, the probability that adjacent pixels belong to the same cluster is great. Therefore, effectively using relationship of neighboring pixels can be of great aid in medical image segmentation [12]. However, if an image is corrupted by noise, the intensity value of some pixels may differ dramatically from that of their neighbors. Due to the fact that FCM algorithm does not take neighboring image pixels into account, neighboring pixels could be wrongly grouped into different clusters. This drawback of FCM algorithm also consequently makes it produce inhomogeneous segmentation results on noisy images. Therefore it is necessary to modify the conventional FCM algorithm for reducing noises and preserving details.

Inspired by the bilateral filtering [4-5], a new segmentation method for FCM clustering is proposed in this paper. This method applies bilateral filtering to utilize

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 221–230, 2012.

information from the neighboring pixels. Experiments of this method on magnetic resonance (MR) images show that this scheme reduces the effect of noise and makes the clustering result more homogeneous with improved image detail preservation.

2 Related Works

Fuzzy c-means (FCM) is one of the commonly-used methods for image segmentation. However, the segmentation result is usually undesired when it is applied to noisy images.

Series of contributions have been made to improve the FCM algorithm. Höppner and Klawonn [6] introduced a new way to modify objective function and to constrain the membership functions. A new FCM algorithm was proposed with improved fuzzy partitions. Zhu et al. [7] introduced a novel membership constraint function and constructed a new objective function which generalized the fuzziness index m. The methods above improved the conventional FCM algorithm by modifying objective function.

The lack of noise-resistance for conventional FCM clustering is partly caused by the fact that the conventional FCM dos not fully utilize the spatial information. A lot of effort has been made to improve the noise-resistance of FCM with utilization of the spatial information.

Pedrycz and Waletzky [8] utilized the available classification information and applied it as part of their optimization procedures. Ahmed et al. [9] modified the objective function of the conventional fuzzy c-means algorithm to compensate for inhomogeneity and to allow the labeling of a pixel to be influenced by the labels in its immediate neighborhood [10]. Chen and Zhang [11] proposed two variants of the method proposed by Ahmed et al. [9] which simplified the neighborhood term of the objective function and reduced the execution times of the algorithm. However, the methods above are only applied to single-feature inputs. Chuang et al. [12] presented a fuzzy c-means algorithm that incorporated spatial information in the membership function for clustering. The spatial function is the summation of the membership function in the neighborhood of each pixel. The modified FCM algorithm significantly improves the noise-resistance of conventional FCM. However, due to the fact that mean filter used in this work simply makes an average of neighboring pixels without taking other information into account, this method may lead to loss of details.

3 Methods

3.1 Fuzzy C-Means(FCM) Algorithm

FCM is one of most widely used methods for data analysis. This algorithm generates fuzzy partitions and prototypes for any set of numerical data [1]. In image segmentation, let $X = (x_1, x_2, x_3 \ldots x_N)$ denotes an image with N pixels to be partitioned into c clusters, where x_i represents multiple-feature data. The objective function J of FCM algorithm can be described as follows:

$$J_{FCM} = \sum_{j=1}^{N}\sum_{i=1}^{c}\mu_{ij}^{m}d_{ij}^{2} = \sum_{j=1}^{N}\sum_{i=1}^{c}\mu_{ij}^{m}\left\|x_j - a_i\right\|^2. \quad (1)$$

where $\mu_{ij} \in [0,1]$, represents the membership degree of pixel $x_j$in the *i*th cluster; a_i is the centroid of clusters *i;* m, $m > 1$, is the fuzzy index which controls the fuzziness of the resulting partition; $||*||$ is the any norm metric that measures the distance between pixel x_j to centroid a_i; and it satisfies the following constraint given by Equation (2).

$$\sum_{i=1}^{c}\mu_{ij} = 1, \forall j = 1,2,\ldots n. \quad (2)$$

The membership function represents the probability that a pixel belongs to a specific cluster. In the FCM algorithm, the probability is dependent solely on the distance between the pixel and each individual cluster centroid in the feature domain. The membership functions are cluster centroids are updated by Equation (3) and Equation (4), respectively.

$$\mu_{ij} = [\sum_{k=1}^{c}(\frac{\|x_j - a_i\|}{\|x_j - a_k\|})^{2/(m-1)}]^{-1};\ 1 \le j \le N, 1 \le i \le c. \quad (3)$$

$$a_i = \frac{\sum_{j=1}^{N}\mu_{ij}^{m}x_j}{\sum_{j=1}^{N}\mu_{ij}^{m}}; 1 \le i \le c. \quad (4)$$

FCM converges to a solution for a_i representing the local minimum or a saddle point of the objective function J_{FCM} in Equation (1). Convergence can be detected by comparing the changes in the membership function or the cluster centroid at two successive iteration steps.

The general procedure of FCM algorithm is shown by following [1]:

1. Determine value of *c*, *m* and converging error ε(in this paper, $\varepsilon = 0.00001$). Randomly choose an initial membership matrix $u^{(0)}$ with the constraint of Equation (2). Then at step *k*, $k = 1,2,3,\ldots,$LMAX.
2. Compute cluster centroids $a_i^{(k)}$, $i = 1,2,3,\ldots,c$ with Equation (4).
3. Compute an updated membership matrix $u^{(k+1)} = [\mu_{ij}^{(k+1)}]$ with Equation (3).
4. Compare $u^{(k)}$ to $u^{(k+1)}$ or $a^{(k)}$ to a^{k+1} or $J^{(k)}$ to $J^{(k+1)}$ in any convenient norm. Computation stops if the difference is less than ε. Otherwise, set $u^{(k)} = u^{(k+1)}$ and return to step 2.

3.2 FCM with Bilateral Filtering

Bilateral filtering proves to be a good method for noise reduction and edge preservation. To incorporate bilateral filtering into FCM algorithm, a filter function is defined as follows:

$$h_{ij} = \sum_{k\epsilon N(x_j)} f(x_k, x_j)g(x_k, x_j)u_{ik} \quad (5)$$

where $N(x_j)$ represents a square window centered on pixel x_j in the spatial domain, $f(x_k, x_j)$ represents domain filter and $g(x_k, x_j)$ represents range filter. The two filters follow the Gaussian distribution. Their variance is denoted as σ_d and σ_r respectively. u_{ik} represents the conventional membership function of point x_k to cluster *i*.

The filter function is incorporated into the membership function in the conventional FCM method as follows:

$$u_{ij}{}' = \frac{u_{ij}{}^p h_{ij}{}^q}{\sum_{k=1}^{c} u_{kj}{}^p h_{kj}{}^q} \tag{6}$$

where *p* and *q* are parameters to control the relative importance of both functions [12]. h_{ij} is transformed into the weight of corresponding u_{ij} to calculate the new membership function $u_{ij}{}'$. If the center pixel is in a homogenous region, the filter function becomes a Gaussian filter, and the clustering result will be the same as the conventional FCM. For a noise point, its filter function is small because its membership function value for a cluster is quite different from its neighbors and the range filter will generate a smaller result. As a result, noisy pixels are more likely to be classified correctly. In the following sections, the FCM with bilateral filtering with parameter *p* and *q* is denoted as $\text{bFCM}_{p,q}$. Note that $\text{bFCM}_{1,0}$ is identical to the conventional FCM.

The membership function is calculated through the conventional FCM algorithm first and the bilateral filter function is then applied to membership function matrix in each iteration. The algorithm ends when the difference of objective function between the two successive iterations is less than a threshold (In this study, the threshold = 0.01). After that, each pixel is grouped into a specific cluster for which the membership function is maximal.

4 Experiments

4.1 Image Data

In this study, magnetic resonance (MR) images are applied to demonstrate the effect of bFCM. The image was corrupted by Gaussian noise with signal-to-noise ratio (SNR) = 10 and 5. The image was grouped into three clusters in experiments and a 5×5 window for filters is chosen throughout the whole work.

4.2 Cluster Validity Functions and Methods for Discriminative Analysis

Three kinds of cluster validity functions were used in this study to evaluate the performance of clustering. They are partition coefficient V_{pc} [13], partition entropy V_{pe} [14] and V_{xb} defined by Xie and Beni [15]. They are defined as follows:

$$V_{pc} = \frac{\sum_j^N \sum_i^c u_{ij}{}^2}{N} \tag{7}$$

$$V_{pe} = \frac{-\sum_j^N \sum_i^c [u_{ij} \log u_{ij}]}{N} \tag{8}$$

V_{pc} and V_{pe} measure fuzziness of the partition. Better partition should have less fuzziness. Therefore greater V_{pc} or smaller V_{pe} stands for better clustering performance [13] [14].

V_{xb} measures feature structure which remedies the lack of connection to the featuring property if V_{pc} or V_{pe} is used. A better clustering is expected to have smaller value of V_{xb}. [15]

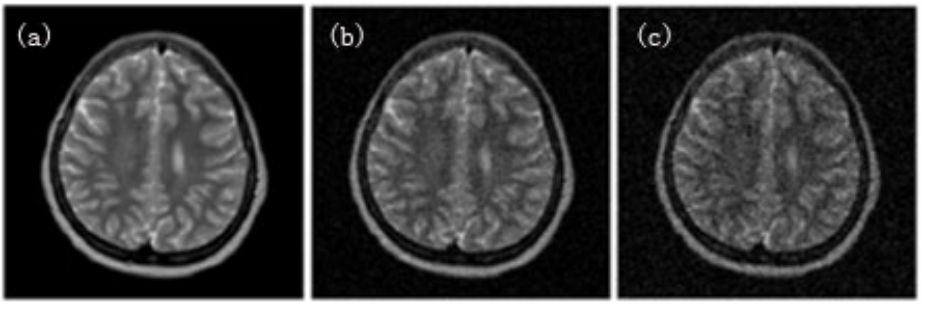


Fig. 1. (a) MR image used for study. (b) Noisy image added with noise SNR = 10. (c) Noisy image added with noise SNR = 5

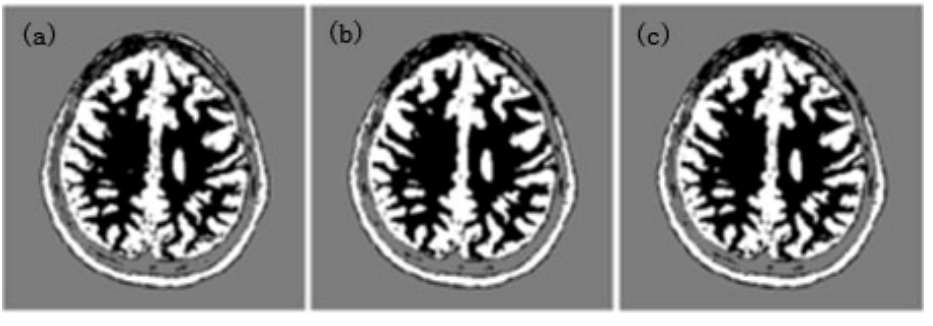


Fig. 2. Segmentation results of MR image using (a) FCM; (b) $bFCM_{0,2}$ (c) $bFCM_{1,1}$

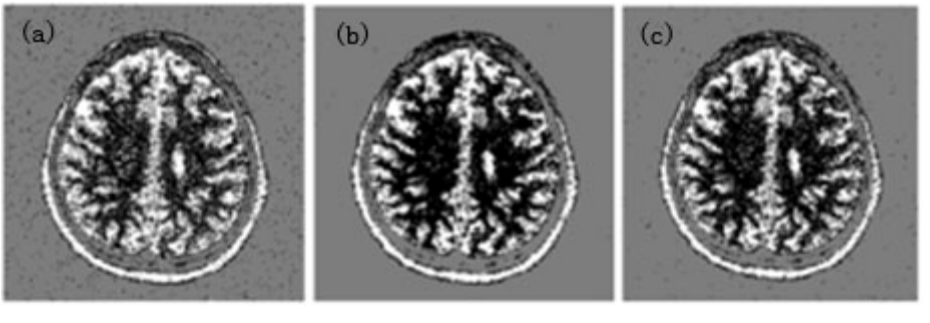


Fig. 3. Segmentation results of noisy MR image (SNR=10) using (a) FCM; (b) $bFCM_{0,2}$; (c) $bFCM_{1,1}$

The conventional FCM and an existing improved FCM algorithm [12] are used in this work for the performance comparison as discriminative analysis. Named as sFCM, with the use of a mean filter, this algorithm incorporates spatial information into the membership function thereby utilizes information from the neighbors of each pixel [12].

5 Results and Discussion

Fig. 1(a) shows the MR image of brain tissues. **Fig. 2**(a) shows the segmentation result generated by the conventional FCM algorithm and **Fig. 2**(b) and (c) show the results of the FCM with bilateral filtering with parameters(p=0, q=2) and (p=1, q=1), respectively. In **Fig. 2**(b) and (c), (σ_d, σ_r) values (4, 0.4). The conventional FCM classifies the MR image into three clusters with relatively high homogeneity. However, some spurious points still appear inside the black area. Compared to the conventional FCM algorithm, the segmentation results of FCM with bilateral filtering is almost the same, while spurious points are reduced in black area and the segmented images are more homogeneous. It is observed that **Fig. 2**(b) looks more homogeneous than **Fig. 2**(c) because it has a higher q parameter. Better smoothing effect is achieved when parameter q is higher.

Table 1. The clustering results of noisy images using various FCM algorithms

Image	Algorithm	V_{pc}	V_{pe}	V_{xb}
Noise added MR image SNR = 10	FCM	0.8498	0.2787	0.0237
	$sFCM_{0,2}$	0.7869	0.3544	0.0743
	$sFCM_{1,1}$	0.9122	0.1562	0.0261
	$bFCM_{0,2}$	0.9279	0.1333	0.0291
	$bFCM_{1,1}$	0.9407	0.1056	0.0263
Noise added MR image SNR = 5	FCM	0.8318	0.3072	0.0264
	$sFCM_{0,2}$	0.7135	0.4905	0.1335
	$sFCM_{1,1}$	0.8796	0.2134	0.0272
	$bFCM_{0,2}$	0.8978	0.1878	0.0322
	$bFCM_{1,1}$	0.9220	0.1377	0.0280

Fig. 1(b) and (c) show the MR images added with Gaussian noise, whose SNR values are 10 and 5, respectively. **Fig. 3**(a) shows the segmentation result of **Fig. 1**(b) (SNR=10) generated by the conventional FCM algorithm. As can be seen, the conventional FCM algorithm misclassifies numerous pixels because the added noises change the pixel values of these points. Compared with **Fig. 2**, the segmentation result of the conventional FCM algorithm has many spurious points caused by noise. Especially in the brain center, which is black in segmented image, there are many white and gray points. **Fig. 3**(b) and (c) show the results of the FCM with bilateral filtering with parameters (p=0, q=2) and (p=1, q=1), respectively. In **Fig. 3**(b) and (c), (σ_d, σ_r) values (4, 0.4). It can be easily observed that, the segmentation results created by

bFCM are more homogeneous than that of the conventional FCM algorithm as the majority of the spurious points are obliterated. The bilateral filter that incorporated with membership function weakens the effect of noise because the weight h_{ij} calculated in its neighboring pixels is small. Furthermore, the pixels that are correctly clustered are enhanced because the weight calculated in its neighbors is large. Consequently, FCM with bilateral filtering significantly corrects the misclassification caused by noise and the result is similar to that of original image.

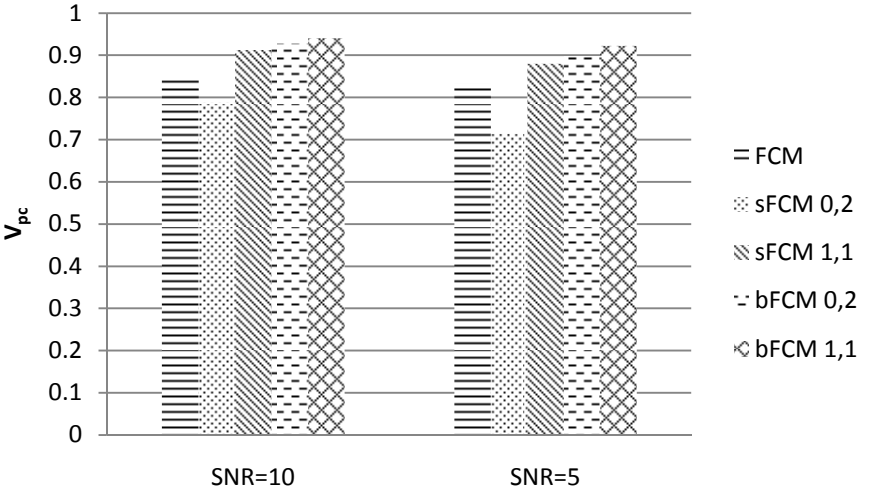


Fig. 4. Vpc of FCM, sFCM and bFCM with (σ_d, σ_r) = (4, 0.4)

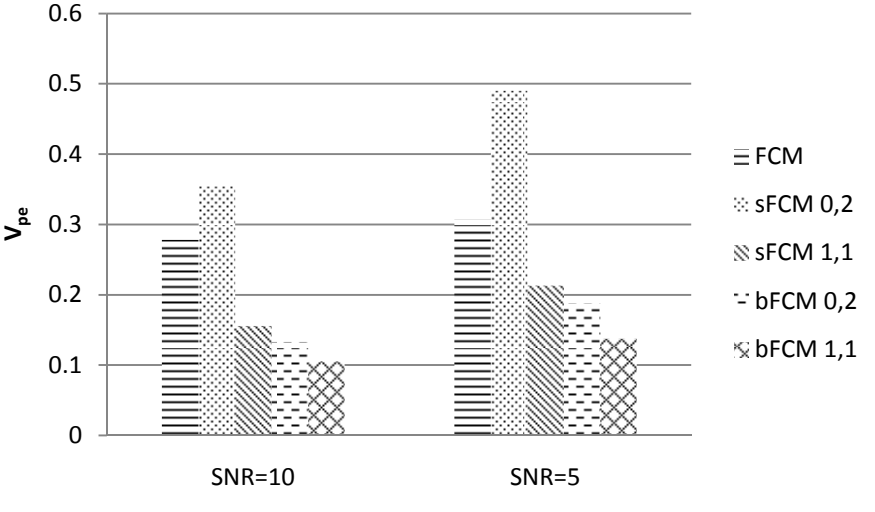


Fig. 5. Vpe of FCM, sFCM and bFCM with (σ_d, σ_r) = (4, 0.4)

Table 1 lists validity functions of the conventional FCM, sFCM [12] and bFCM when they are applied on noisy images. **Fig. 4** and **Fig. 5** illustrate V_{pc} and V_{pe} with bar chart respectively. Considering segmented results as well as validity functions, (σ_d, σ_r) for bFCM and sFCM [12] is defined as (4, 0.4). Compared with the conventional FCM algorithm, V_{pc} and V_{pe} of bFCM are much better. Especially for $bFCM_{1,1}$, the increase in V_{pc} is around 0.1 and the reduction in V_{pe} is more than 50% when SNR equals 10. Compared with sFCM [12], the improvement with parameter (p=0, q=2) is evident. With parameter (p=1, q=1), the increase in V_{pc} is around 0.03 and the reduction in V_{pe} is about 33% when SNR equals 10.

In contrast to the improvement of V_{pc} and V_{pe}, V_{xb} which measures feature structure hardly changes with the use of bFCM and even increases a little. It can be explained by the fact that the FCM with bilateral filtering modifies the membership function on the basis of neighboring information, which reduces the compactness of feature domain.

Table 2. The clustering results of noisy image with SNR = 10 using bFCM

(σ_d, σ_r)	V_{pc}		V_{pe}		V_{xb}	
	$bFCM_{0,2}$	$bFCM_{1,1}$	$bFCM_{0,2}$	$bFCM_{1,1}$	$bFCM_{0,2}$	$bFCM_{1,1}$
(3, 0.2)	0.9552	0.9499	0.0838	0.0892	0.0275	0.0266
(3, 0.3)	0.9457	0.9464	0.1028	0.0958	0.0278	0.0264
(3, 0.4)	0.9281	0.9406	0.1330	0.1056	0.0290	0.0263
(4, 0.2)	0.9554	0.9501	0.0835	0.0889	0.0276	0.0267
(4, 0.3)	0.9457	0.9465	0.1029	0.0957	0.0278	0.0265
(4, 0.4)	0.9279	0.9407	0.1333	0.1056	0.0291	0.0263
(5, 0.2)	0.9555	0.9502	0.0834	0.0888	0.0276	0.0267
(5, 0.3)	0.9457	0.9465	0.1029	0.0956	0.0279	0.0265
(5, 0.4)	0.9278	0.9407	0.1335	0.1056	0.0291	0.0263

Table 3. The clustering results of noisy image with SNR = 5 using bFCM

(σ_d, σ_r)	V_{pc}		V_{pe}		V_{xb}	
	$bFCM_{0,2}$	$bFCM_{1,1}$	$bFCM_{0,2}$	$bFCM_{1,1}$	$bFCM_{0,2}$	$bFCM_{1,1}$
(3, 0.2)	0.9457	0.9390	0.1016	0.1078	0.0304	0.0295
(3, 0.3)	0.9293	0.9325	0.1339	0.1197	0.0297	0.0288
(3, 0.4)	0.8981	0.9221	0.1872	0.1376	0.0319	0.0280
(4, 0.2)	0.9459	0.9392	0.1014	0.1075	0.0304	0.0295
(4, 0.3)	0.9292	0.9325	0.1343	0.1197	0.0297	0.0288
(4, 0.4)	0.8978	0.9220	0.1878	0.1377	0.0322	0.0280
(5, 0.2)	0.9460	0.9393	0.1013	0.1074	0.0305	0.0295
(5, 0.3)	0.9291	0.9326	0.1344	0.1197	0.0297	0.0292
(5, 0.4)	0.8977	0.9220	0.1881	0.1378	0.0326	0.0280

Table 2 and **Table 3** tabulate the validity functions to evaluate the performance of clustering for two noisy images. It can be seen from the results that σ_d has little effect

on values of validity functions. With the increase of σ_r which leads to better smoothing effects on segmented results, the validity functions change oppositely. Validity functions of bFCM with parameter (p=0, q=2) change more swiftly than that with parameter (p=1, q=1).

6 Conclusion

The proposed bFCM for medical image segmentation is designed to reduce image noise as well preserve details. By incorporating bilateral filtering into the conventional FCM algorithm, membership functions of the neighbors of the center pixel are processed by the bilateral filter to calculate the new membership function. The effect from noisy pixels is weakened by this bilateral filter and the segmentation result becomes more homogeneous.

This new method was tested on MR images. An existing modified FCM called sFCM [12] and various validity functions are applied for conducting discriminative and analytical experiments to evaluate the performance of the new method. The experimental results show the new method has superior image noise-resistance and detail-preservation over both the conventional FCM and sFCM [12].

Acknowledgements. This work was partly sponsored by The National Natural Science Foundation of China (Grant No.60970107, 60970108), The Science and Technology Commission of Shanghai Municipality (09510701600).

References

1. James, C.B., Robert, E., William, F.: FCM: the Fuzzy C-means Clustering Algorithm. Computers & Geosciences 10(2-3), 191–203 (1984)
2. Clark, M.C., Hall, L.O., Goldgof, D.B., Clarke, L.P., Velthuizen, R.P., Silbiger, M.S.: MRI Segmentation Using Fuzzy Clustering Techniques. IEEE Eng. Med. Biol. 13, 730–742 (1994)
3. Pham, D.L., Prince, J.L.: Adaptive Fuzzy Segmentation of Magnetic Resonance Image. IEEE Trans. Med. Imaging 18, 737–752 (1999)
4. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of International Conference on Computer Vision, pp. 839–846 (1998)
5. Vijaya, G., Vasudevan, V.: Bilateral Filtering Using Modified Fuzzy Clustering for Image Denoising. International Journal on Computer Science and Engineering 3, 45–49 (2010)
6. Höppner, F., Klawonn, F.: Improved Fuzzy Partitions for Fuzzy Regression Model. International Journal of Approximate Reasoning 32, 85–102 (2003)
7. Zhu, L., Chung, F.L., Wang, S.: Generalized Fuzzy C-means Clustering Algorithm with Improved Fuzzy Partitions. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39(3), 578–591 (2009)
8. Pedrycz, W., Waletzky, J.: Fuzzy Clustering with Partial Supervision. IEEE Trans. Syst. Man Cybern. Part B Cybern. 27, 787–795 (1997)

9. Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A., Moriaty, T.: A Modified Fuzzy C-means Algorithm for Bias Field Estimation and Segmentation of MRI Data. IEEE Trans. Med. Imaging 21, 193–199 (2002)
10. Ahmed, M.N., Yamany, S.M., Farag, A.A., Moriarty, T.: Bias Field Estimation and Adaptive Segmentation of MRI Data Using Modified Fuzzy C-means algorithm. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 250–255 (1999)
11. Chen, S.C., Zhang, D.Q.: Robust Image Segmentation Using FCM with Spatial Constraints Based on New Kernel-Induced Distance Measure. IEEE Transactions on Systems, Man, and Cybernetics, Part B 34(4), 1907–1916 (2004)
12. Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J., Chen, T.J.: Fuzzy C-means Clustering with Spatial Information for Image Segmentation. Computerized Medical Imaging and Graphics 30(1), 9–15 (2006)
13. Bezdek, J.C.: Cluster Validity with Fuzzy Sets. J. Cybern. 3, 58–73 (1974)
14. Bezdek, J.C.: Mathematical Models for Systematic and Taxonomy. In: Proceedings of Eighth International Conference on Numerical Taxonomy, vol. 3, pp. 143–166 (1975)
15. Xie, X.L., Beni, G.: A Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Anal. Mach. Intell. 3, 841–846 (1991)

A Improved Clustering Analysis Method Based on Fuzzy C-Means Algorithm by Adding PSO Algorithm

Liang Pang, Kai Xiao, Alei Liang, and Haibing Guan*

School of Software, Shanghai Key Laboratory of
Scalable Computing and Systems, Shanghai Jiao Tong University, China
cyclone0000@163.com, showkey@gmail.com,
{liangalei,hbguan}@sjtu.edu.cn

Abstract. Fuzzy c-means algorithm (FCM) is one of the most widely used clustering methods for modern medical image segmentation applications. However the conventional FCM algorithm has certain possibilities of converging to a local minimum of the objective function, thus lead to undesired segmentation results. To address this issue, an improved FCM which is based on clustering centroids updates with the use of particle swarm optimization (PSO) is proposed in this paper. This algorithm is designed to support multidimensional feature data and be accessible through parallel computation. The experimental results suggest that, compared to the conventional FCM algorithm, the proposed algorithm leads to higher chances of global optimum clustering and is less computationally intensive when large clustering number is needed.

Keywords: Fuzzy c-means, Particle swarm optimization algorithm, Image segmentation, Clustering, Swarm intelligence.

1 Introduction

Magnetic resonance imaging (MRI), a typical multi-spectral imaging technique, is a preferred imaging modality for examining brain shape, volume and tissue distribution altered by neurological conditions [7]. Image segmentation on the brain magnetic resonance (MR) images is an important process for measuring these physiological and anatomical alterations [7].

Fuzzy c-means (FCM) clustering [1,5,6] is a typical unsupervised pattern recognition method that has been successfully applied to feature analysis, clustering, and classifier designs in the fields such as astronomy, geology, medical imaging, target recognition, and image segmentation. This method was initially developed and improved by Dunn [8] and Bezdek [9], respectively. Being frequently used in pattern recognition, FCM algorithm separates the input multidimensional dataset into clusters by grouping the similar data points. The

* Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 231–242, 2012.

clustering is achieved by iteratively minimizing an objective function which is based on measuring the distance of each data point to the cluster centroids in the feature domain.

Several investigators have developed segmentation schemes by the use of FCM method [1-4]. However, some of these methods do not utilize the multi-spectral information in the MR images [7]. Furthermore, due to the fact that its initial clustering centroids are not globally optimized and each iteration in the computation does not communicate with the previous ones, FCM always converges to a local optimum. Furthermore, its clustering is greatly impacted by the initial centroids or membership function. In the recent years, some optimum approaches have been applied to improve FCM algorithm in this aspect. Most of them are focused on improving the membership function and objective function, such as the use of a genetically guided approach in the objective function [13] [14], the application of Mahalanobis distances [15] [16], the optimization of membership function by particle swarm optimization (PSO) algorithm [16-19] and the inclusion of the spatial information [7].

With the objective of improving the performance of FCM algorithm for brain MR image segmentation, this paper introduces a new clustering scheme by the incorporation of PSO into FCM. By treating FCM algorithm as a process of looking for the minimum value in the potential solutions of a nonlinear function, this study takes advantage of PSO to reach an agreement of the candidate solutions, thereby obtaining optimal centroid value of each cluster. Compared to the conventional FCM, this scheme greatly reduces the possibility of falling into local optimum and shows higher performance when the number of clusters is large.

2 Method

2.1 Fuzzy C-Means(FCM) Algorithm

The FCM algorithm is one of most widely used methods for data analysis. This program generates fuzzy partitions and prototypes for any set of numerical data [20]. In image segmentation, let $X = (x_1, x_2, x_3, ..., x_N)$ denotes an image with N pixels to be partitioned into c clusters, where x_i represents multidimensional data. The objective function of FCM algorithm can be described as follows:

$$\begin{cases} J_{FCM} = \sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^{m} d_{ij}^{2} = \sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^{m} \parallel x_j - a_i \parallel^2 , \\ \sum_{i=1}^{c} \mu_{ij} = 1, \forall j = 1, 2, 3, ..., N . \end{cases} \tag{1}$$

where $\mu_{ij} \in [0, 1]$ represents the membership degree of pixel x_j in the ith cluster; a_i is the centroid of cluster i; $m > 1$ is the fuzzy index which controls the fuzziness of the resulting partition; $d_{ij}^2 = ||x_j - a_i||^2$ is the Euclidean distance between pixel x_j to centroid a_i;

The membership functions and cluster centroids update as shown in Equation (2) and Equation (3):

$$\mu_{ij} = [\sum_{k=1}^{c} (\frac{||x_j - a_i||}{||x_j - a_k||})^{\frac{2}{m-1}}]^{-1}; 1 \le j \le N, 1 \le i \le c\,. \tag{2}$$

$$a_i = \frac{\sum_{j=1}^{N} \mu_{ij}^m x_j}{\sum_{j=1}^{N} \mu_{ij}^m}; 1 \le i \le c\,. \tag{3}$$

The general procedure of FCM algorithm is shown by following [20]:

1. Determine value of c, m and converging error ε(in this paper, $\varepsilon = 0.00001$). Randomly choose an initial membership matrix $u^{(0)}$ with the constraint of Equation (1). Then at step $k, k = 1, 2, 3, ..., LMAX$.
2. Compute cluster centroids $a_i^{(k)}, i = 1, 2, 3, ..., c$ with Equation (3).
3. Compute an updated membership matrix $u^{(k+1)} = [\mu_{ij}^{(k+1)}]_{c \times N}$ with Equation (2).
4. Compare $u^{(k)}$ to $u^{(k+1)}$ or $a^{(k)}$ to $a^{(k+1)}$ or $J^{(k)}$ to $J^{(k+1)}$ in any convenient norm. If the changes are less than ε, stop. Otherwise, set $u^{(k)} = u^{(k+1)}$ and return to 2.

2.2 Particle Swarm Optimization (PSO) Algorithm

There are two insights of a process of optimization of swarm particles [11]: (1) being assumed as particles in space, the points will tend to move toward and influence one another, with the objective of seeking agreement with their neighbors; (2) the space in which the particles move is heterogeneous with respect to evaluation: some regions are better than others. Some points in the parameter space result in greater fitness than others. With this architecture, the process of finding better regions can be viewed as solving the problems of optimization.

As a typical method for the implementation of swarm intelligence (SI), the PSO algorithm [10] [11] was inspired by social behavior of bird flocking or fish schooling. PSO is a population-based stochastic optimization (known as metaheuristics) technique and also a popular and robust strategy for some optimization problems [12]. PSO algorithm suggests that individuals moving through a sociocognitive space should be influenced by their own previous behavior and by the successes of their neighbors [21]. As the system is dynamic, each individual is presumed to be moving at all times: this is the Lewinian concept of locomotion. The direction of movement is a function of the current position and velocity, the location of the previous best success of individual, and the best position found by any member of the neighborhood [21]. It shows as Equation (4).

$$\begin{cases} \boldsymbol{v}_i(t) = \boldsymbol{v}_i(t-1) + \varphi_1(\boldsymbol{p}_l - \boldsymbol{x}_i(t-1)) + \varphi_2(\boldsymbol{p}_g - \boldsymbol{x}_i(t-1)), \\ \boldsymbol{x}_i(t) = \boldsymbol{x}_i(t-1) + \boldsymbol{v}_i(t). \end{cases} \tag{4}$$

In Equation (4), $\boldsymbol{x}_i$ is the algebraic vector symbol of the position of a particle i; $\boldsymbol{v}_i$ is called for velocity, it is a vector that are added to the position coordinates

in order to move the particle through iteration; φ_1 and φ_2 are the effective parameters; $\boldsymbol{p}_l$ is the best solution of particle i in advance; $\boldsymbol{p}_g$ is the best solution of all particles in advance.

2.3 PSO-FCM Algorithm

Taking the membership functions into the objective function, Equation (1) can be changed as follows:

$$J_{FCM} = J_{FCM}(\boldsymbol{a}, X) = \sum_{j=1}^{N}\sum_{i=1}^{c}[\sum_{k=1}^{c}(\frac{||x_j - a_i||}{||x_j - a_k||})^{\frac{2}{m-1}}]^{-m}||x_j - a_j||^2 \,. \tag{5}$$

Minimizing the objective function with the constraint as Equation (1) is a non-trivial constraint nonlinear optimization task with continuous parameters a_i and μ_{ij}, and there is no closed-form solution. However, PSO algorithm is qualified to solve this problem as its swarm particles scan all potential solutions. With the use of PSO algorithm, the task of minimizing the objective function becomes a process of searching the minimum value of J_{FCM} at the optimal cluster centroid $\boldsymbol{a} = (a_1, a_2, ..., a_c)$ in Equation (5).

In order to use PSO algorithm to obtain the minimum value of J_{FCM}, the self-optimal centroid $\boldsymbol{p}_l$ and the global optimal centroid $\boldsymbol{p}_g$ are required to be determined. If n particles are used to search the best solution of cluster centroid $\boldsymbol{a}$, every particle denotes a potential solution to $\boldsymbol{a}$, so the particle h is represented as $\boldsymbol{a}_h = (a_{h1}, a_{h2}, ..., a_{hc})$ and J_h represents the objective function value at $\boldsymbol{a} = \boldsymbol{a}_h$. Then, the self-optimal centroid of particle h named $\boldsymbol{p}_{hl}$ can be defined easily as the value of $\boldsymbol{a}_h$ when the minimum J_h in the previous iterations is obtained. However, with Equation (5) and Equation (1), the global optimal centroid $\boldsymbol{p}_g$ is difficult to be defined correctly. This challenge is caused by that, in these two equations, every particle has its own membership matrix $u_h, 1 \le h \le n$ by which J_h is calculated. Let $\boldsymbol{p}_g$ equal to $\boldsymbol{a}_h$ for $min_{1\le h\le n}(J_h)$, it will result in a large number of iterations or even lead to divergence. Hence, a unified membership matrix u_g produced by the global optimal centroid $\boldsymbol{p}_g = (p_{g1}, p_{g2}, ..., p_{gc})$ as Equation (6) is used. Equation (5) can therefore be improved to Equation (7) for calculating every particle value of objective function.

$$\begin{cases} u_g = [\mu_{ij}]_{c\times N}, \\ \mu_{ij} = [\sum_{k=1}^{c}(\frac{||x_j - p_{gi}||}{||x_j - p_{gk}||})^{\frac{2}{m-1}}]^{-1}; 1 \le j \le N, 1 \le i \le c. \end{cases} \tag{6}$$

$$J_h = J(\boldsymbol{a}_h, X) = \sum_{j=1}^{N}\sum_{i=1}^{c}\mu_{ij}^m||x_j - a_{hi}||^2; 1 \le h \le n \,. \tag{7}$$

To determine the global optimal centroid $\boldsymbol{p}_g$, Equation (7) is divided into c components like Equation (8) where j_{hi} represents the ith cluster component of

objective function of particle h, thus results in c n-dimensional vectors $\boldsymbol{j}_1$ to $\boldsymbol{j}_c$ where $\boldsymbol{j}_i = (j_{1i}, j_{2i}, ..., j_{ni})$.

$$\begin{cases} J_h = j_{h1} + j_{h2} + j_{h3} + ... + j_{hc}; 1 \le h \le n \ , \\ j_{hi} = \sum_{j=1}^{N} \mu_{ij}^m ||x_j - a_{hi}||^2; 1 \le h \le n, 1 \le i \le c \ . \end{cases} \tag{8}$$

To minimize the value of objective function, every cluster component of objective function must be minimized. So every element of the global optimal centroid $\boldsymbol{p}_g$ can be defined as Equation (9).

$$p_{gi} = a_{hi}, where\ h\ produced\ by\ j_{hi} = min(\boldsymbol{j}_i); 1 \le i \le c \ . \tag{9}$$

PSO algorithm can be consequently taken into looking for the cluster centroids by the following Equation (10).

$$\begin{cases} \boldsymbol{v}_h(t) = \boldsymbol{v}_h(t-1) + \varphi_1(\boldsymbol{p}_{hl} - \boldsymbol{a}_h(t-1)) + \varphi_2(\boldsymbol{p}_g - \boldsymbol{a}_h(t-1)), \\ \boldsymbol{a}_h(t) = \boldsymbol{a}_h(t-1) + \boldsymbol{v}_h(t). \end{cases} \tag{10}$$

When using PSO algorithm to find the extremum of a function, after several iterations it will fall in a stable state and almost all particles are equal to the global optimal centroid $\boldsymbol{p}_g$. Then the criteria for stopping iteration of algorithm can be defined as same as FCM algorithm, i.e., compare $(u_g^{(k+1)} - u_g^{(k)})$ or $max(\boldsymbol{a}_h^{(k+1)} - \boldsymbol{a}_h^{(k)})$ or $max(J_h^{(k+1)} - J_h^{(k)})$ or $(\boldsymbol{p}_g^{(k+1)} - \boldsymbol{p}_g^{(k)})$ to ε in any norm. Stop iteration if it is less than ε.

3 Experiments and Analytical Discussions

3.1 Data

In this paper, all test data are brain MR images of Tesla 1.5 with 256×256 pixels [26], one T2-AXIAL MR image is used as 1-dimensional feature data, one T1-CM MR image and one T2-AXIAL MR image from same patient are combined as 2-dimensional feature data. Both FCM and PSO-FCM algorithm use fuzzy index $m = 2$ and the PSO effective parameters $\varphi_1 = 0.1 \times rand(1)$ and $\varphi_2 = 0.3 \times rand(1)$, where $rand(1)$ represents a random number from 0 to 1 and it will make the trajectory of every particle as the sine wave[21].

3.2 Clustering Validity Functions

Two types of clustering validity functions, fuzzy partition and feature structure, are often used to evaluate the performance of clustering in different clustering methods. The validity functions based on fuzzy partition are partition coefficient V_{pc}[22] and partition entropy V_{pe}[23]. They are defined as follows:

$$V_{pc} = \frac{\sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^2}{N} \tag{11}$$

$$V_{pe} = \frac{-\sum_{j=1}^{N}\sum_{i=1}^{c}\mu_{ij}\log\mu_{ij}}{N} \tag{12}$$

The best clustering is achieved when the value V_{pc} is maximal or V_{pe} is minimal, it expresses that less fuzziness means take better performance.

Another kind of validity functions is based on the feature structure [24] [25], it considers not only the fuzzy partition, but also make a direct connection to the featuring property. V_{xb} is widely used one of this kind of validity functions and defined as Equation (13).

$$V_{xb} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{c}\mu_{ij}^{m}||x_j - a_i||^2}{N \times (min_{1\leq k,m\leq c,m\neq k}||a_k - a_m||^2)} \tag{13}$$

A good clustering result generates samples that are compacted within one cluster and samples that are separated between different clusters. Minimizing V_{xb} is expected to lead to a good clustering [24].

3.3 Validity Experiments and Analytical Discussions

The correctness of PSO-FCM algorithm is validated by the comparison with that of the conventional FCM. Both 1-dimensional and 2-dimensional feature data are chosen in this experiment. Twenty images per kind of data were divided into four teams. From team one to team four, images are grouped into three to six clusters by both FCM and PSO-FCM. Three types of average errors are obtained according to Equation (14).

$$\begin{cases} e_u = \sum_{i=1}^{c}\sum_{j=1}^{N}|\mu_{FCM_{ij}} - \mu_{PSO-FCM_{ij}}|, \\ e_a = \frac{1}{c\times d}\sum_{i=1}^{c}\sum_{k=1}^{d}|a_{FCM_{ik}} - a_{PSO-FCM_{ik}}|, \\ e_J = |J_{FCM} - J_{PSO-FCM}| \end{cases} \tag{14}$$

where c is the number of clusters to be partitioned; d is the feature dimension of input data; $a_{FCM_{ik}}$ and $a_{PSO-FCM_{ik}}$ are the centroid of cluster i in kth dimensional data calculated by FCM and PSO-FCM; $\mu_{FCM_{ij}}$ and $\mu_{PSO-FCM_{ij}}$ represent the membership degrees of pixel x_j in cluster i obtained by FCM and PSO-FCM; J_{FCM} and $J_{PSO-FCM}$ represent the values of objective function obtained by FCM and PSO-FCM, respectively. Table 1 shows the experimental results. In Table 1, the means of average errors with different clustering numbers in each team are consistently close to zero. These values indicate that the results of membership functions, cluster centroids and objective functions from both FCM and PSO-FCM algorithms are nearly identical. This shows that FCM and PSO-FCM algorithm almost produce the same experimental results, thus proves the correctness of PSO-FCM algorithm.

3.4 Speed Testing Experiments

The number of iterations which is closely related to the running time of PSO-FCM algorithm is used to test the speed of a clustering process. Three testing

Table 1. Means of average errors between FCM and PSO-FCM in each team of different dimensional feature data

Image feature	Team number	Clustering number	$e_u(10^{-12})$	$e_a(10^{-4})$	e_J
1-dimensional	1	3	4.41	10	0.582
	2	4	5.18	6	0.698
	3	5	0.892	17	0.0264
	4	6	9.88	5.5	0.3249
2-dimensional	1	3	0.277	1.37	0.0075
	2	4	3.13	7.82	0.1042
	3	5	1.58	2.4	0.1856
	4	6	4.33	1.05	0.0945

experiments are included to determine the regular pattern of iterations and the relationships between iterations, particle size and clustering number.

The Regular Pattern of Iterations. In this experiment, we select two sets of images of both 1-dimensional and 2-dimensional feature data as before. Each set has three testing images. All images are applied in PSO-FCM algorithm with three particles and each image is divided into three, four, five clusters. Each trial is repeated ten times. Max/Min column shown in Table 2 represents the maximum/minimum iteration number among the ten trials, while Mean column represents the average value.

Table 2. Mean number of iterations required to partition different clusters on each of the three images for the two sets, and the minimum and maximum number of iterations required

Image number	Clustering number	1-dimensional feature data			2-dimensional feature data		
		Max	Min	Mean	Max	Min	Mean
1	3	754	566	649.3	793	524	636.6
	4	764	581	641.1	800	503	678.5
	5	802	553	648.5	780	546	657.8
2	3	740	524	634.8	772	502	647.7
	4	794	503	646.1	797	509	628.7
	5	783	509	632.1	777	523	631.7
3	3	786	545	670.7	767	515	631.3
	4	762	561	676	782	506	646.4
	5	779	522	660.7	751	519	636.2

From Table 2, it can be seen that the number of iterations is between 500 and 800 and the means are basically fixed in the small range between 630 and 670 with three particles regardless clustering number and data dimension. Furthermore, we can confirm that the number of iterations of PSO-FCM remains stably in a fixed range with the same particle size.

The Relationship between Iterations and Particle Size. We select one 1-dimensional image data and divide it into four clusters with three, five, ten and fifteen particles. Ten trials are repeated for each above case. Table 3 tabulates the results.

Table 3. Mean number of iterations required to partition same clusters with different particle size on the same image, and the minimum and maximum number of iterations required

Particle size	Max	Min	Mean
3	764	581	641.1
5	646	505	572
10	615	416	497.7
15	586	435	487.3

One advantage of PSO algorithm is that the increased particle size improves implementation efficiency which alternatively means the number of iterations decreases. From Table 3, it can be clearly seen that the mean value of iterations decreases while the particle size increases. So it can be inferred that larger particle size leads to less number of iterations.

The Relationship between Iterations and Clustering Number. When using PSO-FCM algorithm in image segmentation, the particles work as individuals. So the computational overhead of all particles can be worked in parallel. In PSO-FCM algorithm, assuming that the swarm only has one particle, the time consumption is as the same as the FCM algorithm. So if using parallel computing approach to implement PSO-FCM algorithm, the overhead of several particles is equal to that of only one particle. In summary, time consumption of both FCM and PSO-FCM can be viewed as iteration number.

In this experiment, only one image is selected and divided into three, four, five, six, seven, eight, nine, ten, twelve and fifteen clusters for FCM and PSO-FCM algorithm respectively. The particle size is fixed at three. This experiment makes a comparison with the conventional FCM algorithm when the clustering number increases. Fig. 1 shows the results of the experiment.

It can be seen from the previous experiments that the PSO-FCM algorithm usually need a large number of fitness evaluations before a satisfying result can be obtained, which can also be viewed as the main difficulty in applying PSO to real-world applications. However, its number of iterations is only affected by the particle size, but not the clustering number. Therefore the increased clustering number improves the efficiency of PSO-FCM. It can be learnt From Fig. 1 that, the number of iterations is linear with the clustering number in the partition by using FCM algorithm. But the number of iterations remains stably in a fixed range in the case of PSO-FCM algorithm. Compared to the conventional FCM,

Fig. 1. The relationship between iterations and desired number of clusters when using FCM and PSO-FCM algorithm

PSO-FCM has worked more efficiently when large clustering number of twelve and fifteen is used, and this maintains in a foreseeable view.

3.5 Performance Experiments

Ten 2-dimensional images are randomly selected and divided into four clusters used FCM and PSO-FCM algorithm, respectively. The particle size of PSO-FCM is fixed at three. The value of the objective function and the cluster validity functions computed through various algorithms are shown in Table 4. In most cases, the objective function and validity functions based on the FCM and PSO-FCM are nearly the same. However it can be observed that, PSO-FCM creates better clustering results than the conventional FCM in 3 out the 10 cases. For example, in the first case, PSO-FCM reduced 28% in the value of objective function, 34% in V_{pe}, 94% in V_{xb} and improved 5% in V_{pc}, compared to those of the conventional FCM.

The robustness of an algorithm can be expressed as whether the algorithm gets the correct results in multiple times. Table 4 tabulates the experimental results used to evaluate the performance of FCM and PSO-FCM clustering for ten images. It is found that the conventional FCM algorithm has a great probability to fall into a local optimum. This time the conventional FCM could not get the minimum value of the objective function thus creates validity functions showing inferior performance.

We also list an example to explain the significance of the PSO-FCM algorithm. Fig. 2(a) (b) is one of the ten double-feature images which fall into a local optimum used conventional FCM. The four target clusters are to partition: cerebral-spinal fluid (CSF), skull, intracranial tissues and fat. Fig. 2 (c), (d), (e) and (f) show the segmentation results obtained by using a conventional FCM algorithm and Fig. 2 (g), (h), (i) and (j) show the results of the PSO-FCM algorithm, respectively. As can be seen, the conventional FCM misclassifies skull and the intracranial tissues like Fig. 2 (f) and Fig. 2 (j) because it fall into a local optimum. Thus the PSO-FCM algorithm makes a significant improvement to avoid this issue in the conventional FCM algorithm.

Table 4. The clustering results of ten 2-dimensional images using various FCM techniques

Image number	Techniques	$J_{obj}(10^7)$	V_{pc}	V_{pe}	V_{xb}
1	FCM	1.883	0.8846	0.2152	0.2917
	PSO-FCM	1.355	0.9327	0.1411	0.0163
2	FCM	2.369	0.8618	0.2614	0.1140
	PSO-FCM	2.369	0.8618	0.2613	0.1140
3	FCM	1.876	0.9110	0.1814	0.0250
	PSO-FCM	1.876	0.9110	0.1814	0.0250
4	FCM	2.341	0.8587	0.2619	0.2732
	PSO-FCM	1.853	0.9050	0.1896	0.0386
5	FCM	2.562	0.8791	0.2260	0.2289
	PSO-FCM	2.213	0.9150	0.1736	0.0274
6	FCM	2.121	0.9075	0.1883	0.0285
	PSO-FCM	2.121	0.9075	0.1883	0.0285
7	FCM	1.423	0.8835	0.2184	0.1767
	PSO-FCM	1.423	0.8835	0.2184	0.1767
8	FCM	1.897	0.8742	0.2386	0.1239
	PSO-FCM	1.897	0.8742	0.2386	0.1239
9	FCM	2.118	0.8753	0.2563	0.0351
	PSO-FCM	2.118	0.8753	0.2563	0.0351
10	FCM	2.200	0.8936	0.2130	0.0289
	PSO-FCM	2.200	0.8936	0.2130	0.0289

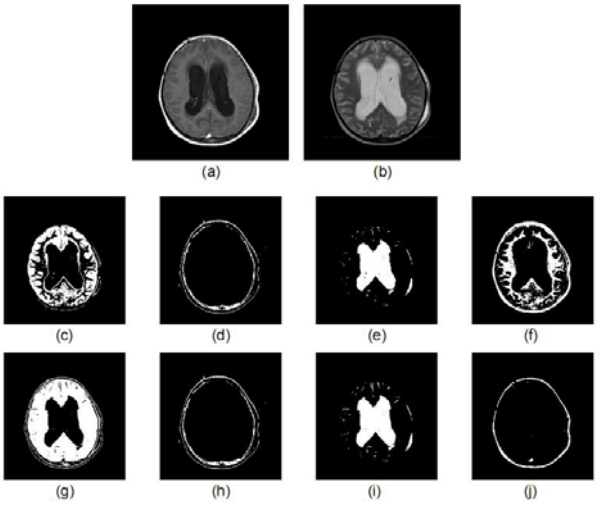


Fig. 2. (a) T1-CM MR image and (b) T2-AXIAL MR image used for the study which take conventional FCM into a local optimum. Segmented images of the two-dimensional MR image using FCM (c)(d)(e)(f) and PSO-FCM (g)(h)(i)(j).

4 Conclusion

FCM is a typical unsupervised pattern recognition method for medical image segmentation by clustering image pixels according to their feature data properties. However, the conventional FCM algorithm may converge to a local minimum of the objective function and lead to non-optimized segmentation. In this paper, in order to direct FCM algorithm to converge to the global minimum of the objective function, a novel PSO-FCM algorithm is proposed. The new method is applied on MR images. Preliminary results show that, compared to the conventional FCM, along with the improved robustness, PSO-FCM algorithm decreases iteration number when clustering number is large. Evaluation by using various validity functions suggests that PSO-FCM creates enhanced clustering performance.

Acknowledgments. This work was supported by The National Natural Science Foundation of China (Grant No.60773093,60873209,60970107,60970108), The Key Program for Basic Research of Shanghai (Grant No.08JC1411800) , The Ministry of Education and Intel joint research foundation (Grant No.MOE-INTEL-08-11), The Science and Technology Commission of Shanghai Municipality(09510701600), IBM SUR Funding and CRL JP Funding.

References

1. Bezdek, J., Hall, L., Clarke, L.: Review of MR image segmentation using pattern recognition. Med. Phys. 20, 1033–1048 (1993)
2. Brandt, M.E., Bohan, T.P., Kramer, L.A., Fletcher, J.M.: Estimation of CSF, white matter and gray matter volumes in hydrocephalic children using fuzzy clustering of MR images. Compute. Med. Imaging Graph. 18, 25–34 (1994)
3. Clark, M.C., Hall, L.O., Goldgof, D.B., Clarke, L.P., Velthuizen, R.P., Silbiger, M.S.: MRI segmentation using fuzzy clustering techniques. IEEE Eng. Med. Biol. 13, 730–742 (1994)
4. Pham, D.L., Prince, J.L.: Adaptive fuzzy segmentation of magnetic resonance images. IEEE Trans. Med. Imaging 18, 737–752 (1999)
5. Lyer, N.S., Kandel, A., Schneider, M.: Feature-based fuzzy classification for interpretation of mammograms. Fuzzy Sets Syst. 114, 271–280 (2002)
6. Yang, M.S., Hu, Y.J., Lin, K.C.R., Lin, C.C.L.: Segmentation techniques for tissue differentiation in MRI of Ophthalmology using fuzzy clustering algorithms. Magn. Reson. Imaging 20, 173–179 (2002)
7. Chuang, K.-S., Tzeng, H.-L., Chen, S., Wu, J., Chen, T.-J.: Fuzzy c-means clustering with spatial information for image segmentation. Compute. Med. Imaging Graph. 30, 9–15 (2006)
8. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact, well-separated clusters. Journal of Cybernetics 3, 32–57 (1973)
9. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms, pp. 65–70. Plenum press, New York (1981)
10. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proc. of the 6th International Symposium on Micro Machine and Human Scince, Nagoya, Japan, pp. 39–43 (1995)

11. Kennedy, J., Engelbrech, R.C.: Particle swarm optimization. In: Proc. of the IEEE International Conference on Neural Networks, Piscataway, NJ, vol. 4, pp. 1942–1948 (1995)
12. Kennedy, J.: The particle swarm: Social adaptation of knowledge. In: Proceedings of the 1997 International Conference on Evolutionary Computation, Indianapolis, Indiana, pp. 303–308. IEEE Service Center, Piscataway (1997)
13. Hall, L.O., Ozyurt, I.B., Bezdek, J.C.: Clustering with a genetically optimized approach. IEEE Trans. Evolutionary Computation 3(2), 103–112 (1999)
14. Alata, M., Molhim, M., Ramini, A.: Optimizing of Fuzzy C-Means Clustering Algorithm Using GA. World Academy of Science, Engineering and Technology 39 (2008)
15. Liu, H.-C., Jeng, B.-C., Yih, J.-M., Yu, Y.-K.: Fuzzy C-Means Algorithm Based on Standard Mahalanobis Distances. In: Proceedings of the 2009 International Symposium on Information Processing (ISIP 2009), Huangshan, P. R, China, pp. 422–427 (2009)
16. Liu, H.-C., Yih, J.-M., Lin, W.-C., Liu, T.-S.: Fuzzy C-Means Algorithm Based on PSO and Mahalanobis Distance. International Journal of Innovative Computing, Information and Control 5, 5033–5040 (2009)
17. Liu, H., Sun, J., Wu, H., Teng, S., Tan, Z.: High Resolution Sonar Image Segmentation by PSO based Fuzzy Cluster Method. In: 2010 Fourth International Conference on Genetic and Evolutionary Computing (2010)
18. Yih, J.-M., Lin, Y.-H., Liu, H.-C.: Clustering Analysis Method based on Fuzzy C-Means Algorithm of PSO and PPSO with Application in Real Data. International Journal of Geology 4(1) (2007)
19. Ichihashi, H., Honda, K., Notsu, A., Ohta, K.: Fuzzy c-Means Classifier with Particle Swarm Optimization. In: 2008 IEEE International Conference on Fuzzy Systems (2008)
20. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: The Fuzzy c-Means Clustering Algorithm. Computers and Geosciences 10, 191–203 (1984)
21. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence (2009)
22. Bezdek, J.C.: Cluster validity with fuzzy sets. J. Cybern. 3, 58–73 (1974)
23. Bezdek, J.C.: Mathematical models for systematic and taxonomy. In: Proceedings of Eigth International Conference on Numerical Taxonomy, San Francisco, pp. 143–166 (1975)
24. Xie, X.L., Beni, G.A.: Validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. 3, 841–846 (1991)
25. Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for the fuzzy c-means method. In: Proceedings of Fifth Fuzzy System Symposium, pp. 247–250 (1989)
26. The Whole Brain Atlas, http://www.med.harvard.edu/AANLIB/home.html

k-Means Clustering of Asymmetric Data

Dominik Olszewski

Faculty of Electrical Engineering,
Warsaw University of Technology, Poland
olszewsd@ee.pw.edu.pl

Abstract. In this paper, an asymmetric k-means clustering algorithm is presented. The asymmetric version of this algorithm is derived using the asymmetric coefficients, which convey the information provided by the asymmetry in analyzed data sets. The formulation of the asymmetric k-means algorithm is motivated by the fact that, when an analyzed data set has the asymmetric nature, a data analysis algorithm should properly adjust to this nature. The traditional k-means approach using the symmetric dissimilarities does not apply correctly to this kind of phenomenon in data. We propose the k-means algorithm using the asymmetric coefficients, which has the ability to reflect the asymmetric relationships between objects in analyzed data sets. The results of our experimental study on real data show that the asymmetric k-means approach outperforms its symmetric counterpart.

Keywords: Clustering, k-means algorithm, Asymmetry, Sound recognition, Human heart rhythm recognition.

1 Introduction

The k-means clustering algorithm [2,9,10,20] is a well-known statistical data analysis tool used in order to form arbitrary settled number of clusters in an analyzed data set. The algorithm aims to separate clusters of possibly most similar objects. Object represented as a vector of d features can be interpreted as a point in d-dimensional space. Hence, the k-means algorithm can be formulated as follows: given n points in d-dimensional space, and the number k of desired clusters, the algorithm seeks a set of k clusters so as to minimize the sum of squared dissimilarities between each point and its cluster centroid. The name "k-means" was introduced in [10], however, the algorithm, itself, was formulated by H. Steinhaus in [20].

The algorithm, in its traditional form, operates in the same way on every kind of data set that is analyzed. However, certain data sets appear to have asymmetric properties, which can significantly affect the results of clustering. Such asymmetric properties occur, for example, in case of hierarchical associations in data sets [11,12]. Therefore, data analysis methods and algorithms should incorporate mechanisms to properly handle this kind of data.

The asymmetric k-means algorithm, proposed in this paper, is an example of the approach, which includes the information associated with the asymmetry in analyzed data.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 243–254, 2012.

The asymmetric version of the k-means clustering algorithm was introduced in [18]. However, the asymmetry in the algorithm in [18] arises caused by use of dissimilarities, which are defined as asymmetric (for example, the Kullback-Leibler divergence). On the other hand, in this paper, we propose the asymmetric k-means algorithm using the symmetric similarities, which are asymmetrized by employing the asymmetric coefficients. This kind of approach provides a proper adjustment to the asymmetric relationships in analyzed data, and this way it provides an accurate reflection of the asymmetric nature of data.

The method, proposed in this paper, can be also regarded as an example of the hybrid artificial intelligence approach [1,4,5], because it operates successfully on two kinds of analyzed data – symmetric and asymmetric.

The rest of this paper is organized as follows: in Section 2, the traditional symmetric version of the k-means algorithm is presented; in Section 3, the asymmetric relationships in data sets are considered; in Section 4, the asymmetric version of the k-means algorithm using the asymmetric coefficients is described; in Section 5, our experimental results are reported along with the experimental setup information; while Section 6 summarizes the whole paper, and points out certain directions of future research.

2 Traditional k-Means Algorithm

The traditional k-means clustering algorithm consists of two alternating steps:

Step 1. Forming of the clusters: The algorithm iterates over the entire data set, and allocates each object to the cluster represented by the centroid – nearest to this object. The nearest centroid is determined with use of a chosen dissimilarity measure. Hence, for each object in an analyzed data set, the following minimal squared dissimilarity has to be found:

$$\min_{j} \; d^2 \left(x_i, c_j \right) \; , \tag{1}$$

where $d\left(\bullet, \bullet\right)$ is a chosen dissimilarity measure, x_i, $i = 1, \ldots, n_j$ is the object in the jth cluster, c_j, $j = 1, \ldots, k$ is the centroid of the jth cluster, and n_j, $j = 1, \ldots, k$ is the number of objects in the jth cluster.

Step 2. Finding centroids for the clusters: For each cluster, a centroid is determined on the basis of objects belonging to this cluster. The algorithm calculates centroids of the clusters so as to minimize a formal objective function, the error distortion:

$$e\left(X_j\right) \; = \; \sum_{i=1}^{n_j} d^2 \left(x_i, c_j \right) \; , \tag{2}$$

where X_j, $j = 1, \ldots, k$ is the jth cluster, x_i, $i = 1, \ldots, n_j$ is the object in the jth cluster, c_j, $j = 1, \ldots, k$ is the centroid of the jth cluster, n_j, $j = 1, \ldots, k$ is the number of objects in the jth cluster, k is the number of clusters, and $d\left(\bullet, \bullet\right)$ is a chosen dissimilarity measure.

In case, when the Ward's criterion is applied in the k-means algorithm, the centroids of the clusters are computed as the arithmetic averages of objects in the clusters.

Both these steps must be carried out with the same dissimilarity measure, in order to guarantee the monotone property of the k-centroids algorithm.

Steps 1 and 2 have to be repeated until the termination condition is met. The termination condition might be either reaching convergence of the iterative application of the objective function (3), or reaching the pre-defined number of cycles.

After each cycle (Step 1 and 2), the value of the error function (2) needs to be computed for the entire analyzed data set, in order to track the convergence of the whole clustering process:

$$e\left(X\right) \;=\; \sum_{j=1}^{k} \sum_{i=1}^{n_j} d^2\left(x_i, c_j\right) \;, \tag{3}$$

where X is the analyzed set of objects, and the rest of the notation is described in (2).

The most popular choice of the dissimilarity measure is the Euclidean distance. In this paper, we will use the Euclidean distance as a basis for formulation of the asymmetric dissimilarity measure using the asymmetric coefficients.

3 Asymmetry in Data

The problem of asymmetry in data analysis was widely studied in the literature. The research of A. Okada and T. Imaizumi [14,15,16,17] is focused on using the dominance point governing asymmetry in the proximity relationships among objects, represented as points in the multidimensional Euclidean space. They claim that ignoring or neglecting the asymmetry in proximity analysis discards potentially valuable information. On the other hand, B. Zielman and W. Heiser in [21] consider the models for asymmetric proximities as a combination of a symmetric similarity component and an asymmetric dominance component. The authors of [11] propose the asymmetric version of the Self-Organizing Map, which is extended in [19]. Finally, the paper [18] introduces the asymmetric version of the k-means clustering algorithm using the dissimilarities, which are defined as asymmetric (for example, the Kullback-Leibler divergence).

When an analyzed data set appears to have asymmetric properties, the symmetric measures of similarity or dissimilarity (for example, the most popular Euclidean distance) does not apply properly to this phenomenon, and for most pairs of data points, they produce small values (similarities) or big values (dissimilarities). Consequently, they do not reflect accurately the relationships between objects. The asymmetry in data set arises, for example, in case, when the data

associations have a hierarchical nature. The hierarchical connections in data are closely related to the asymmetry. This relation has been noticed in [12]. In case of the dissimilarity, when it is computed in the direction – from a more general entity to a more specific one – it should be greater than in the opposite direction. As stated in [11], asymmetry can be interpreted as a particular type of hierarchy.

An idea to overcome this problem is to employ the asymmetric similarities and dissimilarities. They should be applied in algorithms in such way, so that they would properly reflect the hierarchical asymmetric relationships between objects in an analyzed data set. Therefore, it should be guaranteed that their application is consistent with the hierarchical associations in data. This can be achieved by use of the asymmetric coefficients, inserted in the formulae of symmetric measures. This way, we can obtain the asymmetric measures on the basis of the symmetric ones, for example, on the basis of the Euclidean distance. The asymmetric coefficients should assure the consistence with the hierarchy. Hence, in case of the dissimilarities, they should assure greater values in the direction – from more general concept to more specific one.

Our experimental study concerns the sound signals clustering and human heart rhythms clustering, and confirms the existence of the asymmetry in these data sets. However, the phenomenon of asymmetry can occur in various other data sets of different nature. For example, in the work [11], where the asymmetric Self-Organizing Map was introduced, the experimental research was carried out in the field of textual data analysis.

3.1 Asymmetric Coefficients

Asymmetric coefficients convey the information provided by asymmetry. Two coefficients were introduced in [13]. The first one is derived from the fuzzy logic similarity, and the second one formulated on the basis of the Kullback-Leibler divergence. Both of these quantities are widely used in statistics and probability theory. In our experimental study, we have used the first of these coefficients.

Hence, the fuzzy-logic-based asymmetric coefficient is formulated as follows:

$$a_i \;=\; \frac{|f_i|}{\max_j \left(|f_j|\right)}\,, \tag{4}$$

where f_i are the features of objects in the analyzed data set (f_i are the entries of the vectors representing the objects), and $|\bullet|$ is the L_1-norm meaning the number of objects possessing the feature given as the argument.

This coefficient takes values in the $\langle 0, 1\rangle$ interval. Intuitively speaking, it will become large for general (broad) concepts with large L_1-norm.

Note that the asymmetric coefficients must be computed and assigned to each feature of every object in the analyzed data set.

4 Asymmetric k-Means Algorithm

In order to formulate the asymmetric version of the k-means clustering algorithm, we will refer to Steps 1 and 2, to objective function 2, and to function 3, presented in Section 2.

The asymmetric k-means algorithm is derived in three steps:

Step 1. Transform a symmetric dissimilarity (for example, the Euclidean distance) into a similarity:

$$S_{ij}^{\mathrm{SYM}} \;=\; C - d^2\left(x_i, x_j\right) \;, \tag{5}$$

where $d^2\left(x_i, x_j\right)$ is the squared Euclidean distance between objects x_i and x_j, and the constant C is the upper boundary of the squared Euclidean distance.

Step 2. Transform the symmetric similarity into the asymmetric similarity:

$$S_{ij}^{\mathrm{ASYM}} \;=\; a_i \left(C - d^2\left(x_i, x_j\right)\right) \;, \tag{6}$$

where a_i is the asymmetric coefficient defined in Subsection 3.1, in (4), and the rest of notation is described in (5). The asymmetric similarity defined this way, with use of the asymmetric coefficient guarantees the consistency with the asymmetric hierarchical associations among objects in a data set.

Step 3. Insert the asymmetric similarity in (1), (2), and (3), in order to obtain:

$$\max_j \; a_i \left(C - d^2\left(x_i, c_j\right)\right) \;, \tag{7}$$

$$E\left(X_j\right) \;=\; \sum_{i=1}^{n_j} a_i \left(C - d^2\left(x_i, c_j\right)\right) \;, \tag{8}$$

$$E\left(X\right) \;=\; \sum_{j=1}^{k} \sum_{i=1}^{n_j} a_i \left(C - d^2\left(x_i, c_j\right)\right) \;, \tag{9}$$

where $E\left(\bullet\right)$ is the energy function, which needs to be maximized in order to compute the clusters centroids in Step 2, in Section 2, and the rest of the notation is explained in (1), (2), and (9).

Formulae (1) and (2) should be used in Steps 1 and 2, respectively, while the iterative computation of values of (9) should allow for tracking of the convergence of the entire clustering process.

The error function $e\left(\bullet\right)$ is now changed to the energy function $E\left(\bullet\right)$, since the dissimilarity measure $d\left(\bullet, \bullet\right)$ has been changed to the similarity measure $S^{\mathrm{ASYM}}\left(\bullet, \bullet\right)$.

An important property of the asymmetric k-means algorithm, proposed in this paper, is that it maintains the simplicity of the traditional symmetric approach, and does not increase the computational complexity.

5 Experiments

Our experimental study aims to confirm that the asymmetric version of the k-means clustering algorithm, proposed in this paper, is superior over the traditional symmetric form of this algorithm. The experiments have been carried out in the two fields: in the field of sound signals clustering, and in the field of human heart rhythm signals clustering.

The sound signals, we have analyzed, were the piano music recordings, and the human heart rhythm signals were analyzed on the basis of the ECG recordings derived from the MIT-BIH ECG Databases [6].

5.1 Evaluation Criteria

In case of both parts of our experiments, we have compared the results obtained with use of the symmetric and asymmetric k-means algorithms. As the basis of the comparisons, i.e., as the evaluation criteria, we have used the accuracy degree [18,19], and the entropy measure [11,19].

In general, the results of clustering can be assessed with use of two groups of evaluation criteria [7] (also called as the validity indeces). First group are the external criteria, which are computed using the ground knowledge about the clustered data, i.e., what should be the correct result of clustering. These criteria are much easier to formulate, and they allow for the precise assessment of the clustering results, however, they are useless in real-world clustering problems, where no additional information about analyzed data is available. Second group are the internal criteria, which are computed without using the ground knowledge about the clustered data. Formulation of these criteria is, naturally, more difficult, however, they can be employed to assess the results of clustering in real-life problems. Therefore, their usefulness in data analysis is of much value. More information on the issue of clustering output assessment can be found, for example, in [7] and in [8].

In case of both parts of our empirical study, the ground knowledge about the data was known. Therefore, an application of external evaluation criterion (the accuracy degree) was possible. In order to provide more reliable assessment of the clustering results, we have used also the second internal evaluation criterion (the entropy measure).

Hence, the following two evaluation criteria have been used:

1. **Accuracy Degree.** This evaluation criterion determines the number of correctly assigned objects divided by the total number of objects.
   Hence, for the ith cluster, the accuracy degree is determined as follows:

$$q_i = \frac{m_i}{n_i}, \tag{10}$$

   where m_i, $i = 1, \ldots, k$ is the number of objects correctly assigned to the ith cluster, n_i, $i = 1, \ldots, k$ is the number of objects in the ith cluster, and k is the number of clusters.

And, for the entire data set, the total accuracy degree is determined as follows:

$$q_{\text{total}} = \frac{m}{n}, \tag{11}$$

where m is the total number of correctly assigned objects, and n is the total number of objects in the entire data set.

The accuracy degrees q_i and the total accuracy degree q_{total} assume values in the interval $\langle 0, 1\rangle$, and naturally, greater values are preferred.

The total accuracy degree q_{total} was used in our experimental study as the main basis of the clustering accuracy comparison of the two investigated k-means approaches – the symmetric and the asymmetric.

2. **Entropy Measure.** This evaluation criterion determines the number of overlapping objects divided by the total number of objects in a data set. This means, the number of objects, which are in the overlapping area between clusters, divided by the total number of objects. The objects belonging to the overlapping area are determined on the basis of the ratio of similarities between them and the clusters centroids. If this ratio is in the interval $\langle 0.9, 1.1\rangle$, then the corresponding object is said to be in the overlapping area. In other words, the entropy measure determines the uncertainty for the classification of objects that belong to the same cluster.

   The entropy measure is determined as follows:

$$I = \frac{\mu}{n}, \tag{12}$$

   where μ is the number of overlapping objects in the data set, and n is the total number of objects in the data set.

   The entropy measure assumes values in the interval $\langle 0, 1\rangle$, and, smaller values are desired.

5.2 Feature Extraction

Feature discovery process preceding the actual clustering is an important stage of data pre-processing. It has a strong impact on the final accuracy of clustering, and consequently, on the performance of the whole analysis. Feature discovery aims to form possibly smallest set of most relevant, informative, and discriminative features. A proper choice of the feature set results in higher clustering quality.

Feature discovery approach, used for our asymmetric version of the k-means algorithm, is the feature extraction method based on the discrete Fourier transform (DFT). In general, the DFT-based feature extraction is described, for example, in [3]. In our experimental study, feature extraction was carried out according to Procedure 1.

Procedure 1. *Features of analyzed signals are retrieved according to the following procedure:*

Step 1. *Separate N time intervals from the discrete-time signal representation. All intervals are of the same length, hence all have the same number of samples. As a result, one obtains N functions in the discrete-time domain: $f_i(t_j)$, $i = 1, \ldots, N$, $j = 1, \ldots, K$, where K is the number of samples in each separated time interval.*

Step 2. *Perform the DFT on each of the N discrete-time functions obtained in the Step 1, considering the absolute values of the complex DFT-vectors entries. As a result, one obtains N functions in the discrete-frequency domain: $\left|\widetilde{f}_i(\omega_l)\right| = \left|\mathcal{F}(f_i)(\omega_l)\right|$, $i = 1, \ldots, N$, $l = 1, \ldots, \text{floor}\left(\frac{K}{2}\right)$ (the rest of the DFT result is mirrored – it does not contain any new information), where $\text{floor}(\bullet)$ returns the largest integer that is less than or equal to the argument.*

Step 3. *Calculate the average DFT result on the basis of the results of DFT for each of the intervals. The average DFT result denotes the DFT-vector, which entries are the arithmetic averages – calculated on the basis of the corresponding entries of the partial DFT-vectors.*

$$\left|\widetilde{f}_{\text{AVG}}(\omega_l)\right| = \frac{1}{N}\sum_{i=1}^{N}\left|\widetilde{f}_i(\omega_l)\right| \tag{13}$$

where $i = 1, \ldots, N$, $l = 1, \ldots, \text{floor}\left(\frac{K}{2}\right)$.

Step 4. *Normalize the function obtained in Step 3.*

$$f_{\text{FE}}(\omega_l) = \frac{1}{\sum_{p=1}^{r}\left|\widetilde{f}_{\text{AVG}}(\omega_p)\right|}\left|\widetilde{f}_{\text{AVG}}(\omega_l)\right|, \tag{14}$$

where $f_{\text{FE}}(\omega_l)$ is the function representing the set of obtained features, $l = 1, \ldots, r$, and $r = \text{floor}\left(\frac{K}{2}\right)$.

As the final result of Procedure 1, one obtains the discrete function f_{FE} representing the retrieved vector of features of a single signal.

The number of the time intervals N is settled arbitrary. We provide no principled way to determine it, what can be regarded as a drawback of this feature extraction method.

Normalization in Step 4 of Procedure 1 is a common practice in pattern recognition. The normalized features are of benefit in many contexts of multivariate analysis, not only in clustering, but also, for example, in discriminant analysis. Normalization of features especially accounts in the field of sound recognition, which was one of the areas of our experiments. It filters out the irrelevant feature of loudness (feature of loudness should not affect the results of recognition – a music piece played softly remains the same piece if it is played loudly, but, without normalization, the features would change). In other words, Step 4 determines the relative intensities as the characteristic feature set, and not the absolute values, which would be largely influenced by the irrelevant features, like levels of loudness in case of sound recognition.

5.3 Piano Music Composer Clustering

In this part of our experiments, we have tested our asymmetric k-means algorithm and the classical k-means algorithm forming three clusters representing three piano music composers: Johann Sebastian Bach, Ludwig van Beethoven, and Fryderyk Chopin. Each music piece was represented by a 30-seconds sound signal sampled with the 44100 Hz frequency. The entire data set was composed of 32 sound signals. Feature extraction process was carried out according to Procedure 1, described in Subsection 5.2. The DFT was implemented with the fast Fourier transform (FFT) algorithm. Sampling signals with the 44100 Hz frequency resulted in the 44100/2 Hz value of the upper boundary of the FFT result range.

The results of this part of our experiments are reported in Table 1 and in Table 2. These tables present the accuracy degrees and the entropy measures corresponding to the symmetric and asymmetric k-means algorithms.

Table 1. Accuracy degrees of the piano music composer clustering

	Symmetric k-means	Asymmetric k-means
J.S. Bach	$10/11 = 0.9091$	$10/11 = 0.9091$
L. van Beethoven	$9/12 = 0.7500$	$11/12 = 0.9167$
F. Chopin	$5/9 = 0.5556$	$8/9 = 0.8889$
q_{total}	$24/32 = 0.7500$	$29/32 = 0.9063$

Table 2. Entropy measures of the piano music composer clustering

	Symmetric k-means	Asymmetric k-means
I	$7/32 = 0.2188$	$5/32 = 0.1563$

The results of this part of our experimental study show the superiority of the asymmetric k-means algorithm over its symmetric counterpart. The asymmetric approach leads to the higher clustering accuracy measured on the basis of the total accuracy degree (0.9063 vs. 0.7500), and also, it leads to the lower cluster overlapping determined on the basis of the entropy measure (0.1563 vs. 0.2188).

5.4 Human Heart Rhythms Clustering

In this part of our experiments, we have investigated the asymmetric k-means algorithm and the traditional form of this method forming three clusters representing three types of human heart rhythms: normal sinus rhythm, atrial arrhythmia, and ventricular arrhythmia. This kind of clustering can be interpreted as the cardiac arrhythmia detection and recognition based on the ECG recordings. In general, the cardiac arrhythmia disease may be classified either by rate

(tachycardias – the heart beat is too fast, and bradycardias – the heart beat is too slow) or by site of origin (atrial arrhythmias – they begin in the atria, and ventricular arrhythmias – they begin in the ventricles). Our clustering recognizes the normal rhythm, and, also, recognizes arrhythmias originating in the atria, and in the ventricles. We analyzed 20-minutes ECG holter recordings sampled with the 250 Hz frequency. The entire data set was composed of 63 ECG signals. Feature extraction was carried out in the same way, like it was done with the piano music composer clustering, i.e., according to Procedure 1, described in Subsection 5.2.

The results of this part of our experiments are demonstrated in Table 3 and in Table 4, which is constructed in the same way as in Subsection 5.3.

Table 3. Accuracy degrees of the human heart rhythms clustering

	Symmetric k-means	Asymmetric k-means
Normal Rhythm	$16/18 = 0.8889$	$18/18 = 1.0000$
Atrial Arrhythmia	$18/23 = 0.7826$	$21/23 = 0.9130$
Ventricular Arrhythmia	$15/22 = 0.6818$	$19/22 = 0.8636$
q_{total}	$49/63 = 0.7778$	$58/63 = 0.9206$

Table 4. Entropy measures of the human heart rhythms clustering

	Symmetric k-means	Asymmetric k-means
I	$8/63 = 0.1270$	$4/63 = 0.0635$

It is clear that, in the case of the ECG recording clustering, the asymmetric k-means algorithm, again, outperformed the symmetric one, by providing the higher clustering accuracy (total accuracy degree: 0.9206 vs. 0.7778), and the lower clustering uncertainty (entropy measure: 0.0635 vs. 0.1270).

6 Summary and Future Research

In this paper, the asymmetric version of the well-known k-means clustering algorithm was proposed. The traditional k-means algorithm using the symmetric dissimilarities (like, the most popular, Euclidean distance) does not reflect properly the asymmetric relationships in analyzed data. The asymmetry arises in data sets, for example, in case when objects in data sets are associated in a hierarchical way. We proposed using the asymmetric coefficients, which convey the information provided by the asymmetry, in order to asymmetrize the k-means algorithm. These coefficients were used to obtain the asymmetric similarities in the algorithm, and this way, the asymmetric k-means algorithm, proposed in this paper, correctly adjusted to the asymmetric relationships between objects in analyzed data sets.

Our experimental study was carried out on the two data sets: piano music data set and human heart rhythms data set. The results of the conducted empirical research confirmed the superiority of the asymmetric approach in k-means clustering over the classical symmetric approach.

Future study may concern utilizing different form of asymmetric coefficients – not only the fuzzy-logic-based coefficient, like it was done in this paper. Furthermore, the experimental research may be carried out on data sets of different nature – other than signals. Finally, future study may be focused on formulating the asymmetric version of other data analysis methods and algorithms. This goal can be achieved either by using the asymmetric coefficients, like it was proposed in this paper, or by using different quantities, which will incorporate the information associated with the asymmetry in analyzed data sets.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. Neurocomputing 72(13/15), 2729–2730 (2009)
2. Biau, G., Devroye, L., Lugosi, G.: On the Performance of Clustering in Hilbert Spaces. IEEE Transactions on Information Theory 54(2), 781–790 (2008)
3. Chengalvarayan, R., Deng, L.: HMM-Based Speech Recognition Using State-Dependent, Discriminatively Derived Transforms on Mel-Warped DFT Features. IEEE Transactions on Speech and Audio Processing 2(3), 243–256 (1997)
4. Corchado, E., Abraham, A., Carvalho, A.: Hybrid Intelligent Algorithms and Applications. Information Sciences 180(14), 2633–2634 (2010)
5. Corchado, E., Graña, M., Woźniak, M.: New Trends and Applications on Hybrid Artificial Intelligence Systems. Neurocomputing 75(1), 61–63 (2012)
6. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 101(23), e215–e220, circulation Electronic Pages (2000), http://circ.ahajournals.org/cgi/content/full/101/23/e215
7. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. Journal of Intelligent Information Systems 17(2/3), 107–145 (2001)
8. Handl, J., Knowles, J., Kell, D.B.: Computational Cluster Validation in Post-genomic Data Analysis. Bioinformatics 21(15), 3201–3212 (2005)
9. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An Efficient k-Means Clustering Algorithm: Analysis and Implemetation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 881–892 (2002)
10. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
11. Martín-Merino, M., Muñoz, A.: Visualizing Asymmetric Proximities with SOM and MDS Models. Neurocomputing 63, 171–192 (2005)
12. Muñoz, A., Martin, I., Moguerza, J.M.: Support Vector Machine Classifiers for Asymmetric Proximities. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 217–224. Springer, Heidelberg (2003)

13. Muñoz, A., Martín-Merino, M.: New Asymmetric Iterative Scaling Models for the Generation of Textual Word Maps. In: Proceedings of the International Conference on Textual Data Statistical Analysis JADT 2002, pp. 593–603 (2002)
14. Okada, A.: An Asymmetric Cluster Analysis Study of Car Switching Data. In: Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Heidelberg (2000)
15. Okada, A., Imaizumi, T.: Asymmetric Multidimensional Scaling of Two-Mode Three-Way Proximities. Journal of Classification 14(2), 195–224 (1997)
16. Okada, A., Imaizumi, T.: Joint Space Model for Multidimensional Scaling of Two-Mode Three-Way Asymmetric Proximities. In: Innovations in Classification, Data Science, and Information Systems. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 371–378. Springer, Heidelberg (2003)
17. Okada, A., Imaizumi, T.: Multidimensional Scaling of Asymmetric Proximities with a Dominance Point. In: Advances in Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 307–318. Springer, Heidelberg (2007)
18. Olszewski, D.: Asymmetric k-Means Algorithm. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part II. LNCS, vol. 6594, pp. 1–10. Springer, Heidelberg (2011)
19. Olszewski, D.: An Experimental Study on Asymmetric Self-Organizing Map. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 42–49. Springer, Heidelberg (2011)
20. Steinhaus, H.: Sur la Division des Corp Matériels en Parties. Bulletin de l'Académie Polonaise des Sciences, C1. III 4(12), 801–804 (1956)
21. Zielman, B., Heiser, W.J.: Models for Asymmetric Proximities. British Journal of Mathematical and Statistical Psychology 49, 127–146 (1996)

A Max Metric to Evaluate a Cluster

Hosein Alizadeh[1], Hamid Parvin[2], Sajad Parvin[2], Zahra Rezaei[2], and Moslem Mohamadi[2]

[1] Islamic Azad University, Mahdi Shahr Branch, Mahdi Shahr, Iran
halizadeh@iust.ac.ir
[2] Islamic Azad University, Nourabad Mamasani Branch, Mamasani Nourabad, Iran
hamidparvin@mamasaniiau.ac.ir,
{s.parvin,rezaei,mohamadi}@iust.ac.ir

Abstract. In this paper a new criterion for clusters validation is proposed. This new cluster validation criterion is used to approximate the goodness of a cluster. The clusters which satisfy a threshold of the proposed measure are selected to participate in clustering ensemble. To combine the chosen clusters, some methods are employed as aggregators. Employing this new cluster validation criterion, the obtained ensemble is evaluated on some well-known and standard datasets. The empirical studies show promising results for the ensemble obtained using the proposed criterion comparing with the ensemble obtained using the standard clusters validation criterion. Besides to reach the best results, the method gives an algorithm based on which one can find how to select the best subset of clusters from a pool of clusters.

Keywords: Clustering Ensemble, Stability Measure, Extended EAC, Co-association Matrix, Cluster Evaluation.

1 Introduction

Data clustering or unsupervised learning is an important and very difficult problem. The objective of clustering is to partition a set of unlabeled objects into homogeneous groups or clusters [3], [4] and [10]. There are many applications that use clustering techniques to discover latent structures of data, such as data mining [11], information retrieval [2], image segmentation [9], linkage learning [15], and machine learning. In real-world problems, clusters can appear with different shapes, sizes, data sparseness's, and degrees of separation. Clustering techniques require the definition of a similarity measure between patterns. Since there is no prior knowledge about cluster shapes, choosing a specific clustering method is not easy [16]. Studies in the last few years have tended to combinational methods. Cluster ensemble methods attempt to find better and more robust clustering solutions by fusing information from several primary data partitions [8].

Fern and Lin [8] have suggested a clustering ensemble approach which selects a subset of solutions to form a smaller but better-performing cluster ensemble than using all primary solutions. The ensemble selection method is designed based on quality and diversity, the two factors that have been shown to influence cluster

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 255–266, 2012.

ensemble performance. This method attempts to select a subset of primary partitions which simultaneously has both the highest quality and the most diversity. The Sum of Normalized Mutual Information, SNMI [5], [6] and [17], is used to measure the quality of each individual partition with respect to other partitions. Also, the Normalized Mutual Information, NMI, is employed to measure the diversity among partitions. Although the ensemble size in this method is relatively small, this method achieves significant performance improvement over full ensembles. Law et al. proposed a multi-objective data clustering method based on the selection of individual clusters produced by several clustering algorithms through an optimization procedure [13]. This technique chooses the best set of objective functions for different parts of the feature space from the results of base clustering algorithms. Fred and Jain [7] have offered a new clustering ensemble method which learns the pairwise similarities between points in order to facilitate a proper partition of the data without the a priori knowledge of the number and the shape of the clusters. This method which is based on cluster stability evaluates the primary clustering results instead of final clustering.

We propose a new criterion for clusters validation. Then we employ this criterion to select the more robust clusters in the final ensemble. We also propose a new method named Extended Evidence Accumulation Clustering, EEAC, to construct the matrix of similarity from these selected clusters. Finally, we apply a hierarchical method over the obtained matrix to extract the final partition.

Rest of this paper is organized as follows. In section 2, we explain the proposed method. Section 3 demonstrates results of our proposed method against traditional comparatively. Finally, we conclude in section 4.

2 Proposed Method

In this section, first our proposed clustering ensemble method is briefly outlined, and then its phases are described in detail.

The main idea of our proposed clustering ensemble framework is utilizing a subset of best performing primary clusters in the ensemble, rather than using all of clusters. Only the clusters that satisfy a stability criterion can participate in the combination. The cluster stability is defined according to Normalized Mutual Information, NMI. Fig. 1 depicts the proposed clustering ensemble procedure.

The manner of computing stability is described in the following sections in detail. To select a subset with the most stable clusters for combination, we apply a stability-threshold to each cluster. Different sizes of the most stable clusters are explored to find the best option. After selection phase, the selected clusters are used to construct the co-association matrix. Several methods have been proposed for combination of the primary results [1] and [17]. In this work, some clusters in the primary partitions may be absent (having been eliminated by the stability criterion). Since the original EAC method [5] cannot truly identify the pairwise similarity while there is only a subset of clusters, we present a new method for constructing the co-association matrix. We call this method: Extended Evidence Accumulation Clustering method, EEAC. Finally, we use a hierarchical clustering algorithm, like single-link method, to extract the final clusters out of this matrix. For more generality, some heuristic

consensus functions are also used as aggregators of selected clusters [17]. These heuristic consensus functions that are based on hypergraph partitioning and have first introduced by Strehl and Ghosh, are HperGraph Partitioning Algorithm (HGPA), Meta-Clustering Algorithm (MCLA) and Cluster-based Similarity Partitioning Algorithm (CSPA) [17].

2.1 Cluster Evaluation

Since goodness of a cluster is determined by all the data points, the goodness function $g_j(C_i,D)$ depends on both the cluster C_i and the entire dataset D, instead of C_i alone. The stability as measure of cluster goodness is used in [12]. Cluster stability reflects the variation in the clustering results under perturbation of the data by resampling.

A stable cluster is one that has a high likelihood of recurrence across multiple applications of the clustering method. Stable clusters are usually preferable, since they are robust with respect to minor changes in the dataset [13].

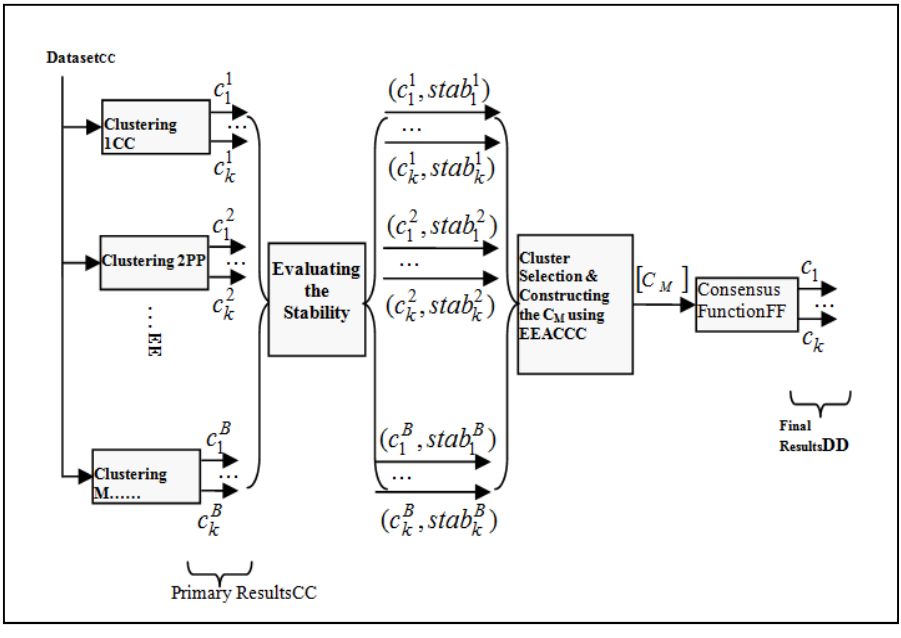


Fig. 1. Training phase of the Bagging method

Now assume that we want to compute the stability of cluster C_i. In this method first a set of partitionings over resampled datasets is provided which is called the reference set. In this notation D is resampled data and $P(D)$ is a partitioning over D. Now, the problem is: "How many times is the cluster C_i repeated in the reference partitions?" Denote by NMI(C_i,$P(D)$), the Normalized Mutual Information between the cluster C_i and a reference partition $P(D)$. Most previous works only compare a *partition with another partition* [17]. However, the stability used in [13] evaluates the similarity

between a *cluster and a partition* by transforming the cluster C_i to a partition and employing common partition to partition methods. To illustrate this method let $P_1 = P^a = \{C_i, D/C_i\}$ be a partition with two clusters, where D/C_i denotes the set of data points in D that are not in C_i.

Then we may compute a second partition $P_2 = P^b = \{C^*, D/C^*\}$, where C^* denotes the union of all "positive" clusters in $P(D)$ and others are in D/C^*. A cluster C_j in $P(D)$ is positive if more than half of its data points are in C_i. Now, define NMI(C_i,$P(D)$) by NMI(P^a,P^b) which is calculated as [6]:

$$NMI(P^a, P^b) = \frac{-2\sum_{i=1}^{k_a}\sum_{j=1}^{k_b} n_{ij}^{ab} \log\left(\frac{n_{ij}^{ab}.n}{n_i^a.n_j^b}\right)}{\sum_{i=1}^{k_a} n_i^a \log\left(\frac{n_i^a}{n}\right) + \sum_{j=1}^{k_b} n_j^b \log\left(\frac{n_j^b}{n}\right)} \tag{1}$$

where n is the total number of samples and n_{ij}^{ab} denotes the number of shared patterns between clusters $C_i^a \in P^a$ and $C_j^b \in P^b$; n_i^a is the number of patterns in the cluster i of partition a; also n_j^b are the number of patterns in the cluster j of partition b.

This computation is done between the cluster C_i and all partitions available in the reference set. Fig. 2 shows this method.

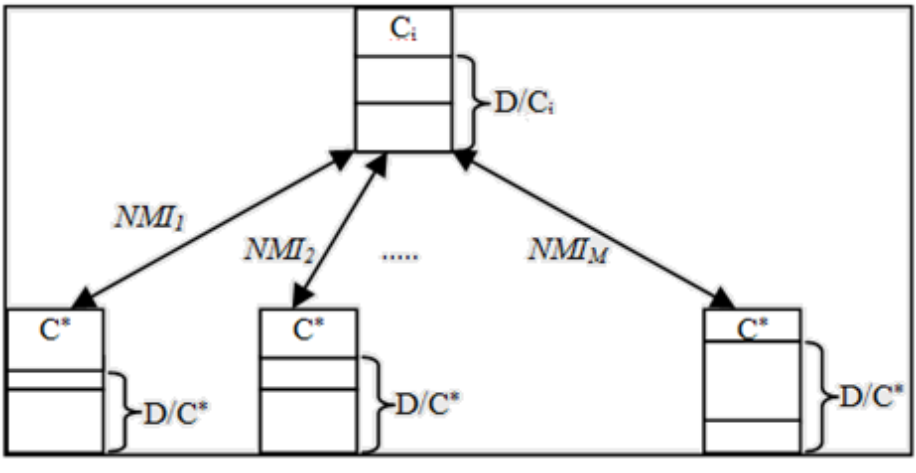


Fig. 2. Computing the Stability of Cluster C_i

NMI_i in Fig. 2 shows the stability of cluster C_i with respect to the i-th partition in reference set. The total stability of cluster C_i is defined as:

$$Stability(C_i) = \frac{1}{M}\sum_{i=1}^{M} NMI_i \tag{2}$$

where M is the number of partitions available in reference set. This procedure is applied for each cluster of every primary partition.

2.2 Max Method

In this section a drawback of computing stability is introduced and an alternative approach is suggested which is named Max method. Fig. 3 shows two primary partitions for which the stability of each cluster is evaluated. In this example K-means is applied as the base clustering algorithm with K=3. For this example the number of all partitions in the reference set is 40. In 36 partitions the result is relatively similar to Fig 3a, but there are four partitions in which the top left cluster is divided into two clusters, as shown in Fig 3b. Fig 3a shows a true clustering. Since the well separated cluster in the top left corner is repeated several times (90% repetition) in partitions of the reference set, it has to acquire a great stability value (but not equal to 1), however it acquires the stability value of 1. Because the two clusters in right hand of Fig 3a are relatively joined and sometimes they are not recognized in the reference set as well, they have less stability value. Fig. 3.b shows a spurious clustering which the two right clusters are incorrectly merged. Since a fixed number of clusters are forced in the base algorithm, the top left cluster is divided into two clusters. Here the drawback of the stability measure is apparent rarely. Although it is obvious that this partition and the corresponding large cluster on the right reference set (10% repetition), the stability of this cluster is evaluated equal to 1. Since the NMI is a symmetric equation, the stability of the top left cluster in Fig 3.a is exactly equal to the large right cluster in Fig 3.b; however they are repeated 90% and 10%, respectively. In other words, when two clusters are complements of each other, their stabilities are always equal. This drawback is seen when the number of positive clusters in the considered partition of reference set is greater than 1. It means when the cluster C* is obtained by merging two or more clusters, undesirable stability effects occur.

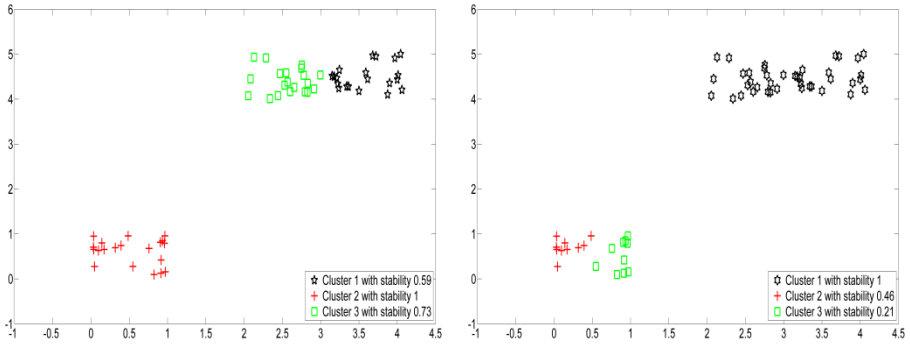


Fig. 3. Two primary partitions with k=3. (a, Left) True clustering. (b, Right) Spurious clustering

To solve this problem we allow only one cluster in reference set to be considered as the *C** (i.e. only the most similar cluster) and all others are considered as *D/C**. In this method the problem is solved by eliminating the merged clusters.

2.3 Consensus Function

One way is to consider the selected clusters as inputs of the HGPA, MCLA and CSPA algorithms [17]. The output of the mentioned algorithms is the final partition which is also called consensus partition.

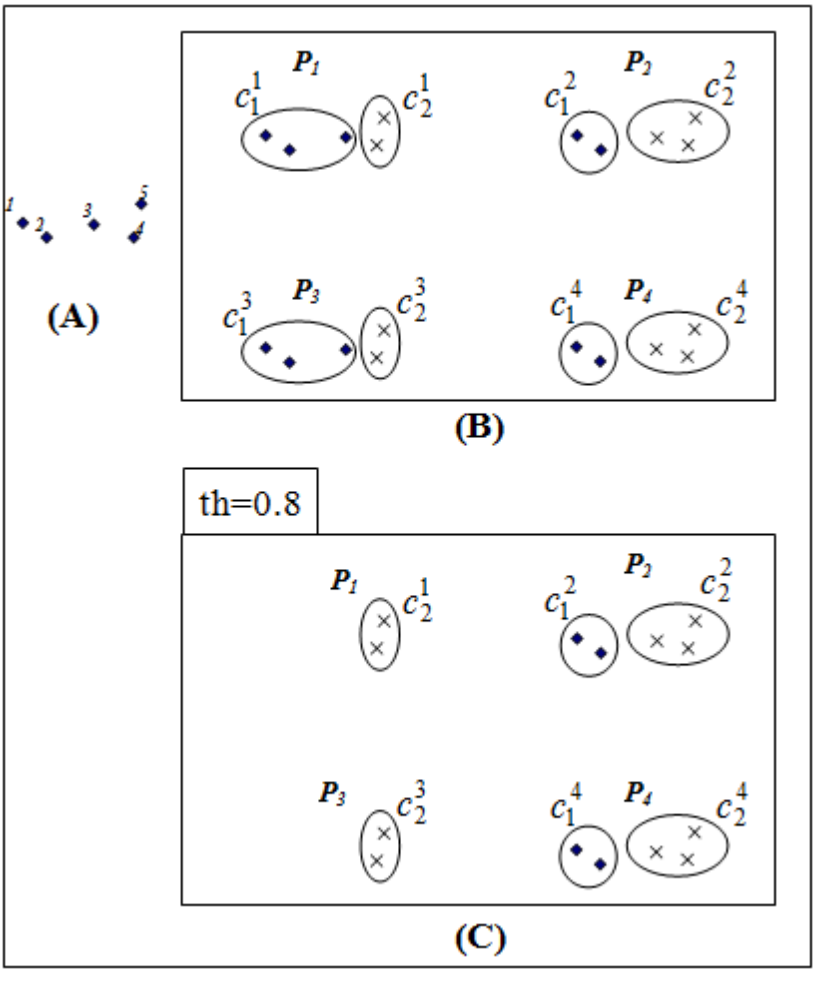


Fig. 4. Computing the co-association matrix by the EEAC method. (A) Data samples. (B) 4 primary clusterings. (C) Remaining clusters after applying threshold, th=0.8.

For the second way to extract the final partition from the selected clusters, the clusters are considered as new space for data, and a clustering algorithm, like fuzzy k-means, is employed to partition the mapped data.

Another alternative way to reach the consensus partition is to use the co-association based methods. In this method, the selected clusters are first used to construct the co-association matrix. In the EAC method the m primary results from

resampled data are accumulated in an $n \times n$ co-association matrix. Each entry in this matrix is computed from equation (3).

$$C(i,j) = \frac{n_{i,j}}{m_{i,j}} \tag{3}$$

where n_{ij} counts the number of clusters shared by objects with indices i and j in the partitions over the B clusterings. Also m_{ij} is the number of partitions where this pair of objects is simultaneously present. There are only a fraction of all primary clusters available, after thresholding. So, the common EAC method cannot truly recognize the pairwise similarity for computing the co-association matrix. In our novel method (Extended Evidence Accumulation Clustering, or EEAC) each entry of the co-association matrix is computed by:

$$C(i,j) = \frac{n_{i,j}}{\max(n_i, n_j)} \tag{4}$$

where n_i and n_j are the number present in remaining (after stability thresholding) clusters for the i-th and j-th data points, respectively. Also, n_{ij} counts the number of remaining clusters which are shared by both data points indexed by i and j, respectively. To further explain, consider this example. Assume that we have five samples (Fig 4a), and that four primary clustering are applied (Fig 4b).

Also, suppose that that stability of the clusters of Fig 4b is as given bellow:

$$Stability(c_2^1) = Stability(c_2^3) = 1$$
$$Stability(c_1^2) = Stability(c_1^4) = 1$$
$$Stability(c_2^2) = Stability(c_2^4) = 0.82$$
$$Stability(c_1^1) = Stability(c_1^3) = 0.55$$

By choosing th=0.8 the first clusters from P1 and P3 are deleted (Fig 4c). According to equation (4), each entry of the co-association matrix is:

$$C(1,2) = \frac{2}{\max(2,2)} = \frac{2}{2} = 1 \qquad C(1,3) = C(2,3) = \frac{0}{\max(2,2)} = \frac{0}{2} = 0$$
$$C(3,4) = C(3,5) = \frac{2}{\max(2,4)} = \frac{2}{4} = 0.5 \quad C(4,5) = \frac{4}{\max(4,4)} = \frac{4}{4} = 1$$

In Fig 4a-c, the data points may be "tracked" by their geometrical arrangement. Example: in computing C(3,4), note that points 3 and 4 both are in cluster 2 of partitions P2 and P4, so that numerator n_{34}=2; also note that n_3=2, since point 3 is only in cluster 2 of P2 and P4, but n_4=4 since point 4 is not only in these clusters, but also in cluster 2 of P1 and P3. Before and after applying threshold, the co-association matrix is given by equation (5) and (6), respectively:

$$C_{before} = \begin{bmatrix} 1 & 1 & 0.5 & 0 & 0 \\ 1 & 1 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 1 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 1 & 1 \\ 0 & 0 & 0.5 & 1 & 1 \end{bmatrix} \tag{5}$$

In this matrix the 3rd object can be considered as both clusters with an equal probability of 50%. The stability measure adds some information to this matrix by applying the threshold.

$$C_{after} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 1 & 1 \\ 0 & 0 & 0.5 & 1 & 1 \end{bmatrix} \tag{6}$$

By comparing these two matrices and also considering the stability values, it can be seen that deletion of unstable clusters improves the co-association matrix. By eliminating the unstable cluster with samples {1, 2, 3} which is spuriously created by primary clusterings.

After computing the co-association matrix by the EEAC method, a consensus function is employed to extract the final clusters from the matrix. Here, the single-link method is used for this task.

3 Experimental Study

Evaluation metric based on which a consensus partition is evaluated is discussed in the first subsection of this section. The details of the used datasets are given in the subsequent section. Then the settings of experimentations are given. Finally the experimental results are presented.

Table 1. Brief information about the used datasets

	Dataset Name	*# of Class*	*# of Features*	*# of Samples*
1	Breast-Cancer*	2	9	683
2	Iris*	3	4	150
3	Bupa*	2	6	345
4	SAHeart*	2	9	462
5	Ionosphere	2	34	351
6	Glass*	6	9	214
7	Halfrings	2	2	400
8	Galaxy*	7	4	323
9	Yeast*	10	8	1484
10	Wine	3	13	178

3.1 Evaluation Metric

After producing the consensus partition, the most important question is "how good a partition is?". The evaluation of a partition is very important as it is mentioned. Here the NMI between the consensus partition and real labels of the dataset is considered as an evaluation metric of the consensus partition. Also accuracy between the consensus partition and real labels of the dataset is considered as another metric.

3.2 Datasets

The proposed method is examined over 9 different standard datasets and one artificial dataset. It is tried for datasets to be diverse in their number of true classes, features and samples. A large variety in used datasets can more validate the obtained results. Brief information about the used datasets is available in Table 1. More information is available in [14].

Note that some of datasets which are marked with star (*) in Table 1 are normalized. All experiments are done over the normalized features in the stared dataset. It means each feature is normalized with mean of 0 and variance of 1, N(0, 1). The artificial Half Ring dataset is depicted in the Fig. 5.

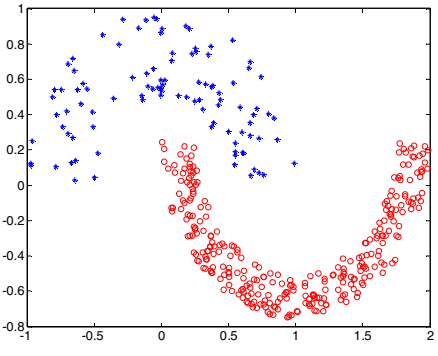


Fig. 5. Half Ring dataset

3.3 Experimental Settings

To be more general and fair, all experiments are averaged over 10 independent runs. In all experimentations there are 120 independent partitions obtained by 120 independent runs of k-means clustering algorithm with different initialized seed points and different k parameter, ranging from k to 2*k.

After selecting a subset of clusters, to extract the final partition from them, the real number of clusters, i.e. the column three of the Table 1, is served by the consensus functions.

As it is known in fuzzy k-means clustering algorithm, each data point belongs to all clusters with different membership values. To extract the final partition from output of fuzzy k-means algorithm as consensus function, each data point is assigned to the most membership value.

3.4 Experimental Results

As it is inferred from the Fig. 6, the best ratio of selection of the stable clusters is 60% and the best option for consensus function is CSPA for Iris dataset.

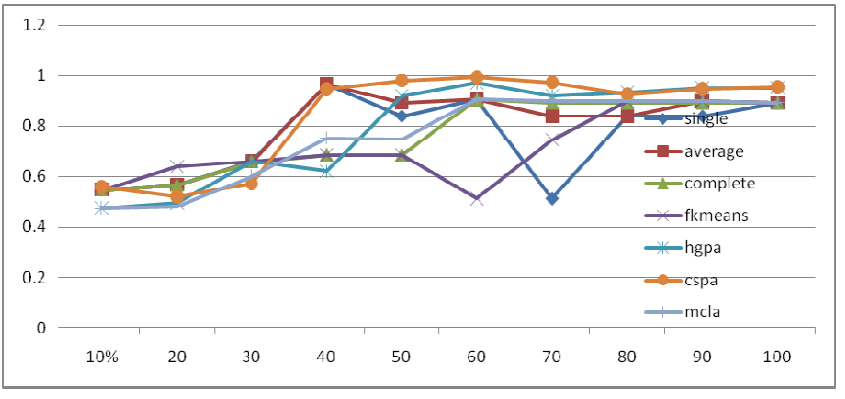


Fig. 6. The horizontal axis stands for the rate of stable clusters that are selected. The vertical axis stands for the NMI values between the labels of Iris dataset and the consensus partitions obtained by different consensus functions over the selected clusters.

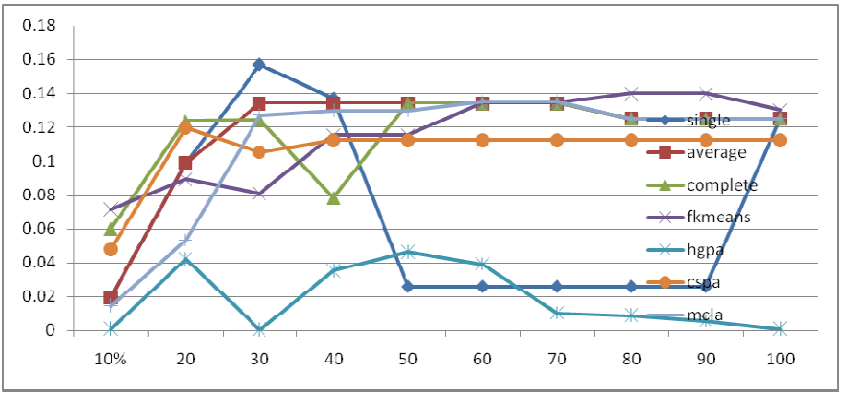


Fig. 7. The horizontal axis stands for the rate of stable clusters that are selected. The vertical axis stands for the NMI values between the labels of Ionosphere dataset and the consensus partitions obtained by different consensus functions over the selected clusters.

Fig. 7 makes it clear that the best ratio of selection of the stable clusters is 30% and the best option for consensus function is Single-Linkage for Ionosphere dataset.

To see whether the use of a subset of the most stable clusters can affect the quality of the final cluster or not, consider Fig. 8. To make a general decisive conclusion, the results for all ten datasets of Table 1 are averaged and the final results are illustrated in the Fig. 8. The Averaged-Linkage consensus function over 50% of the most stable clusters generally reaches the maximum for all dataset.

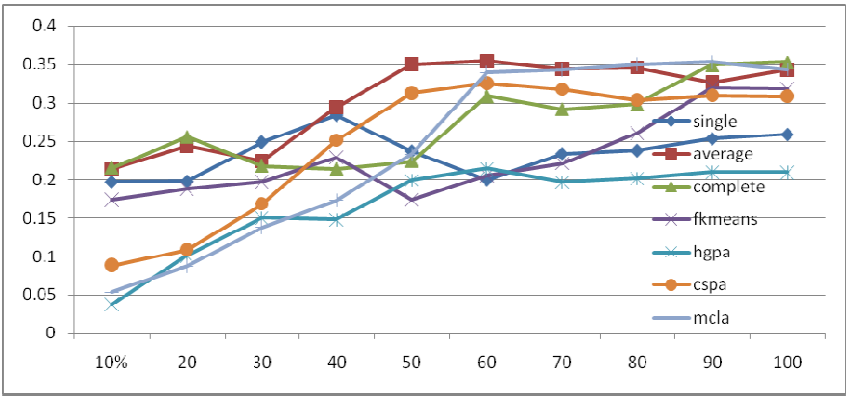


Fig. 8. The horizontal axis stands for the rate of stable clusters that are selected. The vertical axis stands for the averaged NMI values for all ten datasets of Table 1.

Table 2 shows the performance of the proposed method comparing with most common base and ensemble methods.

Table 2. Experimental results

	Simple Methods (%)				Ensemble Methods (%)			
Dataset	Single Linkage	Average Linkage	Complete Linkage	Kmeans	Kmeans Ensemble	Full Ensemble	Cluster Selection by NMI Method	Cluster Selection by Max Method
Wine	37.64	38.76	83.71	96.63	96.63	97.08	**97.75**	97.44
BreastC	65.15	70.13	94.73	95.37	95.46	95.10	95.75	**96.49**
Yeast	34.38	35.11	38.91	40.20	45.46	47.17	47.17	**51.27**
Glass	36.45	37.85	40.65	45.28	47.01	47.83	**48.13**	47.35
Bupa	57.68	57.10	55.94	54.64	54.49	55.83	58.09	**58.40**

4 Conclusion and Future Works

In this paper a new clustering ensemble framework is proposed which is based on a subset of total primary spurious clusters. Also a new alternative method for common NMI is suggested. Since the quality of the primary clusters are not equal and presence of some of them can even yield to lower performance, here a method to select a subset of more effective clusters is proposed. A common cluster validity criterion which is needed to derive this subset is based on normalized mutual information. In this paper some drawbacks of this criterion is discussed and an method is suggested which is called max mehod. The experiments show that the proposed framework commonly outperforms in comparison with the full ensemble; however it uses just 50% of primary clusters. Another innovation of this chapter is a method for constructing the co-association matrix where some of clusters and respectively some of samples do not exist in partitions. This new method is called Extended Evidence Accumulation Clustering, EEAC.

References

1. Ayad, H., Kamel, M.S.: Cumulative Voting Consensus Method for Partitions with a Variable Number of Clusters. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(1), 160–173 (2008)
2. Bhatia, S.K., Deogun, J.S.: Conceptual Clustering in Information Retrieval. IEEE Trans. Systems, Man, and Cybernetics 28(3), 427–536 (1998)
3. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics 19(9), 1090–1099 (2003)
4. Faceli, K., de Carvalho, A.C.P.L.F., de Souto, M.C.P.: Multi-objective Clustering Ensemble. In: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems, HIS 2006 (2006)
5. Fred, A., Jain, A.K.: Data Clustering Using Evidence Accumulation. In: Proc. of the 16th Intl. Conf. on Pattern Recognition, ICPR 2002, Quebec City, pp. 276–280 (2002)
6. Fred, A., Jain, A.K.: Combining Multiple Clusterings Using Evidence Accumulation. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(6), 835–850 (2005)
7. Fred, A., Jain, A.K.: Learning Pairwise Similarity for Data Clustering. In: Proc. of the 18th Int. Conf. on Pattern Recognition (ICPR 2006) (2006)
8. Fred, A., Lourenco, A.: Cluster Ensemble Methods: from Single Clusterings to Combined Solutions. In: Okun, O., Valentini, G. (eds.) Supervised and Unsupervised Ensemble Methods and their Applications. SCI, vol. 126, pp. 3–30. Springer, Heidelberg (2008)
9. Frigui, H., Krishnapuram, R.: A Robust Competitive Clustering Algorithm with Applications in Computer Vision. IEEE Trans. Pattern Analysis and Machine Intelligence 21(5), 450–466 (1999)
10. Jain, A.K., Murty, M.N., Flynn, P.: Data clustering: A review. ACM Computing Surveys 31(3), 264–323 (1999)
11. Judd, D., Mckinley, P., Jain, A.K.: Large-Scale Parallel Data Clustering. IEEE Trans. Pattern Analysis and Machine Intelligence 19(2), 153–158 (1997)
12. Lange, T., Braun, M.L., Roth, V., Buhmann, J.M.: Stability-based model selection. In: Advances in Neural Information Processing Systems, vol. 15. MIT Press (2003)
13. Law, M.H.C., Topchy, A.P., Jain, A.K.: Multiobjective data clustering. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Washington, D.C, vol. 2, pp. 424–430 (2004)
14. Newman, C.B.D.J., Hettich, S., Merz, C.: UCI repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLSummary.html
15. Parvin, H., Minaei-Bidgoli, B., Alinejad, H.: Linkage Learning Based on Differences in Local Optimums of Building Blocks with One Optima. International Journal of the Physical Sciences, IJPS, 3419–3425 (2011)
16. Roth, V., Lange, T., Braun, M., Buhmann, J.: A Resampling Approach to Cluster Validation. In: Intl. Conf. on Computational Statistics, COMPSTAT (2002)
17. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)

Nearest Cluster Classifier

Hamid Parvin, Moslem Mohamadi, Sajad Parvin, Zahra Rezaei, and Behrouz Minaei

Nourabad Mamasani Branch, Islamic Azad University, Nourabad Mamasani, Iran
hamidparvin@mamasaniiau.ac.ir,
{mohamadi,s.parvin, rezaei,b_minaei}@iust.ac.ir

Abstract. In this paper, a new classification method that uses a clustering method to reduce the train set of K-Nearest Neighbor (KNN) classifier and also in order to enhance its performance is proposed. The proposed method is called Nearest Cluster Classifier (NCC). Inspiring the traditional K-NN algorithm, the main idea is to classify a test sample according to the tag of its nearest neighbor. First, the train set is clustered into a number of partitions. By obtaining a number of partitions employing several runnings of a simple clustering algorithm, NCC algorithm extracts a large number of clusters out of the partitions. Then, the label of each cluster center produced in the previous step is determined employing the majority vote mechanism between the class labels of the patterns in the cluster. The NCC algorithm iteratively adds a cluster to a pool of the selected clusters that are considered as the train set of the final 1-NN classifier as long as the 1-NN classifier performance over a set of patterns included the train set and the validation set improves. The selected set of the most accurate clusters are considered as the train set of final 1-NN classifier. After that, the class label of a new test sample is determined according to the class label of the nearest cluster center. Computationally, the NCC is about K times faster than KNN. The proposed method is evaluated on some real datasets from UCI repository. Empirical studies show an excellent improvement in terms of both accuracy and time complexity in comparison with KNN classifier.

Keywords: Nearest Cluster Classifier, K-Nearest Neighbor, Combinational Classification.

1 Introduction

KNN classifier is one of the most fundamental classifiers. It is also the simplest classifier. It could be the first choice for a classification study when there is little or no prior knowledge about the data distribution. The KNN classifies a test sample x by assigning it the label most frequently represented among the K nearest samples; in other words, a decision is made by examining the labels on the K-nearest neighbors and taking a majority vote mechanism. KNN classifier was developed from the need to perform discriminant analysis when reliable parametric estimates of the probability densities are unknown or difficult to determine. In 1951, Fix and Hodges introduced a

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 267–275, 2012.

non-parametric method for pattern classification that has since become known the *K*-nearest neighbor rule [1] and [15]. Later in 1967, some of the formal properties of the *K*-nearest neighbor rule have been worked out; for instance it was shown that for *K*=1 and $n \to \infty$ the KNN classification error is bounded above by twice the Bayes error rate [2]. Once such formal properties of KNN classification were established, a long line of investigation ensued including new rejection approaches [3], refinements with respect to Bayes error rate [4], distance weighted approaches [5-6], soft computing [7] methods and fuzzy methods [8-9].

Some advantages of KNN include: its simplicity to use, its robustness to learn in a noisy training data (especially if it uses the inverse square of weighted distances as the "distance metric"), and finally its effectiveness in learning at a large scale training dataset. Although KNN has the mentioned advantages, it has some disadvantages such as: its high computation cost (because it needs to compute distance of each query instance to all training samples); its need to a large memory (in proportion with the size of training set); its low performance in multidimensional data sets; its sensitivity to well-setting of parameter *K* (number of nearest neighbors); its sensitivity to the used distance metric; and finally there is no solution for the weighting consideration of the features [10].

The computational complexity of the nearest neighbor algorithm, in both space (storage of prototypes) and time (distance computation) has received a great deal of analysis. Suppose we have *N* labeled training samples in *d* dimensions, and seek to find the closest to a test point *x* (*K* = 1). In the most naive approaches we inspect each stored point in turn, calculate its Euclidean distance to x, retaining the identity only of the current closest one. Each distance calculation is O(*d*), and thus this search is O(dN^2) [10].

ITQON et al. in [11] proposed a classifier, TFkNN, aiming at upgrading of distinction performance of KNN classifier and combining plural KNNs using testing characteristics. Their method not only upgrades distinction performance of the KNN but also brings an effect stabilizing variation of recognition ratio; and on recognition time, even when plural KNN classifiers are performed in parallel, by devising its distance calculation it can be done not so as to extremely increase on comparison with that in single KNN classifier.

In this paper a new approach that is based on the idea of KNN classifier is proposed. It can augment the performance of KNN classifier in terms of the accuracy, the time complexity and the memory complexity. This method which is called that named Nearest Cluster Classifier (NCC), applies the clustering techniques to reduce the number of training prototypes. Despite of reducing train samples, the clustering will cause to find the natural groups of data.

The rest of the paper is organized as follows. Section 2 expresses the proposed NCC algorithm. Experimental results are addressed in section 3. Finally, section 4 concludes the paper.

2 NCC: Nearest Cluster Classifier

In all experimentations the 10-fold cross validation is employed to obtain the performance of the NCC algorithm. In 10-fold cross validation the dataset is

randomly partitioned into 10 clusters. Considering each partition as test set, *PTeS*, and the other data as train set, *PTrS*, the NCC algorithm reaches 10 experimentations. Averaging the performances of the NCC algorithm over all 10 test sets, the final performances of the NCC algorithm is obtained. In each experimentation, the train set, *PTrS*, is divided into two sub-partitions, train sub-set, *TS*, and evaluation (validation) sub-set, *VS*. The main idea of the proposed method is assigning the data to the nearest cluster who is naturally consisted the neighbor points. To implement the idea, first, the samples of the train sub-set, *TS*, are clustered into *k* clusters where *k* is a number equal to or greater than the number of real classes, *c*, and equal to or less than 2**c*. The clustering is done using a simple rough clustering algorithm, *clus_alg*. Then, considering the obtained cluster centers as new points, their labels are determined using the simple majority vote mechanism. Indeed the label of a cluster is obtained based on a KNN classifier over train set where *K* is equal to an input parameter, pq. The *condition* is computed as equation 1.

$$condition(cvp, pq) = \begin{cases} cvp \geq pq \\ cvp \geq \dfrac{pq}{2} \\ cvp \leq \dfrac{pq}{2} \end{cases} \quad (1)$$

For example, if $condition(cvp, pq) = cvp > \frac{pq}{2}$; it means that each cluster center that is labeled with less than $\frac{pq}{2}$ votes in KNN classifier (using *PTrS* as train set) is immediately omitted; otherwise it is added in a pool of clusters, *PC*. This procedure is iterated as many as *Maxiteration*. So finally the NCC algorithm has a pool of clusters with $Maxiteration \times c^3$ ($Maxiteration \times k \times k \times \frac{k}{2}$) clusters at most.

The quality of obtained clusters is evaluated employing a 1-NN with considering *PTrS* as test set. Each pattern of the *PTrS* is used as a test sample; determining the nearest cluster, its label is assigned to the sample. After that, in comparison with the ground true labels of data, the accuracy of the obtained classifier is derived. So, after obtaining the pool of clusters, *PC*, the NCC algorithm iteratively selects them into a subset of pool of clusters (*SPC*) if the accuracy of a 1-NN using *SPS* as train set over *PTrS* is improved.

In test phase of the NCC algorithm, a new test sample is assigned to the label of the nearest cluster center. The pseudo code of training phase of the Nearest Cluster Classifier algorithm is shown in Fig. 1. Until here, the training of the NCC is finished. After here, any test samples are classified using the trained classifier.

In the rest of this section the proposed method is described in detail, answering the questions, how to cluster the train set, how to determine the class labels of cluster centers and how to find the final classifier to classify a test sample.

2.1 Determining the Label of Cluster Centers

In this section, we explain how the class labels are used to specify the labels of the cluster centers which are explanatory points of the clusters. There are some combining methods to aggregate the class labels of the cluster members. When the

individual votes of classifiers are crisp (not soft/fuzzy), the Simple Majority Vote approach is the common logical one that votes a pattern x to a class j if a little more than n/c of classifiers (here cluster members) assigns to class j [12], where n and c stand for the number of cluster members and the number of classes, respectively. In the paper, the majority vote mechanism is used to assign a class label to cluster centers. It means that pq nearest neighbors of a cluster center in the *PTrS* make a consensus decision about its label.

Input:
PTrS: patterns of train set
PTeS: patterns of test set
c: the number of classes
clus_alg: clustering algorithm
pq: the threshold for assigning a label to a cluster center
Maxiteration: the maximum of the allowed Iterations
Condition: a condition for decision
Output:
accuracy: accuracy of NCC over *PTeS*
PC={}
partition *PTrS* into two clusters: *TS* and *VS*
For i = 1 to *Maxiteration*
1. *P*=*clus_alg*(*TS*,*k*) where $2*c \geq k \geq c$; resulting cluster centers P_i
2. For each $p \in P$
if *condition*(*cvr, pq*), then p_i will be added to the set *PC;* where *cvr* is the maximum number of the *pq* nearest patterns of P_i in *PTrS* that have consensus vote to an identical label
SPC={}
cur_acc=0
For each $q \in PC$
1. $TTS = SPC \cup q$
2. *acc*=1-NN(*TTS*, *PTrS*)
3. if(*acc*>*cur_acc*) *SPC*=*TTS*
acc=1-NN(*TTS*, *PTeS*)

Fig. 1. Pseudo-code of training phase of the nearest cluster classifier algorithm

2.2 Evaluation of the Cluster Centers

There are many methods to evaluate the clustering result. They may use whether external indices, whether internal indices or relative indices [13]. External index needs further information to evaluate the clusters. In the paper, the *PTrS* is used to

measure the performance of the different clusterings. It is a kind of external index usage. First, the NCC algorithm is trained on the *TS*. Then, by executing the trained classifier on the *PTrS*, the accuracy of this method is obtained using the ground true class labels of the *PTrS*.

2.3 Final Classifier

As it is shown in Fig. 1, the steps 1, 2 and 3 are repeated *Maxiteration* times. In this method there is a procedure to select a set of satisfactory good cluster centers from several times of performing clustering techniques; however, the cluster centers obtained from any iteration can be considered as the solution. The method enhances both the accuracy and robustness of the KNN classifier algorithm, significantly; however, it needs less time and memory in testing phase. Based on empirical study, it can be induced that, usually the best results may be obtained when the *SCP* size is chosen near to the value of number of classes, c. Since each cluster center has d dimensions, examining each test sample needs to O(cd). In the worst case the time complexity is O(c^3d). It shows that the proposed combinational method can be employed with less order than the KNN classifier method which is O(dkN).

Table 1. Brief information about the used datasets

	Dataset Name	*# of Class*	*# of Features*	*# of Samples*
1	Breast-Cancer*	2	9	683
2	Iris*	3	4	150
3	Bupa*	2	6	345
4	SAHeart*	2	9	462
5	Ionosphere	2	34	351
6	Glass*	6	9	214
7	Halfrings	2	2	400
8	Galaxy*	7	4	323
9	Yeast*	10	8	1484
10	Wine	3	13	178

3 Experimental Study

This section discuses the experimental results and compares the performance of the NCC algorithm with original KNN methods.

3.1 Datasets

The proposed method is examined over 9 different standard datasets and one artificial dataset. It is tried for the used datasets to be diverse in their number of true classes, features and samples. A large variety in used datasets can more validate the obtained

results. Brief information about the used datasets is available in Table 1. More information is available in [14]

Note that datasets which are marked with star (*) in paper are normalized. The experiments are done over the normalized features in the stared dataset. It means each feature is normalized with the mean of 0 and the variance of 1, N(0, 1). The artificial HalfRing dataset is depicted in Fig. 2. The HalfRing dataset is considered as one of the most challenging dataset for the proposed NCC algorithms.

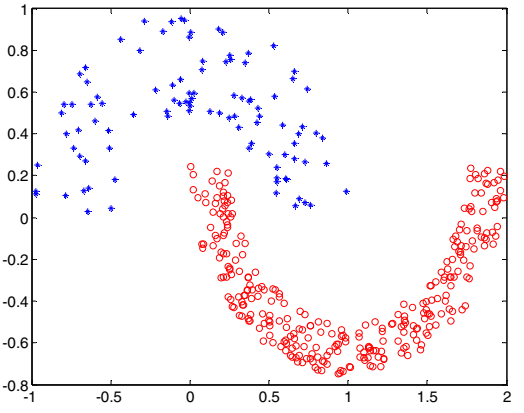


Fig. 2. Half Ring dataset

3.2 Experimental Settings

All experiments are reported on 10-fold cross validation. Parameter *pq* is set to 5 through all experimentations. In all experiments Parameter *Maxiteration* is 10. Parameter *clus_alg* is k-means. It means that the k-means clustering algorithm is considered as clustering algorithm. The maximum allowed iterations in the k-means clustering algorithm is equal to 2 in order to obtain the rough and unlike partitions out of data. The number of partitions that is requested from k-means is a random value between *c* and 2**c*. Validation set, *VS*, is 22.22% of train set, *PTrS*, through all experimentations.

3.3 Experimental Results

Table 2 shows final accuracies of the NCC algorithm using three different conditions. In each column, the accuracies obtained by the NCC algorithm employing one condition are shown. Next to the accuracy of each NCC algorithm over each dataset, the averaged number of cluster centers in the final 1-NN classifier is presented. It is obvious that the condition $cvp \geq \frac{pq}{2}$ is the best condition among the three used conditions. But there is a hidden rule among the results obtained by employing the condition $cvp \geq \frac{pq}{2}$ in the NCC algorithm and the results obtained by employing the condition $cvp \geq pq$ in the NCC algorithm. First we define a new column in Table 2.

It is the ratio of column 3 to column 5. We present this defined column in last column of Table 2. By a detailed considering of the values in the last column of Table 2, it is inferred when the last column for a dataset is lower than the averaged last column over all datasets (depicted at last column and last row in Table 2, it is equal to 1.58), the NCC algorithm with the condition $cvp \geq pq$ is superior to the others; otherwise the NCC algorithm with the condition $cvp \geq \frac{pq}{2}$ is superior to the others.

Table 2. Final results of the NCC algorithm using different conditions

	$cvp \geq \frac{pq}{2}$		$cvp \geq pq$		$cvp \leq \frac{pq}{2}$			
	NCC	Average # of Prototypes	NCC	Average # of Prototypes	NCC	Average # of Prototypes	All Prototypes	Ratio of Column 3 to 5
1	**97.25**	6.90	96.25	6.20	96.25	9.00	280	1.11
2	95.33	8.60	**96.00**	7.30	**96.00**	7.00	105	1.18
3	96.47	6.20	**97.06**	5.10	95.88	5.00	479	1.22
4	**60.29**	10.40	52.53	2.50	58.24	10.00	242	4.16
5	**71.56**	52.00	67.50	28.90	69.69	141.00	227	1.80
6	**66.67**	28.90	52.86	15.00	65.71	98.00	151	1.93
7	82.86	9.90	**85.71**	6.90	84.00	8.00	246	1.43
8	**69.78**	10.10	65.65	1.60	68.26	9.00	324	6.31
9	95.29	9.20	95.29	7.40	**95.88**	8.00	126	1.24
10	**57.70**	92.40	57.53	67.31	54.53	480.00	1040	1.37
average	**79.32**	23.46	76.64	14.82	78.44	77.50	322	1.58

The hidden rule says when the number of prototypes in the NCC with the condition $cvp \geq pq$ is less than the number of prototypes in the NCC with the condition $cvp \geq \frac{pq}{2}$ by large margin, the dataset is a hard one. So we must turn to the NCC with the best prototypes. It means that the NCC with the condition $cvp \geq pq$ is the best option. It also says when the number of prototypes in the NCC with the condition $cvp \geq pq$ is less than the number of prototypes in the NCC with the condition $cvp \geq \frac{pq}{2}$ by small margin, the dataset is an easy one. So we can turn to the NCC with the more prototypes to cover total feature space. It means that the NCC with the condition $cvp \geq \frac{pq}{2}$ is the best option.

By using the hidden rule, we can present a combinatorial selective classifier that contains both NCCs and uses each of them dependent on the defined ratio for the two classifiers.

Table 3. Final accuracies of the NCC comparing with the results of different KNN classifiers

	Different Method								
	1-NN	2-NN	3-NN	4-NN	5-NN	6-NN	7-NN	NCC	Average # of Prototypes
1	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	96.25	6.20
2	96.00	95.33	96.67	97.33	97.33	**98.00**	**98.00**	96.00	7.30
3	96.91	94.71	97.21	96.76	**97.50**	96.18	96.62	97.06	5.10
4	63.53	60.00	**63.82**	62.06	60.59	55.59	58.53	60.29	10.40
5	81.88	**82.81**	80.63	77.51	75.94	71.25	70.94	71.56	52.00
6	69.05	68.57	**69.52**	63.80	64.29	63.33	62.86	66.67	28.90
7	86.57	**89.43**	84.00	86.57	84.86	85.71	83.43	85.71	6.90
8	65.65	66.30	68.70	69.13	65.43	67.17	66.52	**69.78**	10.10
9	95.29	94.12	94.71	94.71	**96.47**	94.71	96.47	95.29	7.40
10	52.77	51.55	55.47	55.61	**57.91**	57.57	57.03	57.53	67.31
average	80.77	80.28	**81.07**	80.35	80.03	78.95	79.04	79.61	20.16

Table 3 shows the performances of different KNN classification and the proposed combinatorial selective classifier comparatively. NCC is compared with original versions of KNN. The NCC method outperforms some of KNN classifiers in terms of average accuracy. In addition, because of the lower number of stored prototypes, the results of the proposed combinatorial selective classifier are gained while the testing phase of the NCC method has less computational burden in both cases of time and memory rather than the KNN classifiers.

4 Conclusion and Future Works

In this paper, a new method is proposed to improve the performance of KNN classifier. The proposed method which is called NCC, standing for Nearest Cluster Classifier, improves the KNN classifier in terms of both time and memory order. The NCC algorithm employs clustering technique to find the same groups of data in multi-dimensional feature space.

Despite of reducing training prototypes, the clustering technique can cause to find the natural groups of data. On the other hands, the natural neighborhoods can be successfully recognized by clustering technique. Moreover, unlike the KNN method which classifies any sample without considering the data distribution, only based on exactly K nearest neighbor, in the NCC algorithm, the data is grouped into k clusters unequally, according to the data distribution and the position of data samples in feature space.

The NCC method is examined over nine benchmarks from UCI repository and one hand made dataset, HalfRing. Regarding to the obtained results, it can be concluded that the proposed algorithm is comparatively not worse than the KNN classifier. The NCC method is even more accurate than the KNN classifier in some cases.

References

1. Fix, E., Hodges, J.L.: Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas (1951)
2. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. Inform. Theory IT-13(1), 21–27 (1967)
3. Hellman, M.E.: The nearest neighbor classification rule with a reject option. IEEE Trans. Syst. Man Cybern. 3, 179–185 (1970)
4. Fukunaga, K., Hostetler, L.: k-nearest-neighbor bayes-risk estimation. IEEE Trans. Information Theory 21(3), 285–293 (1975)
5. Dudani, S.A.: The distance-weighted k-nearest-neighbor rule. IEEE Trans. Syst. Man Cybern. SMC-6, 325–327 (1976)
6. Bailey, T., Jain, A.: A note on distance-weighted k-nearest neighbor rules. IEEE Trans. Systems, Man, Cybernetics 8, 311–313 (1978)
7. Bermejo, S., Cabestany, J.: Adaptive soft k-nearest-neighbour classifiers. Pattern Recognition 33, 1999–2005 (2000)
8. Jozwik, A.: A learning scheme for a fuzzy k-nn rule. Pattern Recognition Letters 1, 287–289 (1983)
9. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nn neighbor algorithm. IEEE Trans. Syst. Man Cybern. SMC-15(4), 580–585 (1985)
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons (2000)
11. Itqon, S.K., Satoru, I.: Improving Performance of k-Nearest Neighbor Classifier by Test Features. Springer Transactions of the Institute of Electronics, Information and Communication Engineers (2001)
12. Lam, L., Suen, C.Y.: Application of majority voting to pattern recognition: An analysis of its behavior and performance. IEEE Transactions on Systems, Man, and Cybernetics 27(5), 553–568 (1997)
13. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
14. Newman, C.B.D.J., Hettich, S., Merz, C.: UCI repository of machine learning databases (1998), `http://www.ics.uci.edu/~mlearn/MLSummary.html`
15. Wu, X.: Top 10 algorithms in data mining. In: Knowledge Information, pp. 22–24. Springer-Verlag London Limited (2007)

Diffusion Maps for the Description of Meteorological Data

Ángela Fernández, Ana M. González, Julia Díaz, and José R. Dorronsoro

Universidad Autónoma de Madrid and Instituto de Ingeniería del Conocimiento
Francisco Tomás y Valiente 11, 28049, Madrid, Spain
{a.fernandez,ana.marcos,julia.diaz,jose.dorronsoro}@uam.es

Abstract. Diffusion Maps is a new powerful technique for dimensionality reduction that can capture geometric structure while taking into account data distribution. In this work we will apply it to time and spatial compression of numerical weather forecasts, showing how it is capable to greatly reduce the initial dimension while still capturing relevant information in the original data.

Keywords: Natural Clustering, Compressed Data, Diffusion Maps.

1 Introduction

Methods for the compression and subsequent analysis of data that also preserve the original information are much valued in data mining and in machine learning. Principal Component Analysis (PCA) [4], spectral dimensionality reduction (DR) [1] or, in general, Diffusion Maps (DMs) [5], [6] are mathematical techniques for unsupervised dimensionality reduction; furthermore, DMs dimensionality reduction lend itself naturally to the subsequent application of clustering methods. More precisely, Diffusion Maps embed the original space into a new smaller one while preserving the original geometry. In addition, there is a natural diffusion metric in the original feature space that corresponds with Euclidean metric in the embedded space. This means that clustering methods that rely on Euclidean metrics, as is the case of k–means, can be applied in the reduced space. In other words, DMs can be seen as an intelligent data analysis technique (for other examples of these techniques, see [13]) for both dimensionality reduction and clustering.

The analysis of meteorological data is a field that can clearly benefit from dimensionality reduction techniques. Considering weather forecasts, they usually involve a large number of variables (about 70 for the forecasts of the European Center for Medium–Range Weather Forecasts–ECMWF [7]) over large spatial grids. Even if we consider a single grid point, we must add to this the fact that forecasts are given for several pressure layers (about 25 for ECMWF) and for several future horizons (75 in the ECMWF case). Of course, what data are to be used largely depends on the problem at hand but, as it will be the case here, two usual situations are, first, to consider a set of forecasts at a single grid point but

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 276–287, 2012.

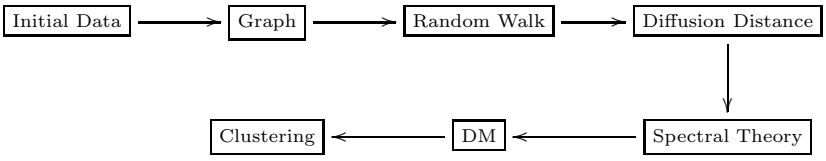


Fig. 1. Steps for Diffusion Maps

taken over a long time period or, second, to consider such a set over a large grid for, say, a given day. When dimension reduction methods are applied to them, we shall talk about time and spatial dimensionality reduction respectively.

In both cases, large dimensional data vectors are to be considered and dimensionality reduction is clearly of interest but, as it is generally the case with unsupervised methods, a reasonable question is how to decide the correctness or usefulness of the new representation. A possible answer can be derived through the use of clustering. Consider first data points made up of forecasts at a given grid point over, say, one year. It is logical to expect that points that are close in the reduced space should also be close in the original space. Since we are dealing with surface points, a first notion of closeness could be just geographic proximity but this may be just too narrow, as we would also expect that far away points may share similar weather. Similarly, when dealing with daily forecasts over a large grid, we should expect that points close in the reduced space are also close in time; that is, the reduced representation should somehow reflect seasonal features. We shall follow these approaches in this work. We will work with ECMWF forecasts for a $1,995$–point square grid that contains the Iberian peninsula and has a resolution of 0.25 degrees (i.e., a grid square corresponds to a land square with about a 27 Km side). We will consider data for a whole year and work with five meteorological variables that are forecasted every 3 hours, i.e., 8 times a day. Thus, in time compression we will have a sample of $1,995$ points with dimension $14,600 = 365 \times 5 \times 8$ while in spatial compression, we will have a 365 point sample with dimension $79,800 = 1,995 \times 5 \times 8$. The paper is organized as follows. In Section 2, we shall briefly review Diffusion Maps while in Section 3 we will first review the problems to solve and the data for them and then we will present the results obtained. Finally, Section 4 ends this paper with a brief discussion and conclusions.

2 Diffusion Maps

Dimensionality reduction techniques seek to obtain low dimensional representations of data either looking to capture as much of the variance of the original data as possible or trying to reflect an underlying low dimensional manifold where the original data lay. Principal components is a well known example of the first approach, Spectral Clustering or, more generally, Diffusion Maps (DMs) of the second one. In this section we will briefly review DMs. Its different steps

are depicted schematically in Figure 1. To identify the underlying manifold, the first step in both Spectral Clustering and Diffusion Maps is to build a connectivity graph of the sample in the initial feature space $\mathcal{S} = \{x_1, \ldots, x_n\}$. The graph nodes are the sample points x_i and the weight matrix $W_{ij} = w(x_i, x_j)$ is defined to be symmetric and point–wise positive. The usual choice for the weight matrix is to use Gaussian Kernel defined as $w(x_i, x_j) = \exp\left(-||x_i - x_j||^2/\epsilon^2\right)$, where ϵ determines the size of the neighborhood (see [10] for some directions on how to choose this parameter). Besides other reasons, this kernel choice allows a natural connection to the Riemannian structure of the underlying manifold [1] and, moreover, makes possible to use Nyström formula [2] to compute the diffusion coordinates of new, unseen points.

Once the matrix W is obtained, we expect the graph $G = (\mathcal{S}, W)$ will capture the local geometry of the data. However, the sample we work with depends not only on the geometry of the underlying manifold, but also on the data distribution on the manifold. We review these double influences next, following the discussion in [5]. Of course, sample distribution may not be related to the manifold's geometry. To control the influence of each, we consider a new parameter α between 0 and 1 that will determine the influence of either geometry or density. To begin with, we consider the density of each vertex i as the degree of the graph $q(x_i) = \sum_{j=1}^{n} w(x_i, x_j)$ and normalize the weight $w(x_i, x_j)$ as $w^{(\alpha)}(x_i, x_j) = \frac{w(x_i, x_j)}{q(x_i)^\alpha q(x_j)^\alpha}$. Working with this new matrix, the degree of a vertex x_i becomes now

$$g^{(\alpha)}(x_i) = \sum_{j=1}^{n} \frac{w(x_i, x_j)}{q(x_i)^\alpha q(x_j)^\alpha} = \sum_{j=1}^{n} w^{(\alpha)}(x_i, x_j),$$

and we define a Markov chain on the graph with transition probability $p_{ij}^{\alpha} = p^{\alpha}(x_i, x_j) = \frac{w^{(\alpha)}(x_i, x_j)}{g^{(\alpha)}(x_i)}$. We can consider one–step Markov neighborhoods or, more generally, larger t–step neighborhoods through the powers of P, i.e., $p_t(x_i, x_j) = (P^t)_{ij}$. This leads to define for each t a diffusion distance that considers two points to be close if they are connected in the graph by many short paths of length t. More precisely, if ϕ_0 denotes the stationary distribution of the Markov process [11], defined as $\phi_0(x) = \frac{g^{(\alpha)}(x)}{\sum_{z \in \mathcal{S}} g^{(\alpha)}(z)}$, the square of the diffusion distance in t time steps [9] is defined by

$$D_t^2(x, z) = ||p_{\alpha,t}(x, \cdot) - p_{\alpha,t}(z, \cdot)||^2_{L^2(\frac{1}{\phi_0})} = \sum_{y \in \mathcal{S}} \frac{(p_{\alpha,t}(x, y) - p_{\alpha,t}(z, y))^2}{\phi_0(y)},$$

where $L^2(\frac{1}{\phi_0})$ represents the L^2–Norm weighted by the stationary distribution which takes into account the (empirical) local density of the points. The theory of spectral graphs [1], [12] allows us to consider an alternative formulation of the diffusion distance, namely

$$D_t^2(x, z) = \sum_{j=1}^{n-1} \lambda_j^{2t} (\psi_j(x) - \psi_j(z))^2, \tag{1}$$

where λ_j are the eigenvalues and ψ_j are the eigenvectors of P^α; we disregard the trivial solution $\psi_0 = 1$. This last expression leads naturally to DMs dimensionality reduction. In fact, we can approximate (11) as

$$D_t^2(x,z) \sim \sum_{j=1}^{d(t)} \lambda_j^{2t}(\psi_j(x) - \psi_j(z))^2,$$

where for a given precision δ we define $d(t) = \max\{l : |\lambda_l^t| > \delta|\lambda_1^t|\}$. In other words, if we define the projection

$$\Psi_t(x) = \begin{pmatrix} \lambda_1^t\psi_1(x) \\ \vdots \\ \lambda_{d(t)}^t\psi_{d(t)}(x) \end{pmatrix}$$

of the original points x_i into the $d(t)$ dimensional space $\mathbb{R}^{d(t)}$, the diffusion distance on the original space can be approximated by Euclidean distance of the $\Psi_t(x)$ projections in $\mathbb{R}^{d(t)}$. That is, we have

$$D_t^2(x,z) \sim \sum_{j=1}^{d(t)} \lambda_j^{2t}(\psi_j(x) - \psi_j(z))^2 = ||\Psi_t(x) - \Psi_t(z)||^2.$$

If the parameters have been chosen adequately, we will have now the original data embedded in a lower dimension space in such a way that we preserve the local geometry of the original one and transform the diffusion distance on the initial feature space to Euclidean distance on the embedding. In view of this last fact, it is quite feasible to apply now the k–means algorithm over this new space. We will thus obtain k clusters $C_1, \cdots, C_k$ that are directly related with the clusters $(A_1, \cdots, A_k)$ in the original space $\mathcal{S}$ as $A_i = \{x_j | \Psi_t(x_j) \in C_i\}$. The previous steps are summarized in Algorithm 1.

The last remaining question is how to choose the parameter α. It can be seen [5] that two extreme cases arise. When $\alpha = 1$ the infinitesimal generator L_1 of the Markov chain is just the Laplace–Beltrami operator in the manifold and the Markov chain converges to Brownian motion. In particular, there is no influence of the underlying distribution and we can expect the diffusion projection to capture the underlying geometry. On the other hand, when $\alpha = 0$ we obtain the normalized graph Laplacian typically used in spectral dimensionality reduction and clustering. For a function f, the infinitesimal generator L_0 acts on it as

$$L_0 f = \frac{\Delta(fq)}{q} - \frac{\Delta(q)}{q} f,$$

and, in general, the underlying density q will also influence the diffusion coordinates, except for uniform densities, where we obtain again $L_0 f = \Delta f$. In what follows, we will consider the case $\alpha = 1$.

Algorithm 1. Diffusion Maps Algorithm

1: Given $\mathcal{S} = \{x_1, \cdots, x_n\}$, our data set.
2: Construct $G = (\mathcal{S}, W)$ where W_{ij} is a symmetric, positive kernel.
3: Define the density function as $q(x_i) = \sum_{j=1}^{n} w(x_i, x_j)$.
4: Normalize the weights by the density: $w^{(\alpha)}(x, y) = \frac{w(x,y)}{q(x)^{\alpha} q(y)^{\alpha}}$.
5: Define the transition probability $P_{ij} = p(x_i, x_j) = \frac{w^{(\alpha)}(x_i, x_j)}{g^{(\alpha)}(x_i)}$, where $g^{(\alpha)}(x_i) = \sum_{j=1}^{n} w^{(\alpha)}(x_i, x_j)$.
6: Obtain eigenvalues $\{\lambda_r\}_{r \geqslant 0}$ and eigenfunctions $\{\psi_r\}_{r \geqslant 0}$ of P such that

$$\begin{cases} 1 = \lambda_0 > |\lambda_1| \geqslant \cdots \\ P\psi_r = \lambda_r \psi_r. \end{cases}$$

7: Compute the threshold $d(t) = \max\{l : |\lambda_l^t| > \delta |\lambda_1^t|\}$.
8: Formulate Diffusion Map:

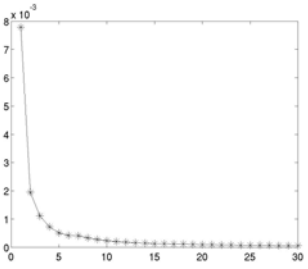
$$\Psi_t = \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \vdots \\ \lambda_{d(t)}^t \psi_{d(t)}(x) \end{pmatrix}.$$

9: Cluster over the embedding (if desired).

3 Diffusion Maps for Meteorological Data

We will illustrate next how to apply Diffusion Maps to the analysis of the meteorological data for the Iberian peninsula. More precisely, we will not work with actual weather measures but, instead, with surface predictions provided by the European Center for Medium-Range Weather Forecasts, ECMWF [7]. These predictions will be in general close to the actual atmospheric conditions. Moreover, they have the advantage of providing values over an uniform large scale grid, giving for each day eight snapshots of the meteorological conditions over a large region. As mentioned before, we will use meteorological forecasts for a whole year, from March 2009 to February 2010, in a grid of resolution 0.25°, working with five surface variables, wind velocity and direction (decomposed in sine and cosine), pressure and temperature, that are available every three hours.

Weather forecasts over a given time period have certainly a time component; if they are given for a large geography area, there is a spatial component as well. Both components are natural candidates for data compression, i.e., we can consider two types of data compression, time and spatial compression, and we will do so next. For time compression we will consider for each of the $1,995$ nodes in the ECMWF grid for the Iberian peninsula a vector made up by the 5 variable surface forecasts at that node for the whole year. Thus, we assign to each node a vector with dimension $5 \times 8 \times 365 = 14,600$, and we seek to compress that vector into another one of a much lower dimension in a way that still provides meaningful information for each node. For spatial compression, we will consider for a given day the vector made up of the 8 daily forecasts of the 5 surface variables at each node. Now, we assign to each day a vector with dimension $5 \times 8 \times 1,995 = 79,800$. We discuss time compression first.



t	$\delta = 0.1$	$\delta = 0.01$
1	**3**	23
2	1	**3**
3	1	2
4	1	1
5	1	1
6	1	1
7	1	1

Fig. 2. Choosing the appropriated embedding dimension. The left hand image shows the eigenvalue decay. The right hand table shows dimension values for two precision values and different values of the diffusion step, t.

3.1 Time Compression

In this section we will explore how to compress for a given grid point a whole year of weather forecasts. Recall that we have $1,995$ grid points for the Iberian peninsula and for each of them we have a $14,600$–dimensional pattern. We will consider first dimensionality reduction and then clustering over the projected data. The first step in DM DR is parameter selection. We shall work with Gaussian kernels and our first task should be to choose the kernel width parameter ϵ. This is a crucial task; we will apply the ideas in [10]. As the parameter ϵ defines the neighborhood size, a good idea is fixing ϵ as $2 \times \text{median}\,(d_{ij})$, where d_{ij} represents the Euclidean distance between points x_i and x_j for every pair of points in $\mathcal{S}$. The main reason to select this measure is its robustness to outliers. For this concrete problem the ϵ value obtained is 2.6×10^{05}.

The next choices are the diffusion step t and the dimension $d(t)$ of the reduced space. We first apply the threshold–based dimension selection method proposed in [5] and already explained in Section 2. Recall that for a given t, we select: $d(t) = \max\{l : |\lambda_l^t| > \delta|\lambda_1^t|\}$. The δ precision parameter is fixed in our experiments either at 0.1 or at 0.01. The right hand table in Figure 2 shows $d(t)$ values for both precisions and different values of t; the figure at left shows eigenvalue decay. In that table we discard $d(t)$ values of 1, as probably yielding a too drastic dimension reduction. We also discard the probably too high value of 23 obtained for $\delta = 0.01$ and $t = 1$. This leaves us with the options $t = 1$, $d = 3$ for the 10% precision and $t = 2$, $d = 3$ or $t = 3$, $d = 2$ for the 1% precision. For a more homogeneous comparison we will work with $d = 3$ in both cases. As the results are very similar with both groups of parameters we just present the images obtained with $t = 1$. The right hand image in Figure 3 shows a bidimensional representation of the corresponding embedding on the coordinates associated to the two largest eigenvalues. To check the relevance of the new coordinates, we will apply k–means on them and analyze the resulting clusters. As it is well known, it is usually quite difficult to determine the number k of clusters to be looked for. One option is likelihood–based methods such as BIC or ICL [3] that assume underlying Gaussian mixture models and penalize their complexity. However, likelihood values are often scale dependent and the complexity penalization may

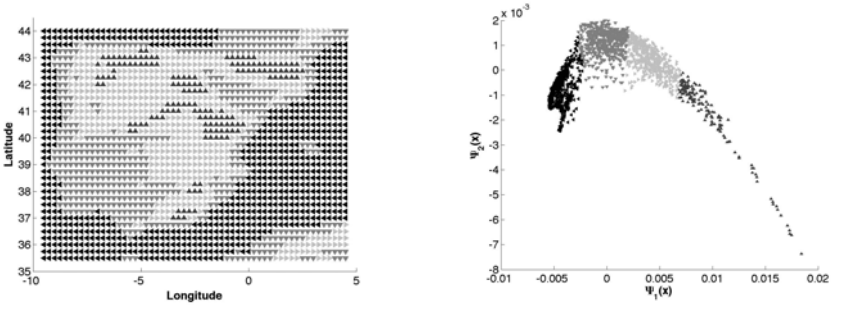


Fig. 3. Grid representation of cluster points with parameters $t = 1$, $d = 3$ and $\delta = 0.1$. The bidimensional embedding space showing the four clusters obtained with $k-$means method is on the right and a map showing the coordinates associated is on the left. Note that the symbol ▲ is drawn in four positions (up ▲, down ▼, right ► and left ◄) and different grayscale tones are used for a better viewing.

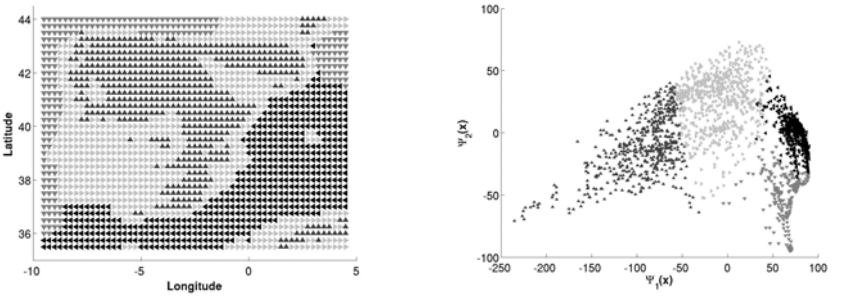
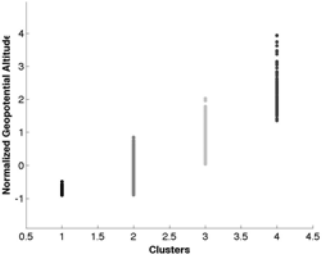


Fig. 4. Grid representation of clusters points with PCA. The right image shows clusters over the PCA projection of the first two components and the left image shows a map representing the original coordinates associated with the groups made on the projection.

be too small to make up for that. There are other, more robust options such as G–means [8] but recall that in our case we are trying to compress a whole year of data over a sample of $1,995$ grid points over the Iberian peninsula. It is thus natural to depict the clusters over a two dimensional grid representation and analyze the resulting figure. In order to make cluster visualization easy in a gray scale, we just choose 4 clusters and assign to them gray scale values. The result appears in the left hand image in Figure 3 (results for $t = 2$ are quite similar). As it can be seen, it depicts clearly the contour of the Iberian peninsula as well as that of the north of Africa. Sea grid points are clearly identified as one of the four clusters. The other three clusters could be roughly assigned to coasts and lower valleys, the central Spain plateaus and the mountains. We observe that while final clusters in k–means usually depend on the random initial centroid election, in this case the results are usually the same. For comparison purposes, we have also applied here PCA. In principle, the covariance matrix would be $14,600 \times 14,600$ but since the data matrix has rank at most $1,995$, this is the



Cluster	$\mu \pm \sigma$
$C1$	-0.88 ± 0.05
$C2$	-0.23 ± 0.35
$C3$	0.90 ± 0.35
$C4$	2.09 ± 0.54

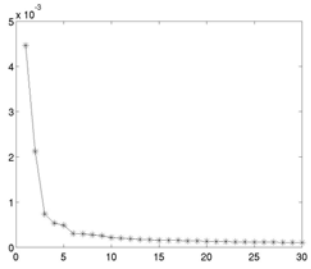
Fig. 5. 4–means clusters over normalized geopotentials that comes from a 3-dimensional embedding obtained with $t = 1$. The table on the right represents normalized geopotential altitude mean and deviation per cluster.

effective dimension. The left hand of figure 4 shows the embedding that would be provided by the two first components. It has a similar structure to that of the DMs embedding although that one is better determined (more on this below). The right hand shows the 4 clusters obtained by k–means over the 3–dimensional PCA embedding. It assigns two cluster for the sea and another two for the land, which hints to a less precise embedded structure. Again, final clusters depend on the initial centroids, but the structure shown is consistently obtained.

In summary DMs dimensionality reduction and clustering of one year of weather prediction from an initial dimension of $14,600$ to a much lower of 3 gives us a meaningful geographical representation. Viewing the resulting map, it could be argued that, at the end, what we get could be a height based representation of the underlying grid. Since we work with surface forecasts, each grid point has associated its geopotential in the underlying orography model. We have considered for the four clusters the geopotentials altitude (normalized to zero mean and 1 deviation) of each one of their points. Figure 5 depicts geopotential values for each cluster (left image for $t = 1, d = 3$ and the right one for $t = 2, d = 3$). While on the potentials on the left most of the groups are quite concentrated, this is not the case with the other ones. The table in Figure 5 gives each cluster mean and standard deviation; they are clearly separated and a Mann–Whitney–Wilcoxon test shows each cluster distribution to be statistically different. We can conclude that height alone does not explain the cluster structure found and that DMs has indeed be able to greatly reduce the original dimension while capturing meaningful information on the initial patterns.

3.2 Spatial Compression

We turn now to the spatial component of weather forecasts. We work with the same forecast set as before but now our sample is made of 365 patterns that contain for each day of the year the eight 3–hourly forecasts of wind velocity, cosine and sine of the wind direction, pressure and temperature for each one of the $1,995$ grid points. Pattern dimension is now $79,800$. We proceed as before, using now an ϵ value of 2.8×10^{05} and determining first the diffusion step t and its associated dimension $d(t)$ at both the 0.1 and 0.01 precisions. The right



t	$\delta = 0.1$	$\delta = 0.01$
1	**5**	9
2	2	**5**
3	2	2
4	1	2
5	1	2
6	1	2
7	1	1

Fig. 6. Choosing the appropriated embedding dimension. The left hand image shows the eigenvalue decay. The right hand table shows dimension values for two precision values and different values of the diffusion step, t.

hand table in Figure 6 shows for them the $\mu \pm \sigma$ pairs while the image at left shows eigenvalue decay. We also discard a projected dimension of 1 and, on the remaining ones, we settle for a dimension of 5 that we achieve when $t = 1$ at the 0.1 precision and when $t = 2$ at the 0.01 one; we have just depicted the results with $t = 1$ as the are very similar to the obtained with $t = 2$. We also use clustering to ascertain the relevance of the reduced dimensions. Since we associate here patterns with calendar days, a logical assumption would be that the reduced patterns should capture seasonal weather behavior and we apply again k–means with $k = 4$ (four seasons).

The right hand image in Figure 7 shows the bidimensional representation of the projected data over the coordinates that correspond to the two largest eigenvalues. The left hand image shows the correspondence between clusters and the seasonal distribution of days. They depict for each cluster the histogram of the distribution of spring, summer, fall and winter days, with the slight modification of considering March, April and May as spring, June, July and August as summer, September, October and November as fall and December, January and February as winter. Clearly the first cluster in both figures can be associated with summer, while winter is associated with the fourth cluster. With a slight abuse of language, we will refer to them as the summer and winter clusters. On the other hand, and perhaps not too surprisingly, there is no such clear cut association of the other two clusters with either spring or fall. Moreover, notice that the size of the clusters associated to summer is bigger than that of the winter clusters, pointing to the fact that for the Iberian peninsula, weather summer is longer than calendar summer and the other way around for winter. Results for PCA are shown in figure 8. The embeddings looks now remarkably similar as well as the cluster structure. In particular, PCA performs better now than for time compression. As a possible explanation, notice that the time correlation of meteorological variables at a single spot (particularly wind) is rather weak. On the other hand, the space correlation over a large area at a given time is stronger. Although this must be further studied, this large spatial correlation may explain why a decorrelation scheme for Dimensionality Reduction such as PCA may perform rather well in this setting.

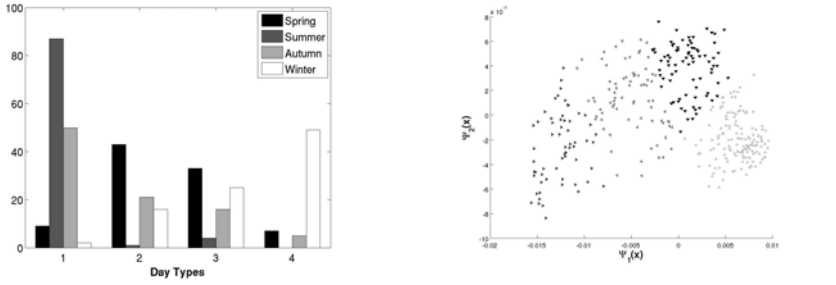


Fig. 7. Day types for the four seasons with $t = 1$, $d = 5$ and $\delta = 0.1$. The bidimensional embedding space showing the four clusters obtained with $k-$means method is on the right and the correspondence between clusters (labeled from 1 to 4) and the seasonal distribution of days is on the left.

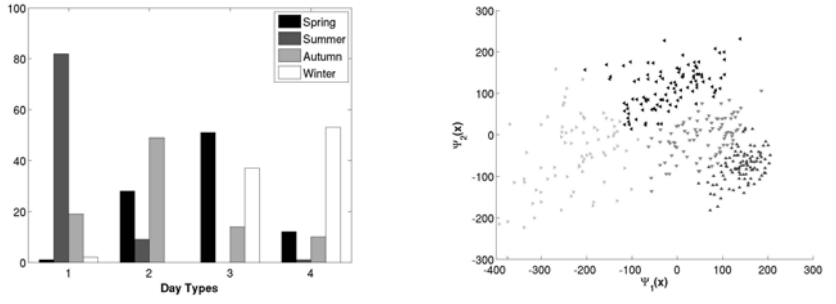
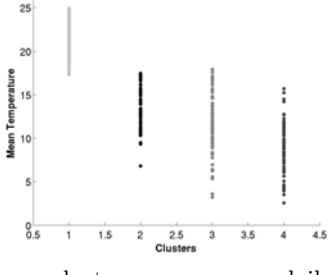


Fig. 8. Day types for the four seasons with PCA. The right image shows clusters over the PCA projection of the first two components and the left image shows the seasonal distribution of days.



Cluster	$\mu \pm \sigma$
$C1$	21.45 ± 2.13
$C2$	13.32 ± 2.32
$C3$	11.72 ± 3.30
$C4$	9.14 ± 2.95

Fig. 9. 4–means clusters over mean daily temperature that comes from and 5-dimensional embedding made with $t = 1$. Mean and deviation per cluster of average daily temperature is presented in the table on the right.

We could also argue here that the embedded representations may essentially just capture a seasonal variable, most likely temperature, instead of performing a more comprehensive dimensionality reduction. As done before, we have considered the daily mean temperature distribution for each one of the four clusters

which is shown in Figure 9; the table in Figure 9 shows the corresponding means and standard deviations. As it could be expected, the summer cluster shows the highest mean temperatures. Mean temperatures are clearly different for the intermediate clusters, lower than those of the summer cluster but higher that those in the winter cluster. Finally, the winter cluster has the lowest daily mean temperature, but it shows a quite big spread. It is thus clear that DMs also captures here more information than that provided by temperature values.

4 Conclusions

Diffusion Maps is a recent method for dimensionality reduction that assumes the original patterns to be embedded in a lower dimensional manifold and allows to consider the influence on the embedding of both the underlying manifold geometry and the data distribution on it. Moreover, it introduces a Markov chain model for data proximity with a natural diffusion distance that becomes Euclidean distance on the embedded space. In turn, this makes quite reasonable to apply on that space clustering methods that rely on Euclidean distance, more particularly k–means. In this paper we have applied Diffusion Maps for the compression and analysis of weather forecasts, considering data for a whole year over a two dimensional grid with a $0.25°$ resolution. The data set has both a time and a spatial component and it is feasible to consider data compression for each one of them. For time compression we have shown that clustering over three dimensional projected patterns gives clusters that relate weather patterns to spatial location in a more meaningful way that that achieved by PCA. For spatial compression we have shown that k–means clustering of five dimensional projected patterns yields clusters that capture seasonal weather patterns. However, here PCA DR gives similar results.

In any case, there is work yet to be done. A first task is, of course, the selection of the number k of clusters when applying k–means. Here $k = 4$ is a good selection for the problems considered but more general and principled approaches and algorithms have to be applied. A second issue is the application of Diffusion Maps in a supervised setting, building dimensionality reduction and clustering on a training set but being able to apply them to new, unseen patterns without having to recompute the eigenvalue and eigenvector structure over which Diffusion Maps rely. As mentioned in the paper, the most promising way to do so is to work with Gaussian kernels and use Nyström formula, but besides having to check whether this works correctly over new patterns, this raises a third issue, how to select the width parameter of these kernels. On top of this the correlation structure of meteorological variables has to be further studied to determine which data organizations can be effectively captured by simpler methods such as PCA. We are currently working on these and other related issues.

Acknowledgement. The authors acknowledge partial support from grant TIN2010-21575-C02-01 of the TIN Subprogram of Spain's MICINN and of the Chair for the Automatic Machine Learning in Modelling and Prediction. The

first author is also supported by the FPI-UAM grant and kindly thanks the Applied Mathematics Department of Yale University for receiving her during a visit supported by the FPI-UAM grant.

References

1. Belkin, M., Nyogi, P.: Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation 15, 1373–1396 (2003)
2. Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J.F., Vincent, P., Ouimet, M.: Learning eigenfunctions links spectral embedding and kernel PCA. Neural Computation 16, 2197–2219 (2004)
3. Biernacki, C., Celeux, G., Govaert, G.: Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 719–725 (2000)
4. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
5. Coifman, R., Lafon, S.: Diffusion Maps. Applied and Computational Harmonic Analysis 21, 5–30 (2006)
6. Coifman, R., Lafon, S., Lee, A., Maggioni, N., Nadler, B., Warner, F., Zucker, S.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the National Academy of Sciences 102, 7432–7437 (2005)
7. European Center for Medium–Range Weather Forecasts: http://www.ecmwf.int/
8. Hammerly, G., Elkan, C.: Learning the K in K–means. In: Advances In Neural Information Processing Systems, vol. 17 (2003)
9. Nadler, B., Lafon, S., Kevrekidis, I.G., Coifman, R.: Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck operators. In: Advances In Neural Information Processing Systems, vol. 18 (2005)
10. Rabin, N.: Data mining in dynamically evolving systems via diffusion methodologies. PhD Thesis, Tel-Aviv University (2010)
11. Rogers, L.C.G., Williams, D.: Diffusions, Markov processes, and martingales. Foundations, vol. 1. Cambridge Mathematical Library (2000)
12. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–895 (2000)
13. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72, 2729–2730 (2009)

Computational Complexity Reduction and Interpretability Improvement of Distance-Based Decision Trees

Marcin Blachnik[1] and Mirosław Kordos[2]

[1] Silesian University of Technology, Department of Management and Informatics, Katowice, Krasinskiego 8, Poland
marcin.blachnik@polsl.pl
[2] University of Bielsko-Biala, Department of Mathematics and Computer Science, Bielsko-Biała, Willowa 2, Poland
mkordos@ath.bielsko.pl

Abstract. Classical decision trees proved to be very good induction systems providing accurate prediction and rule based representation. However, in some areas the application of the classical decision trees is limited and more advanced and more complex trees have to be used. One of the examples of such trees are distance based trees, where a node function (test) is defined by a prototype, distance measure and threshold. Such trees can be easily obtained from classical decision trees by initial data preprocessing. However, this solution dramatically increases computational complexity of the tree. This paper presents a clustering based approach to computational complexity reduction. It also discusses aspects of interpretation of the obtained prototype-threshold rules.

Keywords: Decision trees, instance selection, clustering, prototype rules.

1 Introduction

Decision trees proved to be very useful tool for knowledge induction in classification and regression problems [15,4]. There are two benefits of using them; first because they make the prediction very accurate and second, because the obtained solutions are easy to interpret and can be formulated as a set of rules.

The idea of decision trees also evolved to forests of decision trees, where the output of the system is obtained by voting of individual trees. This extends the generalization abilities of single classical decision tree. The idea of boosting is also closely related to decision trees, where the weak learner is often realized by the decision tree [2]. Unfortunately despite the fact that forests of trees increase the accuracy of the system, they also reduce the comprehensibility of the system thus rejecting one of their advantages.

In the literature decision trees also evolved in a different direction, based on replacing a classic node function - combination of attribute - threshold pair, by more advanced solutions like Naive Bayes tree [9], where each node of the tree consists of a naive Bayes rule. In other approaches some authors like Sumner proposed a decision tree with a linear node function [14]. More advanced solutions also consider neural networks

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 288–297, 2012.

(MLP networks) as nodes of the tree [10,11] or a hybrid trees as in [12]. In general all these solutions are a subset of heterogeneous decision trees or forests of decision trees [8] where a node test can realize any type of decision function. Such mixture of decision trees and complex node functions directly leads to so called hybrid machines or hybrid intelligence [6,5,1].

There are two approaches to building the hybrid or heterogeneous decision trees.

1. The node splitting function can be seen as an independent decision making model with any shape of the decision boundary for example naive Bayes classifier, linear model, quadratic model, etc.
2. The node splitting function can be also realized by initial transformation of the input data or extension of the original input data by some *n*ew features which are constructed before the induction process. In that case the node function remains the basic one; attribute - threshold pair, but the dataset is modified by appending some new attributes obtained by some mapping function. A typical example of that solution are attributes obtained by linear or nonlinear projections.

The two approaches usually lead to different solutions and different trees. The advantage of the first one is construction of the decision function for a particular data available in the particular node, while in the second approach the decision tree is only restricted to the nonlinear transformations used for construction of the new attributes. On the other hand the data delivered to all the nodes except the root of the tree is strongly unbalanced, what makes an important problem in construction of the decision function in the first approach. This is especially important when classical models like linear or nonlinear classifiers are used as the node splitting function. That restriction does not apply to the second scenario, because splitting indexes like *i*nformation gain, or *G*ini index are designed to support unbalanced data.

An interesting example of the nonlinear transformation function of the input data is the kernel or distance based mapping. In case of that transformation the input data are replaced by the distances of each vector to all other vectors, such that denoting n as the number of instances and m as the number of features, the original dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ of size $n \times m$ where $\mathbf{x}_i$ is a vector $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,m}]$ is transformed to the dataset $\mathbf{Z}$ by mapping:

$$z_{i,j} = D(x_i, x_j) \tag{1}$$

where $D(\mathbf{a}, \mathbf{b}) = ||\mathbf{a} - \mathbf{b}||_L$ is the distance function or L norm, such that after the transformation the resulting dataset $\mathbf{Z}$ becomes of size $n \times n$ and the so called *n*ew attributes are defined as distances from reference vectors to instances of the training set. The idea of such mapping was proposed by Duch and Grąbczewski in [8] and leads to the piecewise spherical decision boundaries. Such transformation is an example of heterogeneous trees of type 2.

As it can be noticed that transformation has an important impact on the number of attributes of the dataset, because usually $n \gg m$. This requires considering the computational complexity of the model. The computational complexity of decision trees is defined as $O(n, m) = n \log(n) \cdot m$. According to that equation the number of attributes has linear influence on the computational complexity of a decision tree. Performing

the kernel or distance mapping leads to $m = n$, what makes the overall computational complexity of order $O(n, m) = n \cdot m \log(n) \underset{m=n}{\Rightarrow} n^2 \log(n)$ that is above quadratic complexity. For large datasets that makes this solution very limited.

In that paper we show how to overcome such limitation by reducing the number of reference vectors when building the distance matrix. We also provide the possible interpretation leading to prototype based rules, which can be seen as an extension of classical propositional logical rules.

2 Prototype Based Rules and Interpretability of Distance Based Decision Trees

A typical way of understanding the data leads to propositional logical rules. However, this way of data representation is limited and unable to express some states of nature. A typical example of such limitation is majority voting, a concept which can be simply formulated using M-of-N rules. Given vector $\mathbf{x}$ of m binary answers $x_i \in \{0, 1\}$ (answers of voting participants) the explanation of "majority is for it" concept can be written as $\sum_{i=1}^{m} x_i > 0.5m$, what naturally explains the nature of any voting system. To learn the same concept from binary data with propositional logical rules requires $\binom{m}{m/2}$ rules, what would lead to very low comprehensibility. Such majority voting can be easily learned from the data using the distance or similarity measure and prototype based rules. Considering the Hamming distance and some prototype $\mathbf{p} = [1, 1, \ldots, 1]^T$ of m values, such a rule may be expressed as: $||\mathbf{x} - \mathbf{p}|| < m/2$.

From the perspective of psychology and natural human data processing the concept of prototypes is well known since 70's from the work of Eleonor Rosch [16] and more recent cognitive experiments, which show that the categorization of natural objects and states of nature is based on memorization of numerous examples and creation of prototypes that are abstractions of these examples [17].

In computer science prototype based methods were also commonly used. A good example is nearest neighbor classifier (kNN) and its modifications [18] including case based reasoning. In the classical kNN based approaches the main idea is to provide high accuracy, reducing the interpretability to selecting the nearest instances. However, in prototype based rules we focus not just on accuracy and generalization, but rather on keeping as small as possible the set of instances that are able to represent the underlying concept [7,13], thus finding the compromise between accuracy and complexity (number of prototypes).

Currently prototype based representation, are not used as rule based systems. In practice, most of the common rule based systems including propositional logical and fuzzy rules can be expressed as prototype based rules. That types of rule based system can be defined depending on selected (dis)similarity measure. For example considering Chebyshev distance ($L_{\inf}$ norm) and a nearest neighbor based reasoning one may obtain propositional logical rules and with Euclidian distance or Manhattan distance one may obtain prototype based system equivalent to fuzzy rule based system with $\max$ aggregation function and a product of Gaussian or triangular membership functions [19].

There are two types of prototype based rules:

- Prototype Threshold Rules (PT-rules). In such system the rule is defined as a pair of a prototype $\mathbf{p}_i$ and an associated threshold θ_i value such that i-th rule can be written as:
$$\text{If } S(\mathbf{x}, \mathbf{p}_i) > \theta_i \text{ Then } C(\mathbf{x}) = C(\mathbf{p}_i) \tag{2}$$
where $C(\cdot)$ is a function returning some information associated with the prototype, for example a class labels.
- Nearest Neighbor Rule (PN-rules). In this kind of rules the most similar prototype is selected:
$$\text{If } k = \arg\max_i S(\mathbf{x}, \mathbf{p}_i) \text{ Then } C(\mathbf{x}) = C(\mathbf{p}_k) \tag{3}$$
so the output value depends on the internal mutual relations between prototypes. Considering some fuzzy interpretation or operator "is similar to" this can be also written as:
If $\mathbf{x}$ is similar to $\mathbf{p}_i$ then it is of the same class with some support w_i:
leading to the list of rules
$$\begin{aligned} &\text{If } w_i = S(\mathbf{x}, \mathbf{p}_i) \text{ Then } C(\mathbf{x}) = C(\mathbf{p}_i) \text{ with support } w_i \\ &\text{If } \mathbf{x} \text{ is similar to } \mathbf{p}_2 \text{ Then } C(\mathbf{x}) = C(\mathbf{p}_2) \text{ with support } w_2 \\ &\quad \ldots \\ &\text{If } \mathbf{x} \text{ is similar to } \mathbf{p}_v \text{ Then } C(\mathbf{x}) = C(\mathbf{p}_v) \text{ with support } w_v \end{aligned} \tag{4}$$
where $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_v]^T$ is a set of v prototype vectors and w_i is the support for the conclusion of the i-th rule. The final output of this rule based system is obtained by:
$$C(\mathbf{x}) = A(w_i, C(\mathbf{p}_i)) \tag{5}$$
where $A(\cdot)$ is an aggregation operator, which joins conclusions of individual P-rules.

Distance based decision trees can be interpreted as a set of PT-rules, where a single prototype $\mathbf{p}$ with associated similarity measure $S(\cdot, \mathbf{p})$ defines for a given threshold θ a subspace $\mathcal{S}_p$ of vectors $\mathbf{x}$ for which $S(\mathbf{x}, \mathbf{p}) < \theta$. This subspace is centered at the position of the prototype $\mathbf{p}$ and may have different shapes, depending on the similarity function. Such interpretation defines a crisp logical rule for the new feature $x_p = S(\mathbf{x}, \mathbf{p})$. In this case the antecedent part of a PT-rule uses similarity to a single prototype and the class label of that prototype (in classification tasks) is the consequence part (2).

The similarity value may be used to estimate confidence factor for such rules. The rescaled difference $\mu_p(\mathbf{x}) = S(\mathbf{x}, \mathbf{p}) - \theta$ may obviously be interpreted as a fuzzy membership function defining the degree to which vector $\mathbf{x}$ belongs to the fuzzy subspace $\mathcal{S}_p$. Many similarity functions are separable (are defined as an aggregation of a similarities of each attribute independently). For example:
$$S(\mathbf{x}; \mathbf{p}, \sigma) = \prod_i S(x_i, p_i; \sigma_i) \tag{6}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and $\mathbf{p} = [p_1, p_2, \ldots, p_n]^T$ are n-dimensional vectors, and $S(\cdot)$ is similarity function.

Threshold P-rules with separable similarity functions can be interpreted as fuzzy rules (F-rules) with a product as a fuzzy and aggregation operator. Linguistic interpretation of F-rules relies on semantics of linguistic values assigned to each linguistic variable as adjectives describing the membership functions. Such representation is sensitive to context. A good example of this context dependence is an adjective *high* that may describe objects of different types, for example *a person*, but even in this case different kinds of people: kids, women or basketball players will require different membership function representing variable "high". Thus indirectly fuzzy rules have to rely on prototypes of objects or concepts to define the context, but since in fuzzy rules this context is not explicitly represented, confusion is quite likely. P-rules make this reliance explicit always pointing to prototypes of particular concepts, allowing each concept to be decomposed into independent features that may be treated as linguistic values in the fuzzy sense.

3 Computational Complexity Reduction

In the introduction the problem of computational complexity of distance based decision trees was described. As it was pointed out the computational complexity of such system is of the order $n^2 \log(n)$ that makes that approach limited just to small datasets. A typical approach to overcome the computational complexity of datasets with large number of attributes are random forests [3]. This approach is based on constructing a set of decision trees on randomly selected subset of attributes. In such a case, when each tree is constructed on randomly selected subset of attributes with fixed $m' \ll m$ the computational complexity is becoming m/m' smaller, and instead of $n \log(n) \cdot m$ one may obtain $k \cdot n \log(n) \cdot m'$, where k is the number of constructed decision trees in the forest.

Unfortunately this approach is not based on a single tree but on a set of trees. This has a positive influence on the accuracy of the model, but on the other hand it reduces the comprehensibility of the system. To preserve the interpretability it is desired to build rather a single tree. Reduction of the computational complexity in that case can be obtained by preselecting instances used as reference points. This can be obtained by clustering the training set into h clusters and then extracting the centers of the clusters $\mathbf{v}_j$. Such cluster centers $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_h]$ are then considered as reference points for *n*ew attributes. Because $h \ll n$ the computational complexity is reduced and can be controlled by defining the number of cluster centers h. In the presented research the idea of clustering the input data for preselecting prototypes was compared to the full distance based decision trees, and to classical decision trees.

4 Numerical Experiments

4.1 Experiment Description

To verify the proposed approach a series of experiments were performed. In all the experiments three scenarios were compared

1. Classical decision tree with optimized pruning parameter
2. Distance based decision tree with full distance matrix, also with optimized pruning parameter
3. Distance based decision tree based on reference vectors obtained by Fuzzy c-means clustering also with optimized pruning parameter

The experiments were performed on well known datasets from UCI repository including Heart Disease (Cleveland), Diabetes (Pima Indian), Sonar, Breast Cancer (Wisconsin), Ionosphere, Spam Base. In all experiments a 10-fold crossvalidation test was performed and the average results are reported. The tree pruning level and number of clusters were fixed and not optimized within the crossvalidation for any of the methods.

The testing scenarios are presented on figure (1) The number of clusters in all experiments were proportional to the number of samples in each class. The procedure was as follows:

1. determine the number of samples of each class for a given dataset $\mathbf{s} = [s_1, s_2, \ldots, s_c]$
2. calculate the number of clusters by dividing $\mathbf{s}$ by one of the values representing the fraction of the instances in the training set $\mathbf{t} = [10, 5, 2, 1.5, 1.2]$ (The experiments were conducted with the arbitrarily chosen number of clusters proportionally to the dataset size, so that the centers of clusters replaced the original dataset.)
3. for each number of cluster centers perform independent crossvalidation test. In all experiments the instances of each classes were clustered independently and finally the obtained cluster centers were joined into a single set.

When performing the comparison also the computational time was recorded. The time includes not only the tree construction process, but also the time required to build the distance matrix, and the time required for clustering. All the experiments were performed using Spider toolbox for Matlab. The version of Spider used in the experiments is available from `http://mblachnik.pl`. The decision tree used in the experiments was the default Matlab tree and was based on Gini index.

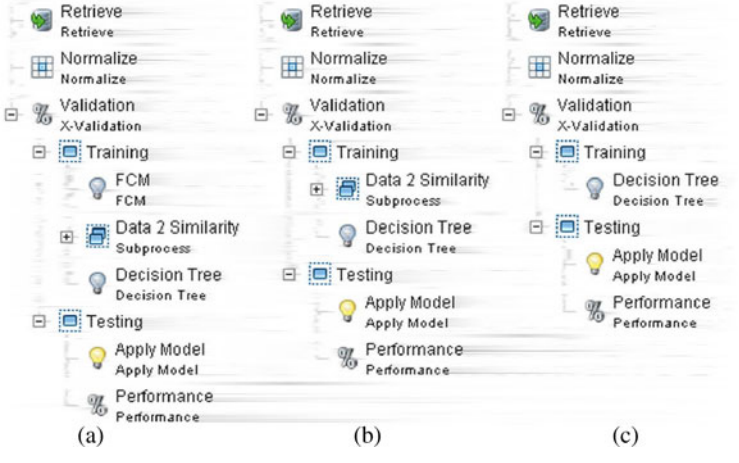
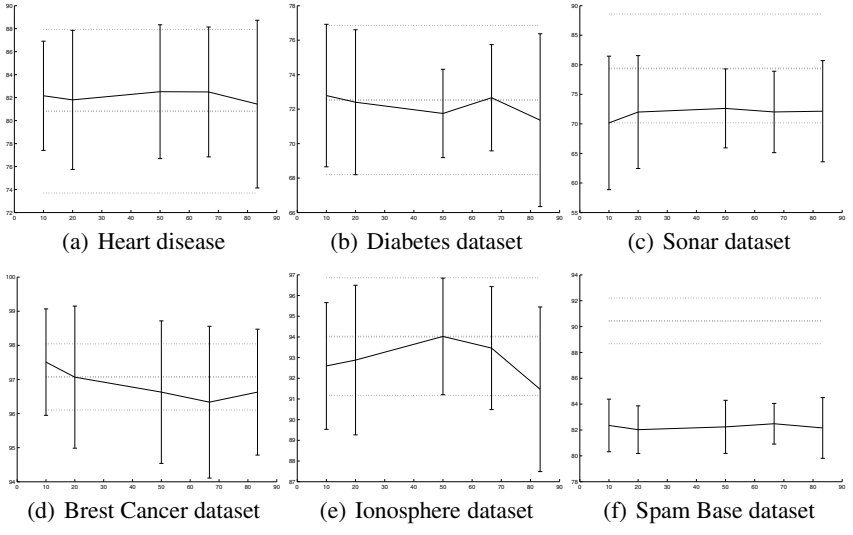


Fig. 1. Testing procedure. Three scenarios: a) using clustering, b) using full similarity data, c) classical decision tree

4.2 Results and Remarks

The above described procedure was used to obtain the results. The relationship between the accuracy and the number of cluster centers (the compression ratio) is presented in figure (2). Values on axis X represent the percent of compression, interpreted as a fraction of the number of cluster centers to the number of instances in the training data. Y axis shows the accuracy of the model. The dotted lines represent the accuracy and standard deviation of the distance based decision tree trained on a full feature set and the solid line is the accuracy of the distance based decision tree obtained by initial clustering for different number of clusters. As it can be seen in most of the cases, except Spam



(a) Heart disease (b) Diabetes dataset (c) Sonar dataset

(d) Brest Cancer dataset (e) Ionosphere dataset (f) Spam Base dataset

Fig. 2. The relationship between the classification accuracy (Y axis) and the number of reference vectors (X axis) for 6 datasets. The accuracy and its standard deviation obtained on a full training set used as a reference point is plotted with a dashed line.

Base and Sonar dataset, there are no significant differences between results obtained with the full dataset and with the clustered data. Moreover, for *H*eart Disease dataset the results of clustered based decision tree are even better then the results obtained with the full similarity matrix. A comparison of the best results on all these datasets in presented in table (1). This table also includes time required to build the models, size of the trees expressed in the number of nodes and as a reference the results of the classical decision tree. An example of the relation between computational time and compression of *S*pam base dataset is presented on figure (3). This figure points out that the dependency is linear. *S*pam Base dataset was used for the comparison of computational complexity, because this dataset is not suitable for the distance based decision trees. This resulted in a very small accuracy and very complex tree structure, such that for most of the parameters the full tree had to be built. When analyzing the results presented in table (1) one may observe that the reduction of the training time does not always occur. This

Table 1. Comparison of the best results obtained in all tree scenarios, using: clustering based distance tree, full distance matrix based decision tree and classical decision tree

Dataset	Clustering based Distance Tree			Distance Tree			Classical Decision tree		
	Acc±Std	time	#nodes	Acc±std	time	#nodes	Acc±std	time	#nodes
Heart disease	82,17±4,95	0,42	3	80,81±7,12	1,42	5	78,7816±7,70	0,05	15
Diabetes	72,79±4,13	1,26	5	72,53±4,33	3,36	15	76,4371±5,87	0,08	43
Sonar	72,62±6,68	0.51	7	79,38±9,20	0,40	3	74,4048±9,80	0,50	21
Breast cancer	97,51±1,56	1,14	7	97,07±0,97	1,09	3	94,8721±2,71	0,03	25
Ionosphere	94,02±2,82	2,09	13	94,02±2,84	1,49	9	89,4524±3,33	0,03	5
Spam base	82,07±1,75	91,15	399	90,44±1,77	161,13	65	92,3276±0,57	1,03	139

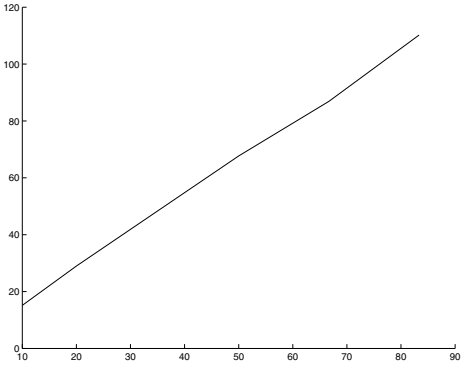


Fig. 3. Time required to build the distance based decision tree as a function of data compression

is related to the depth of the constructed tree. When the stop criteria are able to stop the tree construction process earlier then it results in a lower computational time.

5 Conclusions

In this paper the problem of computational complexity of distance based decision trees was discussed including the aspects of interpretability of that models. A simple approach based on initial clustering of input data to reduce the number of attributes used for tree construction was used to build the tree. This method allows for linear reduction of the computational complexity. In general, the obtained results were comparable to the results obtained with full distance matrix and in some cases were even better like for *H*eart Disease or *B*rest Cancer datasets. It can be also seen that the for both these datasets and also in case of the *D*iabetes dataset the distance based decision trees performed better then the classical decision tree. However, as it was shown in table (1) for some datasets like the *S*pam Base or *S*onar the obtained results after clustering were worse then the results obtained with the full size distance matrix. Moreover, in the case of *S*pam Base the obtained results were significantly worse then the results obtained with a classical decision tree. This means that this dataset does not have piecewise spherical decision border, what confirms the *n*o free lunch theorem. Among all tested datasets it can be seen that for *B*rest Cancer and *H*eart Disease it is possible to extract

simple prototype threshold decision rules (PT-rules). For the *B*rest Cancer dataset a single PT-Rule allows for classification with 97% of accuracy. Also in the case of *S*onar dataset a single PT-rule allows for classification with 79% accuracy, while the classical decision tree creates very complex rule based system. Similarly for the *I*onosphere dataset instead of 22 classical logical rules only 3 PT-rules allow for classification with the accuracy only 2% worse then that of a classical tree.

From the perspective of model comprehensibility it is desired to simplify the PT-rules. At the current state all of the PT-rules include all of the input attributes. This situation arises from the fact that the similarity matrix is created as a preprocessing step, without any node function modification. To increase the interpretability of the PT-rules, reduction and adjustment of appropriate attributes included in the similarity matrices is necessary. To solve this problem the node function has to be modified so that it includes the feature selection step of the input data. That is our future research direction.

In general, the proposed approach is suitable for large datasets reducing the computational complexity of the training process. The presented results confirmed that the approach is rather not dedicated for accuracy improvement, but for decreasing of training time, which for large dataset may be an important limitation for applications of the distance based decision trees. Any accuracy improvement is just a side effect. However, there are other advantages of that kind of trees, like model comprehensibility, especially when decision border can be interpolated as a piecewise elliptical function.

Acknowledgment. The work was sponsored by the Polish Ministry of Science and Higher Education, project No. 4421/B/T02/2010/38 (N516 442138).

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting and variants. ML 36, 105–142 (1999)
3. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
4. Breiman, L., Friedman, J.H., Olshen, A., Stone, C.J.: Classification and regression trees. Wadsworth, Belmont (1984)
5. Corchado, E., Abraham, A., de Carvalho, C.A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
6. Corchado, E., Grana, M., Woźniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
7. Duch, W., Grudziński, K.: Prototype based rules - new way to understand the data. In: IEEE International Joint Conference on Neural Networks, pp. 1858–1863. IEEE Press, Washington, D.C (2001)
8. Grąbczewski, K., Duch, W.: Heterogeneous Forests of Decision Trees. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 504–509. Springer, Heidelberg (2002)
9. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In: Second International Conference on Knowledge Discovery and Data Mining, pp. 202–207 (1996)

10. Kordos, M., Blachnik, M., Wieczorek, T., Golak, S.: Neural Network Committees Optimized with Evolutionary Methods for Steel Temperature Control. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS, vol. 6922, pp. 42–51. Springer, Heidelberg (2011)
11. Kordos, M., Blachnik, M., Wieczorek, T.: Temperature prediction in electric arc furnace with neural network tree. In: Honkela, T., et al. (eds.) ICANN 2011, Part II. LNCS, vol. 6792, pp. 71–78. Springer, Heidelberg (2011)
12. Kordos, M., Blachnik, M., Perzyk, M., Kozłowski, J., Bystrzycki, O., Gródek, M., Byrdziak, A., Motyka, Z.: A Hybrid System with Regression Trees in Steel-Making Process. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 222–230. Springer, Heidelberg (2011)
13. Blachnik, M., Duch, W., Wieczorek, T.: Selection of Prototype Rules: Context Searching Via Clustering. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 573–582. Springer, Heidelberg (2006)
14. Landwehr, N., Mark Hall, E.F.: Logistic model trees. ML 95(1-2), 161–205 (2005)
15. Quinlan, J.R.: C 4.5: Programs for machine learning. Morgan Kaufmann, San Mateo (1993)
16. Rosch, R.H.: Cognitive reference points. Cognitive Psychology 4(7) (1975)
17. Roth, I., Bruce, V.: Perception and Representation, 2nd edn. Open University Press (1995)
18. Stanfill, C., Waltz, D.: Toward memory-based reasoning. Communications of the ACM 29(12), 1213–1228 (1986)
19. Duch, W., Blachnik, M.: Fuzzy Rule-Based Systems Derived from Similarity to Prototypes. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 912–917. Springer, Heidelberg (2004)

Improving the Generalization Capability of Hybrid Immune Detector Maturation Algorithm

Jungan Chen, Feng Liang, and Zhaoxi Fang

Electronic Information Department
Zhejiang Wanli University No.8 South Qian Hu Road
Ningbo, Zhejiang, 315100, China
{friendcen21,liangf_hz}@hotmail.com, zhaoxifang@gmail.com

Abstract. In this work, an augmented hybrid immune detector maturation algorithm applied in anomaly detection is proposed to improve the generalization capability. Experiment results show the algorithm is more effective and its generalization capability to detect more similar patterns is improved.

Keywords: Artificial immune system, generalization capability, hybrid immune detector.

1 Introduction

Nowadays, Hybrid Intelligent Systems are becoming popular due to their capabilities of handling many real world complex problems such as multi objective optimization, data mining [1]. As an intelligent technique, Artificial Immune System (AIS) has been applied to many areas such as computer security, classification, learning and optimization [2]. Negative Selection Algorithm, Clonal Selection Algorithm, Immune Network Algorithm and Danger Theory Algorithm are the main algorithms in AIS [3][4].

Reference [5] mentioned 'Th-cells are general, nonspecific detectors, and so are not efficient at detecting specific pathogens. B-cells, by contrast, can adapt to become more specific, and thus more effective at detecting particular pathogens'. So there is a balance between generality and specialty. The detectors generated by T-detector Maturation Algorithm (TMA) are just like the generalized lymphocytes (or Th-cells) and the detectors generated by affinity maturation are the specialized lymphocytes (or B-cells). So it is reasonable that Hybrid Immune Detector Maturation Algorithm (HIDMA) combines TMA with affinity maturation. HIDMA with Lifecycle Model (HIDMA-LM) is proposed to solve the adaptive problems. Lifecycle is used as a pressure to improve the effect of the selection operator [6].

To improve the generalization capability [7] and detect more similar patterns, an augmented HIDMA-LM algorithm called HIDMA-GC (Generalization Capability) is proposed and applied in anomaly detection.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 298–308, 2012.

2 Algorithm

2.1 Match Range Model

$U=\{0,1\}^n$,n is the length of binary string. The normal set is defined as selves and anomaly set is defined as nonselves. selves∪nonselves=U. selves∩nonselves=Φ. There are two binary strings $Sg=g_1g_2\ldots g_n$, $Sb=b_1b_2\ldots b_n$. The hamming distance between Sg and Sb is:

$$d(Sg,Sb)=\sum_{i=1}^{n}\overline{g_i \oplus b_i} \tag{1}$$

The detector is defined as dct = {<Sb, selfmin, selfmax> |Sb∈U, selfmin, selfmax∈N}. selfmax is the maximized distance between dct.Sb and selves, selfmin is the minimized distance. The detector set is defined as DCTS. Selfmax and selfmin is calculated by setMatchRange(dct, selves), i∈[1, |selves|], $self_i$∈selves

$$\text{setMatchRange}=\begin{cases}\text{selfmin}=\min(\{d(self_i,dct.Sb)\})\\ \text{selfmax}=\max(\{d(Self_i,dct.Sb)\})\end{cases} \tag{2}$$

[selfmin,selfmax] is defined as self area. Others are as nonself area. The antigen is defined as Ag= {<Sg > | Sg∈U}, The antigen set is defined as AGS.

Suppose there is one antigen ag∈AGS and one detector dct ∈ DCTS. When d(ag.Sg,dct.Sb) $\notin$ [dct.selfmin, dct.selfmax], ag is detected as anomaly. It is called as Range Match Rule (RMR) shown in equation 3.Value true means that ag is anomaly.

$$\text{RMRMatch}(ag,dct)=\begin{cases}\text{false}, d(ag.Sg,dct.Sb)\in[dct.selfmin,dct.selfmax]\\ \text{true}, d(ag.Sg,dct.Sb)\notin[dct.selfmin,dct.selfmax]\end{cases} \tag{3}$$

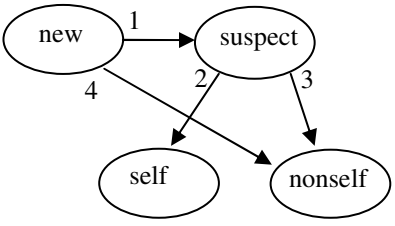
Based on RMR, the detect procedure detect(ag, DCTS) is defined as equation 4. True means that ag is anomaly.

$$\text{detect}(ag,DCTS)=\begin{cases}\text{true}, \exists dct_k\in DCTS, \text{RMRMatch}(ag,dct_k)=\text{true}\\ \text{false}, \quad \text{others}\end{cases} \tag{4}$$

2.2 The State Transformation Model

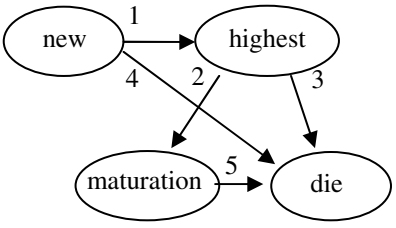
In this model, The antigen is redefined as Ag= {<Sg,state,undetectedCount > |undetectedCount∈N, state∈{'new','suspect','self','nonself'}}. Antigen has four states shown in Fig.1. State 'new' means an antigen just inject into the algorithm and 'nonself' means an antigen is detected by the detectors. If an antigen cannot be detected after one generation, its state is changed to 'suspect' and its undetectedCount is inscreased. If an antigen cannot be detected over many generations and

its undetectedCount is bigger than the max undetected generations, maxUndetectedCount, its state is changed to 'self'. The 'self' and 'nonself' will be removed from AGS.



1. can not be detected by existed maturation detectors
2. can not be detected over specific generation
3,4. detected by any new or maturation detectors

Fig. 1. Antigen's transformation model



1. has the highest affinity with antigen
2. detected an nonself antigen
3,4. has lower affinity with all antigens and can not detect any antigen
5. the lifecycle decreased to 0

Fig. 2. Detector's transformation model

The detector is redefined as dct = {<Sb, d, harmmax, selfmin, selfmax, state, lifecycle, detectedAgNum, oldDetectedAgNum > | d,harmmax, selfmin, selfmax, lifecycle, detectedAgNum, oldDetectedAgNum∈N, state∈{'new','highest', 'maturation','die'}}. Detector has also four states shown in fig.2. State 'new' means a detector or lymphocyte is just generated. If a detector detects any nonself antigen, its state is changed to 'maturation'. If a detector has the highest distance or affinity with a specific antigen than other detectors, its state is changed to 'highest'. Otherwise, its state is changed to 'die' and it will be removed from DCTS.

Other properties in the definition of Ag, dct are calculated by the following steps:

$$M = \mid AGS \mid, N = \mid DCTS \mid, i \in [1, M], j \in [1, N] \tag{5}$$

The value i is the index of antigen in AGS and the value j is the index of detector in DCTS. The value of d is the distance between dct and current antigen Ag.

1. Equation 6 is used to calculate the distance or affinity between Ag and dct.

$$dct_j.d_i = d_{ij} = d(Ag_i.Sg, dct_j.Sb) \tag{6}$$

2. Equation 7,8 is used to calculate the property of dct's harmmax. The children of the detector dct reproduced are proportion to Harmmax.

$$d_{*j} = \{d_{1j}, d_{2j}, ...d_{Mj}\} \tag{7}$$

$$dct_j.harmmax = \max(d_{*j}) \tag{8}$$

3. According equation 9, if angtigen i cannot detect by any detector in DCTS, antigen i is taken as suspect antigen, which means that antigen i required to be detected over many generations until antigen is is taken as self antigen or nonself antigen. If antigen i can not be detected over maxundetectedCount generations, it is changed to 'self antigen'. Suppose detector dct_x has the max distance with agi, detector dct_x is changed to highest detector.

$$\begin{aligned} &if(!detect(Ag_i, DCTS)) \\ &\begin{cases} Ag_i.state = 'suspect' \\ Ag_i.undetectedCount = Ag_i.undetectedCount + 1 \\ dct_x.state = 'highest' \quad , \exists dct_x, dct_x.harmmax = \max(d_{i*}) \end{cases} \\ &if(Ag_i.undetectedCount > maxundetectedCount) \\ &Ag_i.state = 'self' \end{aligned} \tag{9}$$

4. In equation 10, If detector y detects the antige i, antigen i is changed to nonself antigen and detector y is changed to maturation detector. The detectedAgNum of detector y is increased. $DCTS_{iM}$ is defined as the set of detectors which can detect the antigen i as nonself.

$$\begin{aligned} &if(RMRMatch(Ag_i, dct_y)) \\ &\begin{cases} Ag_i.state = 'nonself' \\ dct_y.state = 'maturation' \\ dct_y.detectedAgNum = dct_y.detectedAgNum + 1 \\ DCTS_{iM} = DCTS_{iM} \cup dct_y \end{cases} \end{aligned} \tag{10}$$

In equation 11, r_{life} is a parameter used to control the lifecycle of maturation detector. Suppose detector dct_m has the max detectedAgNum, dct_m's lifecycle is increased after ag_i is detected. So dct_m can be reserved to detect more similar antigens and the generalization capability of the algorithm is improved. If r_{life} is set to ∞, it will not die.

$$\begin{aligned}
&\exists dct_m \in DCTS_{iM} \\
&dct_m.detectedAgNum = \max(dct_*.detectedAgNum) \\
&AgNum = dct_m.detectedAgNum - dct_m.oldDetectedAgNum \\
&if(AgNum > 0)\{ \\
&dct_m.lifecycle = dct_m.lifecycle + r_{life} * AgNum \\
&dct_m.oldDetectedAgNum = dct_m.detectedAgNum \\
&\}
\end{aligned} \tag{11}$$

5. In equation 12, the lifecycle of the detector will decrease. If lifecycle is equal to 0, it will be removed

$$\begin{aligned}
&\exists dct_m \in DCTS, dct_m.lifecycle = dct_m.lifecycle - 1 \\
&if(dct_m.lifecycle == 0), dct_m.state = 'die'
\end{aligned} \tag{12}$$

2.3 The Detect Process

The algorithm proposed has combined TMA with Affinity Maturation. So it has two detect processes. Some variables are defined in equation 13~16.

$$\forall Ag_{new} \in AGS_{new} \in AGS, Ag_{new}.state = 'new' \tag{13}$$

$$\forall Ag_{suspect} \in AGS_{suspect} \in AGS, Ag_{suspect}.state = 'suspect' \tag{14}$$

$$\forall dct_{highest} \in DCTS_{highest} \in DCTS, dct_{highest}.state = 'highest' \tag{15}$$

$$\forall dct_{maturation} \in DCTS_{marturation} \in DCTS, dct_{highest}.state = 'maturation' \tag{16}$$

1. AGSnew is first detected by DCTSmaturation,called TMADetect. After the process, Angtigen in AGSnew will be split into suspect or nonself antigen. If one antigen cannot detected by any detector, highest detectors will be generated.

2. AGSsuspect is detected by DCTSmaturation and DCTShighest ,called AMDetect. In this process, new detectors are generated from the highest detectors through affinity maturation. Also, new detectors are randomly generated to implement the TMA.

2.4 Implementation of HIDMA-GC Algorithm

The algorithm is shown in fig.3. TMA is used to detect the new antigen and Affinity Maturation is used to detect the suspect antigen. Step 8 is as a part of TMA and used to generation new detectors. The number of detectors generated is $| AGS_{suspect} |$. Step 9-10 is as the process of Affinity Maturation. In step.9, Each detector with higher affinity in $DCTS_{highest}$ reproduces child detectors. The higher affinity the detector has , the higher the number of clones generated[7]. The number of new detectors generated is the value of harmax. In step.10, There is no cross operator but mutate operator according the hypermuate principle. The higher affinity, the smaller mutate rate [7]. So mutate Rate is (the length of dct.agbin-dct.harmmax)/2.

1. For generation g=0 to maxgeneration maxg
2. AGS_{new} =Get all antigens injected in the generation g
3. if($|AGS_{new}|>0$){//TMADetect
4. **Detect**(AGS_{new}, $DCTS_{maturation}$) }// shown in fig.4
5. If($|AGS_{suspect}|>0$){//AMDetect
6. **Detect**($AGS_{suspect}$, $DCTS_{maturation}$ U $DCTS_{highest}$)// shown in fig.4
7. If($|AGS_{suspect}|>0$){
8. Randomly generate new detectors//As a part of TMA
9. clone operator. //As a part of Affinity Maturation
10. hypermuate operator.
11. **setMatchRange**//according equation 2}}
12 if($|AGS_{new}|>0$ or $|AGS_{suspect}|>0$){
//according equation 12
Each detector in $DCTS_{maturation}$ with lifecyle=0 is removed}

Fig. 3. Model of the algorithm

3 Experiments

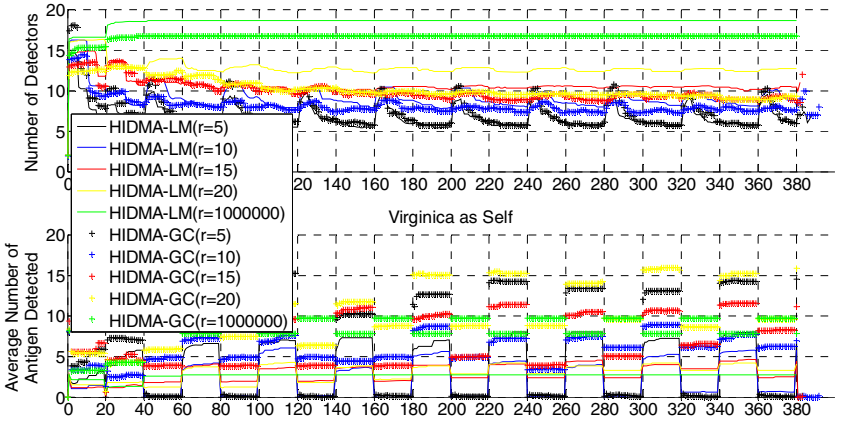
The objective of the experiments is to investigate the generalization capability. Experiments are carried out using the famous benchmark Fisher's Iris Data. Minimal entropy discretization algorithm is used to discretize these data sets [8]. For verifying the adaptive character, nonself data are changed every 20 generations, maxundetectCount=maxg. In the Iris Data, It has 4 attributes and has total 150 examples with three classes: 'Setosa', ' Versicolour', ' Virginica'. Each class has 50 examples. One of the three types of iris is considered as normal data. The other two are considered anomaly and injected into the algorithm in turn and repeatedly. The proposed algorithm HIDMA-GC and HIDMA-LM runs for 10 times especially with different r_{life} which is set to 5, 10, 15, 20, 1000000. The max generation maxg=1000000.

```
Detect(parAGS,parDCTS){
1.      Each dct in parDCTS,Set harmmax=0
2.      For int i=0 to | parAGS |{
3.              Harmmax=0
4.              dctx=null;
5.              DCTS_iM=null;
6.              dcty=null
7.              For int j=0 to | parDCTS |{
8.                      d[i][j]=Distance(parAGS [i], parDCTS [j])
                        //according to equation 8
9.                      If(d[i][j]> parDCTS[j].harmmax){
10.                             parDCTS[j].harmmax = d[i][j]}
                        //according to equation 10
11.                     If(d[i][j]> parDCTS[j].selfmax
                          || d[i][j]< parDCTS[j].selfmim){
12.                             DCTS_iM= DCTS_iM U parDCTS [j]}
                }
13.             If(DCTS_iM==null) {
                        //according to equation 9
14.                     dctx.state='highest'
15.                     parAGS [i].state='suspect'
16.                     parAGS [i].undetectedCount++;
17.                     if(parAGS [i].undetectedCount >maxundetectedCount){
18.                             parAGS [i].state='self'}
                }Else{
                        //according to equation 11
19.                     maxDetectedAgNum=0
20.                     dctm=null
21.                     for int y=0 to |DCTS_iM!{
22.                             dcty=DCTS_iM[y]
23.                             dcty.state='maturation'
24.                             dcty.detectedAgNum= dcty.detectedAgNum+1
25.                             if(maxDetectedAgNum< dcty.detectedAgNum){
26.                                     Dctm=dcty}
27.                             parAGS [i].state='nonself'
                        }
28.                     Dctm.lifecycle= Dctm.lifecycle+rlife*
                                (dctm.detectedAgNum-  dctm.olddetectedAgNum)
29.                     dctm.olddetectedAgNum= dctm.detectedAgNum
                }
    }
```

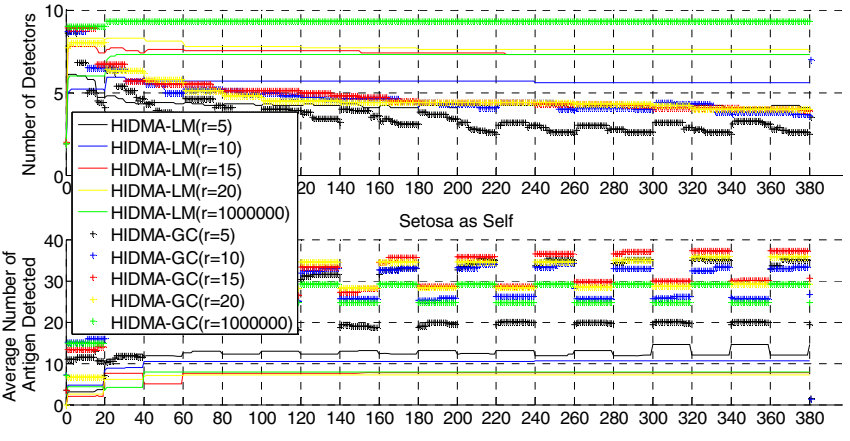
Fig. 4. Detect function

3.1 Generalization Capability

In Fig.5(a), it shows that HIDMA-GC use less detectors in the top figure and detect more antigens in the bottom figure. So HIDMA-GC has more generalization capability than HIDMA-LM. So does in figure b and c.

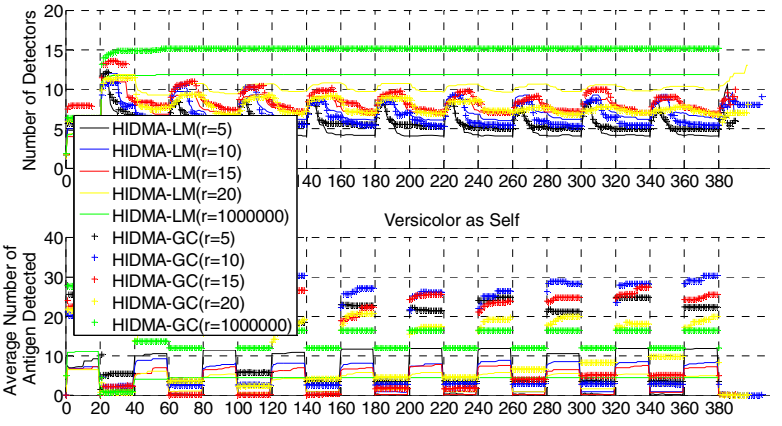


(a)



(b)

Fig. 5. The comparation of algorithm using Iris Data



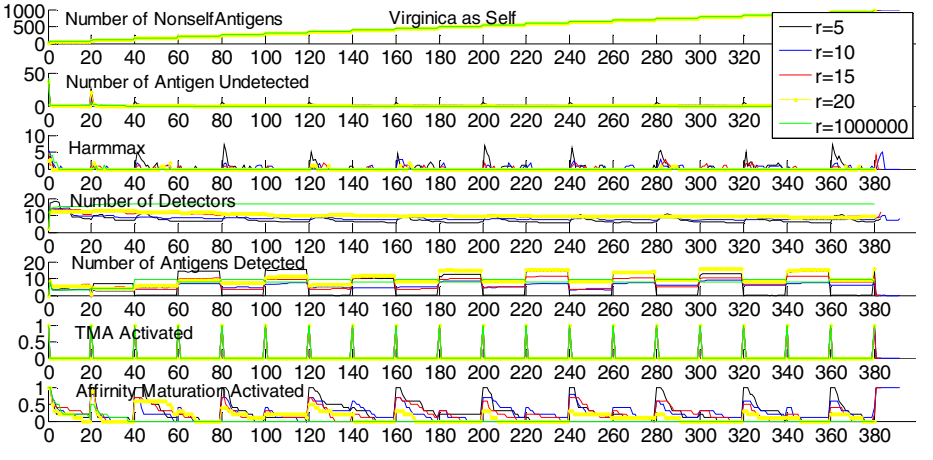
(c)

Fig. 5. (*continued*)

3.2 The Effect of Affinity Maturation and Adaptive Population

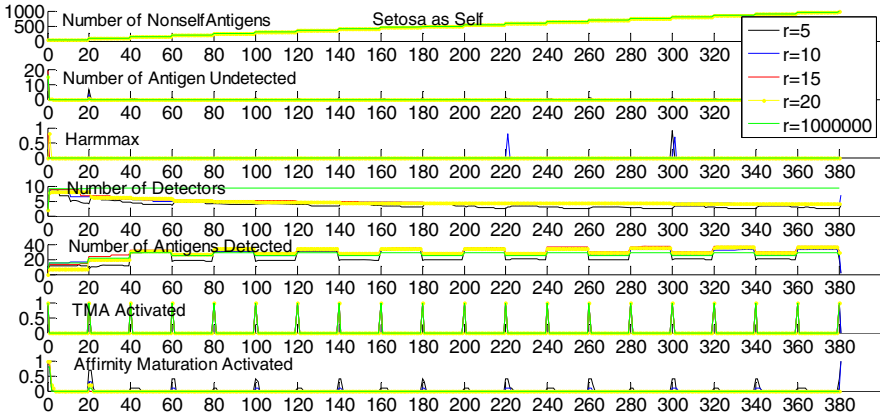
In fig.6(a), the first figure shows the number of antigens changed in every 20 generations. No matter how the antigens change, there are fewer antigens which cannot be detected in the second figure because of the adaptive population of HIDMA shown in the forth figure.

Furthermore, in the algorithm, affinity maturation operator is activated intensely in the seventh figure when the number of detectors in the forth figure is not enough to detect all the antigens in the second figure. In the forth figure, larger the r_{life} is, more stable the population is and less effective Affinity Maturation is in the seven figure. So it is conclude that affinity maturation is the effective mechanism to regulate the population.

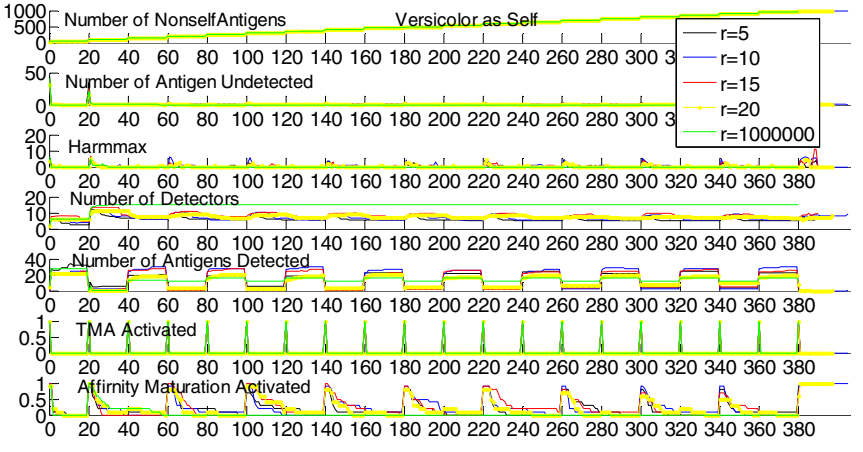


(a)

Fig. 6. Results of HIDM-GC using Iris Data



(b)



(c)

Fig. 6. (*continued*)

4 Conclusion

In this work, to improve the generalization capability and detect more similar patterns, an augmented HIDMA-LG algorithm (HIDMA-GC) is proposed. The results show that the generalization capability is improved and more similar pattern can be detected. It is concluded that affinity maturation is the effective mechanism to regulate the population. Furthermore, with different r_{life} value, the ability to detect common patterns of the detectors is different. The optimized r_{life} value is required to be solved in the future.

Acknowledgement. This work is supported by National Natural Science Foundation of China 71071145,, Zhejiang Provincial Nature Science Foundation Y1110200 , Y6090027, Ministry of Science and Technology project 2009GJC20045. Thanks for the assistance received by using KDD Cup 1999 data set [http://kdd.ics.uci.edu/databases / kddcup99/ kddcup99.html].

References

1. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
2. Hart, E., Timmis, J.: Application areas of AIS: The past, the present and the future. Journal of Applied Soft Computing 8(1), 191–201 (2008)
3. Timmis, J., et al.: An interdisciplinary perspective on artificial immune systems. Evolutionary Intelligence 1(1), 5–26 (2008)
4. Greensmith, J., Aickelin, U., et al.: Information Fusion for Anomaly Detection with the Dendritic Cell Algorithm. Information Fusion 11(1), 21–34 (2010)
5. Hofmeyr, S.A.: An Immunological Model of Distributed Detection and its Application to Computer Security, PhD Dissertation. University of New Mexico (1999)
6. Chen, J., et al.: Hybrid Immune Detector Maturation Algorithm with LifeCycle Model. In: International Conference on Computer Science and its Applications, pp. 213–218 (2009)
7. de Castro, L.N., et al.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation 6(3), 239–251 (2002)
8. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features, `http://robotics.stanford.edu/~ronnyk/disc.ps`

White Box Classification of Dissimilarity Data

Barbara Hammer, Bassam Mokbel,
Frank-Michael Schleif, and Xibin Zhu

CITEC Centre of Excellence, Bielefeld University, 33615 Bielefeld - Germany
{bhammer,bmokbel,fschleif,xzhu}@techfak.uni-bielefeld.de

Abstract. While state-of-the-art classifiers such as support vector machines offer efficient classification for kernel data, they suffer from two drawbacks: the underlying classifier acts as a black box which can hardly be inspected by humans, and non-positive definite Gram matrices require additional preprocessing steps to arrive at a valid kernel. In this approach, we extend prototype-based classification towards general dissimilarity data resulting in a technology which (i) can deal with dissimilarity data characterized by an arbitrary symmetric dissimilarity matrix, (ii) offers intuitive classification in terms of prototypical class representatives, and (iii) leads to state-of-the-art classification results.

Keywords: prototype-based classification, dissimilarity data, GLVQ.

1 Introduction

Machine learning has revolutionized the possibility to deal with large electronic data sets by offering powerful tools to automatically extract a regularity from given data. Rapid developments in modern sensor technologies, dedicated data formats, and data storage continues to pose challenges to the field: on the one hand, data often display a complex structure and a problem-specific dissimilarity measure rather than the Euclidean metric constitutes the interface to the given data. Examples include biological sequences, mass spectra, or metabolic networks, where, e.g. complex alignment techniques, background information, or general information theoretical principles drive the comparison of data points [24,21,14]. These complex dissimilarity measures cannot be computed based on a Euclidean embedding of data, and they often do not even fulfill the properties of a metric. On the other hand, the learning tasks become more and more complex, such that the specific objectives and the relevant information are not clear a priori. This leads to increasingly interactive systems which allow humans to shape the objectives according to human insights and expert knowledge at hand and to extract the relevant information on demand [29,1,7]. This principle requires intuitive interfaces to the machine learning technology which enable humans to interpret the way in which decisions are taken by the system. Hence these requirements lead to the necessity that machine learning techniques provide information which can directly be displayed to the human observer.

Albeit techniques such as the support vector machine (SVM) or Gaussian processes provide efficient state-of-the-art techniques with excellent classification

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 309–321, 2012.

ability, it is often not easy to manually inspect the way in which decisions are taken. Hence, it is hardly possible to visualize its decisions to domain experts in such a way that the results can be interpreted and relevant information can be inferred based thereon. The same argument, although to a lesser degree, is valid for alternatives such as the relevance vector machine or sparse models which, though representing decisions in terms of sparse vectors or class representatives, typically still rely on complex nonlinear combinations of several terms [30,5].

Dissimilarity- or similarity-based machine learning techniques such as nearest neighbor classifiers rely on distances of given data to known labeled data points. Hence it is usually very easy to visualize their decision: the closest data point or a small set of closest points can account for the decision, and this set can directly be inspected by experts in the same way as any data point. Because of this simplicity, (dis)similarity techniques enjoy a large popularity in application domains, whereby the methods range from simple k-nearest neighbor classifiers up to advanced techniques such as *affinity propagation* (AP) which represents a clustering in terms of typical exemplars [16,10].

(Dis)similarity-based techniques can be distinguished by different criteria: (i) The number of data points used to represent the classifier, ranging from dense models such as k-nearest neighbor to sparse representations such as prototype-based methods. To arrive at easily interpretable models, a sparse representation in terms of few data points is necessary. (ii) The degree of supervision, ranging from clustering techniques such as AP to supervised learning. Here we are interested in classification techniques, i.e. supervised learning. (iii) The complexity of the dissimilarity measure the methods can deal with, ranging from vectorial techniques restricted to Euclidean spaces, adaptive techniques which learn the underlying metrics, up to tools which can deal with arbitrary (dis)similarities [27,25]. Typically, Euclidean techniques are well suited for simple classification scenarios, but fail if high-dimensionality or complex structures are encountered.

Learning vector quantization (LVQ) constitutes one of the few methods to infer a sparse representation in terms of prototypes from a given data set in a supervised way [16], such that it offers a good starting point as an intuitive classification technique which decisions can directly be inspected by humans. Albeit original LVQ has been introduced on somewhat heuristic grounds [16], recent developments in this context provide a solid mathematical derivation of its generalization ability and learning dynamics: explicit large margin generalization bounds of LVQ classifiers are available [8,27]; further, the dynamics of LVQ type algorithms can be derived from cost functions which model the classification accuracy referring to the hypothesis margin or a statistical model, for example [27,28]. Interestingly, already the dynamics of simple LVQ as proposed by Kohonen provably leads to surprisingly good generalization curves when investigated in the framework of the theory of online learning [3].

When dealing with modern application scenarios, one of the largest drawbacks of LVQ type classifiers is their dependency on the Euclidean metric. Because of this, LVQ is not suited for complex or heterogeneous data sets where input dimensions have different relevance or where high dimensionality leads to

accumulated noise disrupting the classification. This problem can partially be avoided by metric learning, see e.g. [27], or by kernel variants, see e.g. [25], which turn LVQ classifiers into state-of-the-art techniques e.g. with humanoid robotics or computer vision [9,15]. However, if data are inherently non-Euclidean, these techniques cannot be applied. In modern applications, data are often addressed using dedicated non-Euclidean dissimilarities such as dynamic time warping for time series, alignment for symbolic strings, the compression distance to compare sequences based on an information theoretic ground, and similar [6].

In this contribution, we propose an extension of *generalized LVQ* (GLVQ) [26,27] to general dissimilarity data; GLVQ being a popular LVQ-type algorithm derived from a cost function which is related to the hypothesis margin. This way, the technique becomes directly applicable for data sets which are characterized in terms of a symmetric dissimilarity matrix only. The key ingredient is taken from recent approaches in the unsupervised domain [13,23]: if prototypes are represented implicitly as linear combinations of data in the so-called pseudo-Euclidean embedding or, more generally, a so-called Krein space (see [23, p.77]), the relevant distances of data and prototypes can be computed without an explicit reference to a vectorial representation. This principle holds for every symmetric dissimilarity matrix and thus, allows us to formalize a valid objective of GLVQ for dissimilarity data, which we refer to as relational GLVQ since it deals with data characterized by pairwise relations. Based on this observation, optimization can be performed using gradient techniques. Interestingly, the results are comparable to the state-of-the-art, but GLVQ additionally offers an intuitive interface in terms of prototypes [6].

Due to its dependency on the dissimilarity matrix, relational GLVQ has squared complexity, the computation of the dissimilarities often constituting the bottleneck in applications. By integrating approximation techniques [31], the effort can be reduced to linear time methods. We demonstrate the feasibility of this approach with the popular SWISSPROT protein data base [4].

2 Generalized Learning Vector Quantization

In the classical vectorial setting, data $\boldsymbol{x}^i \in \mathbb{R}^n, i = 1, \ldots, m$, are given. Prototypes $\boldsymbol{w}^j \in \mathbb{R}^n, j = 1, \ldots, k$ decompose data into receptive fields $R(\boldsymbol{w}^j) := \{\boldsymbol{x}^i : \forall k\ d(\boldsymbol{x}^i, \boldsymbol{w}^j) \leq d(\boldsymbol{x}^i, \boldsymbol{w}^k)\}$ based on the squared Euclidean distance $d(\boldsymbol{x}^i, \boldsymbol{w}^j) = \|\boldsymbol{x}^i - \boldsymbol{w}^j\|^2$. The goal of prototype-based machine learning techniques is to find prototypes which represent a given data set as accurately as possible. For supervised learning, data $\boldsymbol{x}^i$ are equipped with class labels $c(\boldsymbol{x}^i) \in \{1, \ldots, L\}$. Similarly, every prototype is equipped with a priorly fixed label $c(\boldsymbol{w}^j)$. A data point is classified according to the class of its closest prototype. The classification error of this mapping is given by the term $\sum_j \sum_{\boldsymbol{x}^i \in R(\boldsymbol{w}^j)} \delta(c(\boldsymbol{x}^i) \neq c(\boldsymbol{w}^j))$ with the delta function δ. This cost function cannot easily be optimized explicitly due to vanishing gradients and discontinuities. Therefore, LVQ relies on a reasonable heuristic by performing Hebbian updates of the prototypes, given a data point [16]. Recent alternatives derive similar update rules from explicit

cost functions which are related to the classification error, but display better numerical properties such that efficient optimization is possible [27,26,28].

Generalized LVQ [26] is derived from a cost function which can be related to the generalization ability of LVQ classifiers [27]:

$$E_{\mathrm{GLVQ}} = \sum_i \Phi\left(\frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}\right)$$

where Φ is a differentiable monotonic function such as the hyperbolic tangent, and $\boldsymbol{w}^+(\boldsymbol{x}^i)$ refers to the prototype closest to $\boldsymbol{x}^i$ with the same label as $\boldsymbol{x}^i$, $\boldsymbol{w}^-(\boldsymbol{x}^i)$ refers to the closest prototype with a different label. Hence, the contribution of a data point to these costs is small if and only if the closest correct prototype is much closer than the closest incorrect one, resulting in a correct classification and, at the same time, aiming at a large hypothesis margin, i.e., a good generalization ability.

A learning algorithm can be derived thereof by means of standard gradient techniques. After presenting data point $\boldsymbol{x}^i$, its closest correct and wrong prototype, respectively, are adapted according to the rules:

$$\begin{aligned}
\Delta\boldsymbol{w}^+(\boldsymbol{x}^i) &\sim -\Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^+(\boldsymbol{x}^i) \cdot \nabla_{\boldsymbol{w}^+(\boldsymbol{x}^i)} d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) \\
\Delta\boldsymbol{w}^-(\boldsymbol{x}^i) &\sim \Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^-(\boldsymbol{x}^i) \cdot \nabla_{\boldsymbol{w}^-(\boldsymbol{x}^i)} d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))
\end{aligned}$$

where

$$\begin{aligned}
\mu(\boldsymbol{x}^i) &= \frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))} \;, \\
\mu^+(\boldsymbol{x}^i) &= \frac{2 \cdot d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{(d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))^2} \;, \\
\mu^-(\boldsymbol{x}^i) &= \frac{2 \cdot d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)}{(d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))^2} \;.
\end{aligned}$$

For the squared Euclidean norm, the derivative yields $\nabla_{\boldsymbol{w}^j} d(\boldsymbol{x}^i, \boldsymbol{w}^j) = -2(\boldsymbol{x}^i - \boldsymbol{w}^j)$, leading to Hebbian update rules of the prototypes according to the class information. GLVQ constitutes one particularly efficient method to adapt the prototypes according to a given labeled data set. Alternatives can be derived based on a labeled Gaussian mixture model, see e.g. [28]. Since the latter can be highly sensitive to model meta-parameters [3], we focus on GLVQ.

3 Dissimilarity Data

We assume that data $\boldsymbol{x}^i$ are characterized by pairwise dissimilarities $d_{ij} = d(\boldsymbol{x}^i, \boldsymbol{x}^j)$. D refers to the corresponding dissimilarity matrix. We assume symmetry $d_{ij} = d_{ji}$ and zero diagonal $d_{ii} = 0$. However, D need not be Euclidean, i.e. it is not guaranteed that vectors $(\boldsymbol{x}^i, \boldsymbol{x}^j)$ can be found with $d_{ij} = \|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2$. For every such dissimilarity matrix D, an associated similarity matrix is induced by

$S = -JDJ/2$ where $J = (I - \mathbf{1}\mathbf{1}^t/n)$ with identity matrix I and vector of ones $\mathbf{1}$. D is Euclidean if and only if S is positive semidefinite (*pdf* in the following). In general, p eigenvectors of S have positive eigenvalues and q have negative eigenvalues, $(p, q, n - p - q)$ is referred to as the signature.

For kernel methods such as SVM, a correction of the matrix S is necessary to guarantee pdf. Two techniques are very popular: the spectrum of the matrix S is changed, possible operations being *clip* (negative eigenvalues are set to 0), *flip* (absolute values are taken), or *shift* (a summand is added to all eigenvalues). Interestingly, some operations such as shift do not affect the location of local optima of important cost functions such as the quantization error [18], albeit the transformation can severely affect the performance of optimization algorithms [13]. As an alternative, data points can be treated as vectors which coefficients are given by the pairwise similarity. These vectors can be processed using standard kernels. In [6] an extensive comparison of these preprocessing methods in connection to SVM is performed for a variety of benchmarks.

Alternatively, one can directly embed data in the pseudo-Euclidean vector space determined by the eigenvector decomposition of S. A symmetric bilinear form is induced by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{p,q} = \boldsymbol{x}^t I_{p,q} \boldsymbol{y}$ where $I_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. Taking the eigenvectors of S together with the square root of the absolute value of the eigenvalues, we obtain vectors $\boldsymbol{x}^i$ in pseudo-Euclidean space such that $d_{ij} = \langle \boldsymbol{x}^i - \boldsymbol{x}^j, \boldsymbol{x}^i - \boldsymbol{x}^j \rangle_{p,q}$ holds for every pair of data points. If the number of data is not limited a priori, a generalization of this concept to Krein spaces with according decomposition is possible [23, p.77].

Vector operations can be directly transferred to the pseudo-Euclidean space, i.e. we can define prototypes as linear combinations of data in this space. Hence, we can perform techniques such as GLVQ explicitly in pseudo-Euclidean space since it relies on vector operations only. One problem of this explicit transfer is given by the computational complexity of the embedding which is $\mathcal{O}(n^3)$, and, further, the fact that out-of-sample extensions to new data points characterized by pairwise dissimilarities are not immediate. Because of this fact, we are interested in efficient techniques which implicitly refer to this embedding only. As a side effect, such algorithms are invariant to coordinate transforms in pseudo-Euclidean space. The key assumption is to restrict prototype positions to linear combinations of data points of the form

$$\boldsymbol{w}^j = \sum_i \alpha_{ji} \boldsymbol{x}^i \text{ with } \sum_i \alpha_{ji} = 1 \,.$$

Since prototypes are located at representative points in the data space, this is reasonable. Then dissimilarities can be computed implicitly by means of

$$d(\boldsymbol{x}^i, \boldsymbol{w}^j) = [D \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j$$

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\boldsymbol{w}^j$ implicitly, as shown in [13].

This observation constitutes the key to transfer GLVQ to relational data. Prototype $\boldsymbol{w}^j$ is represented implicitly by means of the coefficient vectors α_j

and distances are computed by means of these coefficients. The corresponding cost function of relational GLVQ (RGLVQ) becomes:

$$E_{\mathrm{RGLVQ}} = \sum_i \Phi\left(\frac{[D\alpha^+]_i - \frac{1}{2}\cdot(\alpha^+)^t D\alpha^+ - [D\alpha^-]_i + \frac{1}{2}\cdot(\alpha^-)^t D\alpha^-}{[D\alpha^+]_i - \frac{1}{2}\cdot(\alpha^+)^t D\alpha^+ + [D\alpha^-]_i - \frac{1}{2}\cdot(\alpha^-)^t D\alpha^-}\right),$$

where as before the closest correct and wrong prototype are referred to, now in terms of the corresponding coefficients α^+ and α^-, respectively. A simple stochastic gradient descent leads to adaptation rules for the coefficients α^+ and α^- in relational GLVQ: component k of these vectors is adapted as

$$\Delta\alpha_k^+ \sim -\Phi'(\mu(\boldsymbol{x}^i))\cdot\mu^+(\boldsymbol{x}^i)\cdot\frac{\partial\left([D\alpha^+]_i - \frac{1}{2}\cdot(\alpha^+)^t D\alpha^+\right)}{\partial\alpha_k^+}$$

$$\Delta\alpha_k^- \sim \Phi'(\mu(\boldsymbol{x}^i))\cdot\mu^-(\boldsymbol{x}^i)\cdot\frac{\partial\left([D\alpha^-]_i - \frac{1}{2}\cdot(\alpha^-)^t D\alpha^-\right)}{\partial\alpha_k^-}$$

where $\mu(\boldsymbol{x}^i)$, $\mu^+(\boldsymbol{x}^i)$, and $\mu^-(\boldsymbol{x}^i)$ are as above. The partial derivative yields

$$\frac{\partial\left([D\alpha_j]_i - \frac{1}{2}\cdot\alpha_j^t D\alpha_j\right)}{\partial\alpha_{jk}} = d_{ik} - \sum_l d_{lk}\alpha_{jl}$$

Naturally, alternative gradient techniques can be used. After every adaptation step, normalization takes place to guarantee $\sum_i \alpha_{ji} = 1$. We also restrict the possible prototype positions to the convex hull of the data by enforcing all $\alpha_{ji} \geq 0$ in every iteration. This way, a learning algorithm which adapts prototypes in a supervised manner similar to GLVQ is given for general dissimilarity data, whereby prototypes are implicitly embedded in pseudo-Euclidean space. The prototypes are initialized as random vectors corresponding to random values α_{ij} which sum to one. It is possible to take class information into account by setting all α_{ij} to zero which do not correspond to the class of the prototype. Out-of-sample extension of the classification to new data is possible based on the following observation [13]: for a novel data point $\boldsymbol{x}$ characterized by its pairwise dissimilarities $D(\boldsymbol{x})$ to the data used for training, the dissimilarity of $\boldsymbol{x}$ to a prototype α_j is $d(\boldsymbol{x}, \boldsymbol{w}^j) = D(\boldsymbol{x})^t \cdot \alpha_j - \frac{1}{2}\cdot\alpha_j^t D\alpha_j$.

Interpretability and Speed-Up

Relational GLVQ extends GLVQ to general dissimilarity data. Unlike Euclidean GLVQ, it represents prototypes indirectly by means of coefficient vectors which are not directly interpretable since they correspond to typical positions in pseudo-Euclidean space. However, because of their representative character, we can approximate these positions in pseudo-Euclidean space by their respective closest exemplars, i.e. data points originally contained in the training set. Unlike prototypes, these exemplars can be directly inspected. We refer to such an approximation as K-approximation if a prototype is substituted by its K closest exemplars,

the latter being directly accessible to humans. We will see in experiments that the resulting classification accuracy is still quite good for small values K in $\{1, \ldots, 5\}$, and we present an example showing the interpretability of the result. We refer to results obtained by a K-approximation by the subscript RGLVQ_K.

In addition, RGLVQ (just as SVM) depends on the full dissimilarity matrix and thus displays quadratic computational and memory complexity. Depending on the chosen dissimilarity, the main computational bottleneck is given by the calculation of the dissimilarity matrix itself. Alignment of biological sequences, for example, is quadratic in the sequence length (linear, if approximations such as FASTA are used), such that a computation of the full dissimilarities for about 11,000 data points (the size of the Swissprot data set as considered below) would already lead to a computation time of more than eight days (Intel Xeon Quad-Core 2.5 GHz, alignment done by Smith-Waterman) and a storage requirement of about 500 Megabyte, assuming double precision. The Nyström approximation as introduced in [31] allows for an efficient approximation of a kernel matrix by a low rank matrix. This approximation can directly be transferred to dissimilarity data. The basic principle is to pick M representative landmarks which induce the rectangular sub-matrix $D_{M,m}$ of dissimilarities of data points and landmarks. This matrix is of linear size, assuming M is fixed. The full matrix can be approximated in an optimum way in the form $D \approx D_{M,m}^t D_{M,M}^{-1} D_{M,m}$ where $D_{M,M}$ is the rectangular sub-matrix of D. The computation of $D_{M,M}^{-1}$ is $\mathcal{O}(M^3)$ instead of $\mathcal{O}(m^2)$ for the full matrix D. The resulting approximation is exact if M corresponds to the rank of D. For 10% landmarks, computing $D_{M,M}$ instead of D leads to a speed-up factor of 50, i.e. given $11,000$ sequences, it can be computed in less than two hours instead of eight days. The storage capacity reduces to 4.5 Megabytes as compared to 500 Megabytes in this case. Note that the Nyström approximation can be directly integrated into the distance computation of relational GLVQ in such a way that the overall training complexity is linear instead of quadratic. We refer to results obtained by a Nyström approximation by the superscript RGLVQ^ν. We use 10% landmarks by default.

4 Experiments

We evaluate relational GLVQ for several benchmark data sets characterized by pairwise dissimilarities. These data sets have been used extensively in [6] to evaluate SVM classifiers for general dissimilarity data. Since SVM requires a pdf matrix, appropriate preprocessing has been done in [6]: flip, clip, shift, and vectorial representation together with the linear and Gaussian kernel, respectively, is used in conjunction with a standard SVM. In addition, we consider a few benchmarks from the biomedical domain. The data sets are as follows:

1. Amazon47 consisting of 204 data points from 47 classes, representing books and their similarity based on customer preferences. The similarity matrix S was symmetrized and transferred by means of $D = \exp(-S)$, see [18].
2. Aural Sonar consists of 100 signals with two classes (target of interest/clutter), representing sonar signals with dissimilarity measures according to an ad hoc classification of humans.

3. The Cat Cortex data set consists of 65 data points from 5 classes. The data originate from anatomic studies of cats' brains. The dissimilarity matrix displays the connection strength between 65 cortical areas. A preprocessed version as presented in [12] was used.
4. The Copenhagen Chromosomes data set constitutes a benchmark from cytogenetics [19]. A set of 4,200 human chromosomes from 21 classes (the autosomal chromosomes) are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings are compared using edit distance [22].
5. Face Recognition consists of 945 samples with 139 classes, representing faces of people, compared by the cosine similarity.
6. Patrol consists of 241 data points from 8 classes, corresponding to seven patrol units (and non-existing persons, respectively). Similarities are based on clusters named by people.
7. Protein consists of 213 data from 4 classes, representing globin proteins compared by an evolutionary dissimilarity measure.
8. The SwissProt data set consists of 10,988 samples of protein sequences in 32 classes taken as a subset from the SwissProt database [4]. The considered subset of the SwissProt database refers to the release 37 mimicking the setting as proposed in [17]. The full dataset consists of 77,977 protein sequences. The 32 most common classes such as Globin, Cytochrome a, Cytochrome b, Tubulin, Protein kinase st, etc. provided by the Prosite labeling [11] where taken leading to 10,988 sequences. We calculate a similarity matrix based on a 10% Nyström approximation. These sequences are compared using exact Smith-Waterman. This database is the standard source for identifying and analyzing protein measurements such that an automated sparse classification technique would be very desirable. A detailed analysis of the prototypes of the different protein sequences opens the way towards an inspection of typical biochemical characteristics of the represented data.
9. The Vibrio data set consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra contain approx. 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The mass spectra are preprocessed with a standard workflow using the BioTyper software [21]. As usual, mass spectra display strong functional characteristics due to the dependency of subsequent masses, such that problem adapted similarities such as described in [2,21] are beneficial. In our case, similarities are calculated using a specific similarity measure as provided by the BioTyper software[21]. The Vibrio similarity matrix S has a maximum score of 3. The corresponding dissimilarity matrix is obtained as $D = 3 - S$.
10. Voting contains 435 samples in 2 classes, representing categorical data compared based on the value difference metric.

As pointed out in [6], these matrices cover a diverse range of different characteristics such that they constitute a well suited test bench to evaluate the performance of algorithms for similarities/dissimilarities. In addition, benchmarks from the biomedical domain have been added, which constitute interesting applications

per se. All datasets are non-Euclidean, the signatures can be found in Tab. 1. For every data set, we used a number of prototypes which corresponds to the number of classes, representing every class by only few prototypes relating to the choices as taken in [13], see Tab. 1. The evaluation of the results is done by means of the classification accuracy obtained on the test set in a ten-fold repeated cross-validation with ten repeats (two-fold cross-validation for Swissprot). Classes are reasonably balanced in the data sets, the largest observed difference in class sizes being 26% of the total number of data points. For this reason, and to maintain comparability with [6], we consider the classification accuracy to be an appropriate evaluation measure. For comparison, we report the results of a SVM after appropriate preprocessing of the dissimilarity matrix to guarantee a pdf kernel [6]. In addition, we report the results of a powerful unsupervised exemplar-based technique, AP [10], which optimizes the quantization error for arbitrary similarity matrices based on a message passing algorithm for a corresponding factor graph representation of the cost function. Here the classification is obtained by posterior labeling. For relational GLVQ, we train the standard technique for the full dissimilarity matrix, and we compare the result to the sparse models obtained by a K-approximation with $K \in \{1, 3\}$ and a Nyström approximation of the dissimilarity matrix using 10% of the training data. The mean classification accuracies are reported in Tab. 2 and Tab. 1.

Interestingly, in all cases but one (the almost Euclidean data set proteins), results are comparable to SVM taking the respective best preprocessing as reported in [6]. Unlike SVM, relational GLVQ makes this preprocessing superfluous. In contrast, SVM requires preprocessing to guarantee pdf, leading to divergence or

Table 1. Results of prototype-based classification by means of relational GLVQ in comparison to SVM with pdf preprocessing and an SMO implementation and in comparison to AP with posterior labeling for diverse dissimilarity data sets. The classification accuracy obtained in a repeated ten-fold cross-validation with ten repeats is reported (only two-fold for Swissprot), the standard deviation is given in parenthesis. SVM results marked with * are taken from [6]. The number of prototypes used for RGLVQ and AP as well as the characteristic of the dissimilarity matrix are included. For SVM, the respective best and worst result using the different preprocessing mechanisms clip, flip, shift, and similarities as features with linear and Gaussian kernel are reported.

	RGLVQ	**AP**	**SVM**	Signature	# Prototypes
Aural Sonar	88.4 (1.6)	68.5 (4.0)	87.00 - 85.75*	(54,45,1)	10
Amazon47	81.0 (1.4)	75.9 (0.9)	82.20 - 74.4	(136,68,0)	94
Cat Cortex	93.0 (1.0)	80.4 (2.9)	95.00 - 72.00	(41,23,1)	12
Chromosome	92.7 (0.2)	89.5 (0.6)	95.10 - 92.20	(1951,2206,43)	63
Face rec.	96.4 (0.2)	95.1 (0.3)	96.08 - 95.71*	(311,310,324)	139
Patrol	84.1 (1.4)	58.1 (1.6)	61.25 - 57.81*	(173,67,1)	24
Protein	92.4 (1.9)	77.1 (1.0)	98.84 - 97.56*	(218,4,4)	20
SwissProt	81.6 (0.1)	82.6 (0.3)	82.10 - 78.00	(8488,2500,0)	64
Vibrio	100 (0.0)	99.0 (0.0)	100	(573,527,0)	49
Voting	94.6 (0.5)	93.5 (0.5)	95.11 - 94.48*	(105,235,95)	20

Table 2. Results of the relational GLVQ obtained in a repeated ten-fold cross-validation using the full dissimilarity matrix and prototype representation and approximations of the matrix by means of Nyström and approximation of the prototype vectors by means of K-approximations, respectively.

	RGLVQ	**RGLVQ$_1$**	**RGLVQ$_3$**	**RGLVQ$^\nu$**	**RGLVQ$_1^\nu$**	**RGLVQ$_3^\nu$**
Aural Sonar	88.4 (1.6)	78.7 (2.7)	86.4 (2.7)	86.4 (0.8)	79.7 (2.6)	84.3 (2.6)
Amazon47	81.0 (1.4)	67.5 (1.4)	77.2 (1.0)	81.4 (1.1)	66.2 (2.6)	77.7 (1.2)
Cat Cortex	93.0 (1.0)	81.8 (3.5)	89.6 (2.9)	92.2 (2.3)	79.8 (5.5)	89.5 (2.8)
Chromosome	92.7 (0.2)	90.2 (0.0)	91.2 (0.2)	78.2 (0.4)	84.4 (0.4)	86.3 (0.2)
Face rec.	96.4 (0.2)	96.8 (0.2)	96.8 (0.1)	96.4 (0.2)	96.6 (0.3)	96.7 (0.2)
Patrol	84.1 (1.4)	51.0 (2.0)	69.0 (2.5)	85.6 (1.5)	52.7 (2.3)	72.0 (3.7)
Protein	92.4 (1.9)	69.6 (1.7)	79.4 (2.9)	55.8 (2.8)	64.1 (2.1)	54.9 (1.1)
Vibrio	100 (0.0)	99.0 (0.1)	99.0 (0)	99.2 (0.1)	99.9 (0.0)	100 (0.0)
Voting	94.6 (0.5)	93.7 (0.5)	94.7 (0.6)	90.5 (0.3)	89.5 (0.9)	89.6 (0.9)

very bad classification accuracy otherwise. Further, different preprocessing can lead to very diverse accuracy as shown in Tab. 1, no single preprocessing being universally suited for all data sets. Thus, these results seem to substantiate the finding of [18] that preprocessing of a non pdf Gram matrix can influence the classification accuracy. Further, an improvement of the classification accuracy as compared to the state-of-the-art unsupervised prototype-based technique AP (using the same number of prototypes) can be observed, which is statistically significant in all cases (according to a two-sided t-test with a 5% significance level). This shows the benefits of including supervision in the training objective if classification is the goal.

Unlike for SVM which is based on support vectors in the data set, solutions are represented as typical prototypes. Similar to AP, these prototypes can be approximated by K nearest exemplars representing the classification explicitly in terms of few data points instead of prototypes. See Fig. 1 for an inspection of a typical exemplar for the Vibrio data set. As can be seen from Tab. 2, a 3-approximation leads to a loss in accuracy of more than 5% in only two cases. Interestingly, a 3-approximation of a prototype-based classifier for the Swissprot benchmark even leads to an increase of the accuracy from 81.6 to 84.0.

As a further demonstration, we show the result of RGLVQ trained to classify 84 e-books from to 4 different authors; data are taken from the *Project Gutenberg* (www.gutenberg.org). One prototype per class is used with 3-approximation for visual inspection. Data are compared by the normalized compression distance. In Fig. 2, books and representative exemplars found by RGLVQ$_3$ are displayed in 2D using *t-distributed stochastic neighbor embedding* (t-SNE) [20]. While SVM leads to a classification accuracy of more than 95% (like RGLVQ), it picks almost all data points as support vectors, i.e. no direct interpretation is possible.

The Nyström approximation offers a linear time and space approximation of relational GLVQ performed on the full matrix. The decrease in accuracy due to this approximation is documented in Tab. 2 for all except the Swissprot data set – since the computation of the full dissimilarity matrix for the Swissprot

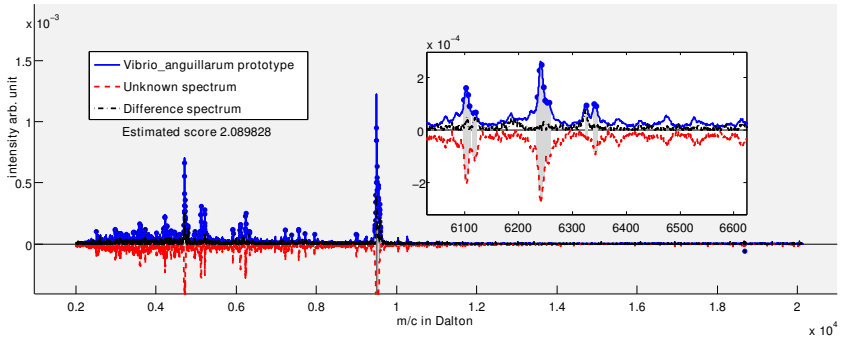


Fig. 1. White box analysis of RGLVQ. The prototype (straight line) represents the class of the test spectrum (dashed line). The prototype is labeled as *Vibrio Anguillarum*. It shows high symmetry to the test spectrum and the similarity of matched peaks (zoom in) highlights good agreement by bright gray shades, indicating the local error of the match. The prototype model allows direct identification and scoring of matched and unmatched peaks, which can be assigned to its mass to charge (m/c) positions, for further biochemical analysis.

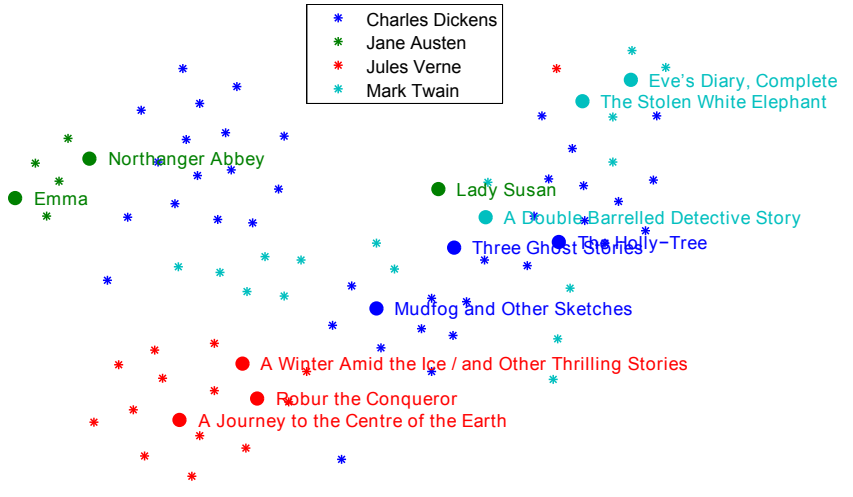


Fig. 2. Visualization of e-books and typical exemplars found by RGLVQ_3

data set would require more than 8 days on a standard PC, we used a Nyström approximation right from the beginning for Swissprot. The quality of the approximation depends on the rank of the dissimilarity matrix. Thus, the results differ a lot depending on the characteristics of the eigenvalue spectrum for the data. Interestingly, it seems possible in more than half of the cases to substitute full relational GLVQ by this linear complexity approximation without much loss of accuracy.

5 Conclusions

We have presented an extension of generalized learning vector quantization to non-Euclidean data sets characterized by symmetric pairwise dissimilarities by means of an implicit embedding in the pseudo-Euclidean space and a corresponding extension of the cost function of GLVQ to this setting. As a result, a very powerful learning algorithm can be derived which, in most cases, achieves results which are comparable to SVM but without the necessity of according preprocessing and with direct interpretability of the classification in terms of the prototypes and their corresponding exemplars in a K-approximation. As a first step to an efficient linear approximation, the Nyström technique has been tested leading to promising results in a number of benchmarks, particularly making the technology feasible for relevant large databases such as the Swissprot data base.

Acknowledgement. Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative and from the "German Science Foundation (DFG)" under grant number HA-2719/4-1 is gratefully acknowledged. We would like to thank Dr. Markus Kostrzewa and Dr. Thomas Maier for providing the Vibrio data set and expertise regarding the biotyping approach and Dr. Katrin Sparbier for discussions about the SwissProt data.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Barbuddhe, S.B., Maier, T., Schwarz, G., Kostrzewa, M., Hof, H., Domann, E., Chakraborty, T., Hain, T.: Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry. Applied and Environmental Microbiology 74(17), 5402–5407 (2008)
3. Biehl, M., Ghosh, A., Hammer, B.: Dynamics and generalization ability of LVQ algorithms. J. Machine Learning Res. 8, 323–360 (2007)
4. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. 31, 365–370 (2003)
5. Chan, A., Vasconcelos, N., Lanckriet, G.: Direct Convex Relaxations of Sparse SVM. In: Proc. of ICML 2007 (2007)
6. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based Classification: Concepts and Algorithms. J. of Machine Learning Res. 10, 747–776 (2009)
7. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
8. Crammer, K., Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin Analysis of the LVQ Algorithm. In: Proc. of NIPS 2002 (2003)
9. Denecke, A., Wersing, H., Steil, J.J., Koerner, E.: Online Figure-Ground Segmentation with Adaptive Metrics in Generalized LVQ. Neurocomputing 72(7-9), 1470–1482 (2009)

10. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
11. Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R.D., Bairoch, A.: ExPASy: the proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Res. 31, 3784–3788 (2003)
12. Haasdonk, B., Bahlmann, C.: Learning with Distance Substitution Kernels. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 220–227. Springer, Heidelberg (2004)
13. Hammer, B., Hasenfuss, A.: Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9), 2229–2284 (2010)
14. Ingram, P.J., Stumpf, M.P.H., Stark, J.: Network motifs: structure does not determine function. BMC Genomics 7, 108 (2006)
15. Kietzmann, T., Lange, S., Riedmiller, M.: Incremental GRLVQ: Learning Relevant Features for 3D Object Recognition. Neurocomputing 71(13-15), 2868–2879 (2008)
16. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, New York (2001)
17. Kohonen, T., Somervuo, P.: How to make large self-organizing maps for nonvectorial data. Neural Networks 15(8-9), 945–952 (2002)
18. Laub, J., Roth, V., Buhmann, J.M., Müller, K.-R.: On the information and representation of non-Euclidean pairwise data. Pattern Recognition 39, 1815–1826 (2006)
19. Lundsteen, C., Phillip, J., Granum, E.: Quantitative analysis of 6985 digitized trypsin g-banded human metaphase chromosomes. Clinical Genetics 18, 355–370 (1980)
20. van der Maaten, L.J.P., Hinton, G.E.: Visualizing high-dimensional data using t-sne. J. of Machine Learning Res. 9, 2579–2605 (2008)
21. Maier, T., Klebel, S., Renner, U., Kostrzewa, M.: Fast and reliable maldi-tof ms–based microorganism identification. Nature Methods (3) (2006)
22. Neuhaus, M., Bunke, H.: Edit distance based kernel functions for structural pattern classification. Pattern Recognition 39(10), 1852–1863 (2006)
23. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
24. Penner, O., Grassberger, P., Paczuski, M.: Sequence Alignment, Mutual Information, and Dissimilarity Measures for Constructing Phylogenies. PLoS ONE 6(1) (2011)
25. Qin, A.K., Suganthan, P.N.: A novel kernel prototype-based learning algorithm. In: Proc. of ICPR 2004, pp. 621–624 (2004)
26. Sato, A., Yamada, K.: Generalized learning vector quantization. In: Mozer, M.C., Touretzky, D.S., Hasselmo, M.E. (eds.) Proc. of NIPS 1995, pp. 423–429. MIT Press, Cambridge (1996)
27. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization. Neural Computation 21(12), 3532–3561 (2009)
28. Seo, S., Obermayer, K.: Soft learning vector quantization. Neural Computation 15(7), 1589–1604 (2003)
29. Thomas, J.J., Cook, K.A.: A Visual Analytics Agenda. IEEE Trans. on Computer Graphics and Applications 26(1), 12–19 (2006)
30. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. J. of Machine Learning Res. 1, 211–244 (2001)
31. Williams, C., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Proc. of NIPS 2000, pp. 682–688. MIT Press (2001)

On Ensemble Classifiers for Nonintrusive Appliance Load Monitoring

Oliver Kramer, O. Wilken, P. Beenken, A. Hein, A. Hüwel, T. Klingenberg, C. Meinecke, T. Raabe, and M. Sonnenschein

OFFIS Institute for Information Technology,
Escherweg 2, 26111 Oldenburg, Germany

Abstract. In this work we employ ensemble classifiers for the problem of nonintrusive appliance load monitoring. In practical scenarios the question arises how to efficiently and automatically learn statistical models for appliance recognition, which is an important step for various problems in process recognition, healthcare, and energy consulting. This work is an application study that analyzes multi-class support vector machines (SVMs), and K-nearest neighbors (KNN) in the problem domain of automatically recognizing appliances. By combining two types of classifiers with varying parameterizations to ensembles, we reduce the classification error, and increase the robustness of the classifier. In the experimental part we consider a field study with household appliances, and compare the classifiers w.r.t. various training set and neighborhood sizes. It turns out that the ensembles belong to the best classifiers in all training set scenarios.

Keywords: Ensemble classifiers, support vector machines, k-nearest neighbors, appliance recognition, NIALM.

1 Introduction

Nonintrusive appliance load monitoring (NIALM, cf. [9]) is the recognition of changes in voltage and current in households for recognition of appliances, and energy consumption. In a smart grid, NIALM can be used for a variety of problem classes. Three examples are:

- Energy management and consulting. The improvement of energy-efficiency for everyday processes is an important task. But it affords the recognition of usage habits of appliances, e.g., to answers questions like *which* appliance is used *when* and *how often*.
- Assistance systems for humans in healthcare scenarios. Monitoring of everyday activities of old needy or disabled humans via the usage of appliances allows the recognition of alert states, and emergency situations.
- Load forecasting. Load forecasting of appliances allows to balance energy systems. Balancing authorities have to consider produced and consumed energy, in particular in distributed smart grid scenarios with volatile renewable energy resources. Appliance recognition is the first step of a many load forecasting systems.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 322–331, 2012.

It is well-known that no superior classifier for every recognition task exists. Ensemble classifiers solve this problem taking into account the classification results of more than one classifier. The hybridization of techniques from computational intelligence turns out to be a very successful strategy in real-world problems [1,4,5], e.g., in optimization (hybrid meta-heuristics), or in classification (ensemble classifiers). In this work we introduce SVM- and KNN-ensemble classifiers for the NIALM problem. The ensembles are hybrids of *local* nearest neighbor classifiers that are based on averaging labels in the neighborhood of unknown patterns, and the *global* SVMs that make use of separating hyperplanes. Due to their specialization the hybridization of both methods seems to be appealing from a theoretical perspective. Taking advantage of both worlds becomes very relevant for practical scenarios, in particular in appliance recognition, where training set sizes and pattern dimensionalities might vary significantly. Training sets are often very limited due to time and budget constraints, and are at the same time not balanced (different number of patterns for different classes). The following research questions arise: (1) What is the classification performance of SVMs and KNN in load-based appliance recognition tasks for our electricity feature set, (2) do ensemble classifiers improve the classification results in terms of accuracy and robustness, and (3) what is the size of a minimal training set for classification?

In Section 2 we review related work in NIALM, and ensemble classifiers. In Section 3 we describe the data acquisition process, and the feature preprocessing steps. Section 4 introduces the ensemble classifiers, starting with a short introduction to the basic classifiers SVM, and KNN. In Section 5 we present an experimental study on a real-world data set of appliance load measurements. Last, Section 6 summarizes the results, and gives an overview of prospective future work.

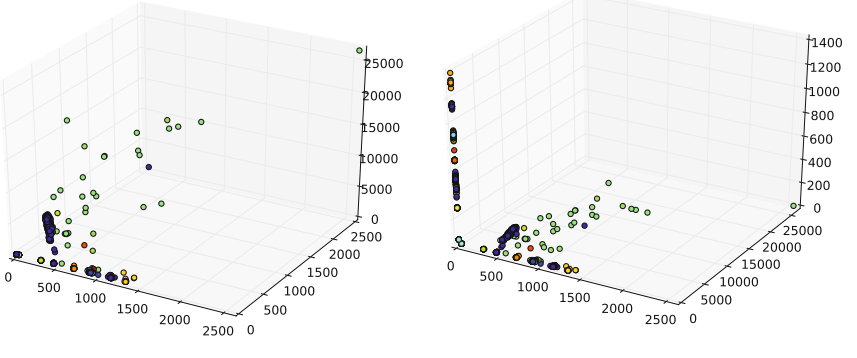
2 Related Work

NIALM. Nonintrusive load monitoring of appliances has a long tradition since the mid-nineties, see Hart [9], who introduced an approach able to recognize appliances as of 150 Watt considering a continuous signal with 1 Hz sample rate. Patel *et al.* [13] employed SVMs for the recognition of 19 appliances, and achieved an accuracy rate of $85-90\%$. Recently, Lin and Tsai [11] proposed a nonintrusive load monitoring system based on hierarchical SVMs decomposing the multi-class problem into a series of binary classification problems. The approach is based on transient features from electricity waveforms. Further classification methods have been employed in the past. Chang *et al.* [3] apply back propagation, and learning vector quantization to load monitoring. The selection of appropriate features has an important part to play for successful recognition processes. Evolutionary approaches have been employed for feature selection [2] in load monitoring.

Ensemble Classifiers. Ensemble classifiers have been proposed to improve the performance of classifiers by combing multiple predictions [12,16]. Homogeneous ensemble classifiers consist of the same type of classifier with varying parameterizations or training data, while heterogenous ensemble classifiers combine different learning algorithms (cf. [15]). Boosting is the technique to aggregate multiple weak learners to a strong classifier (cf. [10]). Ensemble classifiers have successfully been applied in various applications, e.g., text classification [7]. To the best of our knowledge, ensemble classifiers have not been applied to NIALM problems yet, although they appear to be very attractive in scenarios of varying training sets in terms of size, balance, and number of dimensions.

3 Data Acquisition and Feature Computation

Before we introduce the hybrid classifier we introduce the data our experimental study is based on. We employ a data set that contains measurements of everyday appliances that are turned *on* and *off*. We use two data sets, see Appendix A: (1) The *install* data set consists of 120 patterns that have manually been recorded and labeled at the beginning of the field study, when the system was installed. This data set serves as minimal training set, and is consequently very important for practical scenarios, e.g., for calibration of a novel system. It is balanced, i.e., the number of patterns for each class is approximately equal. (2) The electrical data of the *field study* consists of patterns that have been recorded in a household test environment, and a study that lasted approximately one month. We used motion sensors in every room of the test environment to facilitate the manual labeling of the data.



(a) (b)

Fig. 1. Visualization of the *field study* data space, different classes are shown in different colors: (a) features F_1, F_2 and F_3, and (b) features F_1, F_3 and F_4. Some patterns are accumulated in a small part of data space, others are scattered in a larger part of data space.

Based on the measurement of electrical parameters (voltage $U_{\text{eff}}(t)$, amperage $I_{\text{eff}}(t)$, and phase angle $\varphi(t)$) by a current sensor (with a sample rate of 5 Hz) that was centrally installed in the fuse box, the active resistance $R(t)$ is computed, see Equation (1), which is usually constant for the same appliance in different environments (that means different numbers of concurrently running appliances at a circuit):

$$R(t) := \frac{U_{\text{eff}}(t)}{I_{\text{eff}}(t) \cdot \cos\varphi(t)} = \frac{U_{\text{eff}}^2(t)}{P(t)}, \tag{1}$$

with power $P(t)$. Based on resistance from the *turn on* event of an appliance, the following features are extracted:

- mean resistance $F_1 := \overline{R}$ of $[R_{t_0+2}, \ldots R_{t_n}]$ [1],
- corresponding standard deviation $F_2 := \sigma_R$ from $[R_{t_0+2}, \ldots R_{t_n}]$, and
- maximum phase of DFT: $F_3 := \max\varphi$ of $[R_{t_0+2}, \ldots, R_{t_n}]$.

From the *turn off* event the following feature is extracted:

- median resistance $F_4 := \overline{R}$ of $[R_{t_n}, \ldots R_{t_{n-2}}]$.

Normalization did not have a significant influence on the experimental results in Section 5. To summarize, for the experimental part we employ measurements of 15 appliances that can be turned *on* and *off*, resulting in $N = 2,620$ four-dimensional patterns, and 30 different classes. Figure 1 shows a visualization of the whole data set. The majority of the patterns is accumulated in a small part of the data space. Few patterns are scattered around in data space (e.g. the green spots), and are hence easier to separate. This also motivates the application of ensemble classifiers employing techniques tailored to special data space conditions.

4 Hybrid Classifier

In this section we introduce an ensemble classifier that hybridizes a multi-class SVM and K-nearest neighbors for device recognition. SVMs are based on optimizing a decision boundary in a feature space, while K-nearest nearbors is based on aggregating labels of the closest patterns in data space. In the last part of this section we introduce the hybridization of both methods.

4.1 Multi-class Support Vector Machines

Classification is the problem to predict discrete class labels $y \in \mathcal{Y}$ based on input patterns $\mathbf{x} \in \mathbb{R}^d$, where $\mathcal{Y}$ is the discrete set of classes. The classification process uses observations of the form $\mathcal{T} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N))$. Objective is to learn a model that is able to predict the class labels for unknown data samples. It is

[1] The first two values R_{t_0}, and R_{t_0+1} turned out to vary too much for the same appliance, and are therefore left out.

not easy to learn reliable predictions for all kinds of data sets. The data set may be noisy, or classes may not be linearly separable, and difficult to separate with simple rules or mathematical formulations.

SVMs have developed to strong methods for classification [14]. They are based on maximization of the *margin* corresponding to the distance of a linear decision boundary that classifies data samples $\mathbf{x} \in \mathbb{R}^d$ into two classes. The hard margin SVM defines the linear decision boundary that correctly classifies all input examples. The optimization problem becomes:

$$\begin{aligned} &\text{minimize}_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ &\text{subject to : } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \\ &\text{for } i = 1, \ldots, N. \end{aligned} \tag{2}$$

With kernel functions the problem can be transformed into a space where the data samples can be separated with the decision boundary. Multi-class SVMs arise as an extension of two-class SVMs in a one vs. all training process.

4.2 K-Nearest Neighbors

KNN is a technique with long tradition. Cover and Hart [6] investigated the approach experimentally in the sixties. Interesting properties have been found, e.g., that for $K = 1$, and $N \to \infty$ KNN is bound by the Bayes error rate. For an unknown pattern $\mathbf{x}'$ KNN for classification predicts the class label of the majority of the K-closest patterns in data space. The multi-class KNN-classifier is defined as follows:

$$f_{\text{KNN}}(\mathbf{x}') := \arg\max_{y \in \mathcal{Y}} \sum_{i \in \mathcal{N}_K(\mathbf{x}')} \mathcal{I}(y_i = y) \tag{3}$$

with set $\mathcal{N}_K(\mathbf{x}')$ containing the indices of the K-nearest neighbors of $\mathbf{x}'$, and indicator function $\mathcal{I}(\cdot)$ that returns *one* if its argument is true[2], and *zero* otherwise. The choice of a distance metric has an important part to play. As the patterns in our recognition task are low-dimensional, we employ the Euclidean distance.

4.3 Ensemble

SVMs are known to be advantageous in *global* scenarios, i.e., high-dimensions, and sparse data [10], while KNN is known to be a *local* method, i.e. for low-dimensional data spaces and a large number of training samples. Hence, the hybridization of both classifiers is a reasonable undertaking in practical applications, where training set sizes, and numbers of features may vary. Algorithm 1 shows the pseudo-code of the ensemble classifier template that is the basis of the classification process. The ensemble classifier gets the training set $\mathcal{T}$ as input. For a pattern $\mathbf{x}'$ to be classified each classifier $f_i \in \mathbf{f}$ of the ensemble returns a

[2] i.e., label y_i of pattern $\mathbf{x}_i$ is y.

prediction. In our ensemble all classification results are aggregated to one label with the same weight (also known as bootstrapping or bagging), employing a majority vote. Let $y \in \mathcal{Y}$ be a class label. The ensemble classifier decision is defined as follows:

$$f_{\text{ENS}}(\mathbf{x}') = \arg\max_{y \in \mathcal{Y}} \sum_{f_i \in \mathbf{f}} \mathcal{I}(f_i(\mathbf{x}') = y) \tag{4}$$

Like the multi-class KNN classifier it makes use of the indicator function $\mathcal{I}(\cdot)$ to count the number of classifiers that vote for each label, and choose the decision of the most classifiers.The idea of a majority vote is that the majority corrects potentially false decisions of the minority. If there is no reason to believe that one of the classifiers achieves a better accuracy than the others, a majority vote probably obtains the best predictive performance. This principle is employed in Algorithm 1. We combine the classifiers to the following ensembles:

- ENS-SVM is an SVM ensemble classifier, which combines the SVM with linear kernel, and the SVM with RBF-kernel,
- ENS-KNN combines three KNN classifiers with different neighborhood sizes, i.e., $K = 1, 5$, and 7, and
- ENS* combines all five classifiers (SVMs with both kernels, and KNN with three neighborhood sizes).

Algorithm 1: Ensemble Classifier Template

Require: Training set $\mathcal{T}$, pattern $\mathbf{x}'$, set of classifiers $\mathbf{f}$, **Request:** $f_{\text{ENS}}(\mathbf{x}')$

1: **for** f_i in $\mathbf{f}$ **do**
2: compute $f_i(\mathbf{x}')$
3: **end for**
4: **return** $f_{\text{ENS}}(\mathbf{x}') = \arg\max_{y \in \mathcal{Y}} \sum_{f_i \in \mathbf{f}} \mathcal{I}(f_i(\mathbf{x}') = y)$

5 Experimental Analysis

In the following, we experimentally analyze multi-class SVMs, and KNN for appliance recognition, first for the *install* data set, then for the whole *field study* data set.

5.1 Installation Data

In the first experimental setup we employ the small data set *install* as training set, see first row of Table 1. The *field study* data is employed as test set. The SVM with linear kernel, KNN with $K = 1$, and two of the ensemble methods achieve a low error rate. An appliance recognition rate of over 92% is achieved, which is probably sufficient for many practical scenarios. While the SVM with

Table 1. Experimental results of the SVM, KNN and ensemble classifiers on the *Lab*-, and the field study data set with varying training set sizes α (proportion of training samples and the data set size). The lowest error rates are shown in **bold**, the second best in *italic* numbers. The best classifier in each experiment gets two points, the second best one point for the score.

training set	SVM linear	SVM RBF	KNN $K=1$	KNN $K=5$	KNN $K=7$	ENS SVM	ENS KNN	ENS *
install	**0.0787**	0.4767	0.0883	0.2977	0.2927	0.0837	0.2739	*0.0802*
10^{-1}	*0.0526*	0.0915	0.0652	0.0560	0.1430	0.0594	0.0560	**0.0514**
9^{-1}	**0.0480**	0.0858	0.0606	0.0537	0.0697	0.0549	0.0526	*0.0491*
8^{-1}	**0.0480**	0.0823	0.0629	0.0549	0.0663	0.0560	0.0514	*0.0491*
7^{-1}	**0.0480**	0.0800	0.0617	0.0549	0.0617	0.0549	**0.0480**	**0.0480**
6^{-1}	0.0491	0.0789	0.0629	0.0537	0.0629	0.0549	**0.0469**	*0.0480*
5^{-1}	0.0480	0.0778	0.0606	**0.0446**	0.0491	0.0537	*0.0469*	*0.0469*
4^{-1}	0.0491	0.0709	0.0594	**0.0446**	0.0480	0.0549	*0.0469*	*0.0469*
3^{-1}	0.0514	0.0663	0.0606	**0.0434**	*0.0480*	0.0549	0.0503	0.0491
2^{-1}	0.0457	0.0617	0.0617	0.0491	**0.0434**	0.0526	0.0457	*0.0446*
$2 \cdot 3^{-1}$	–	0.0572	0.0617	*0.0400*	**0.0389**	0.0572	0.0434	0.0446
$\sum$ score	9	0	0	6	5	0	7	11

linear kernel is the best classifier in this scenario, the SVM with RBF kernel fails. We expect a strength of KNN in case of small training sets. The data set *install* is obviously too small for the KNN classifiers; for $K = 5, 7$ error rates larger than 25% have been achieved. The KNN ensemble also achieves high error rates due to the results of both weak KNN variants. The other ensembles take advantage of the strengths of the SVM with linear kernel, or KNN with $K = 1$.

5.2 Field Study Data

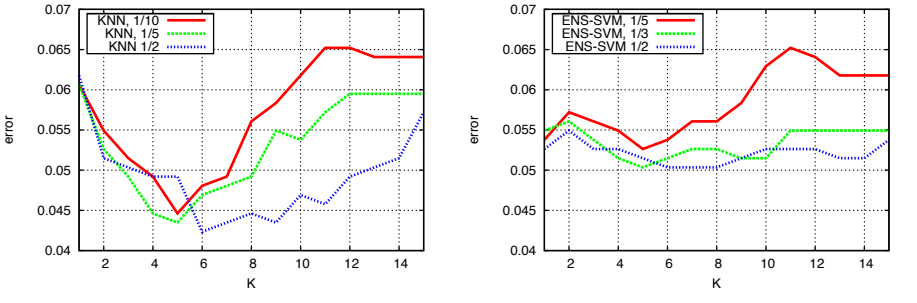
In the second experimental setup we analyze the classification error rate w.r.t. the rate $\alpha = \frac{|\mathcal{T}|}{N}$ of the size of the training set $|\mathcal{T}|$ and the data set size N. The error rate is computed on a test set of size $1/3|\mathcal{T}|$. It can be observed that a training set size of $N = 262$ (corresponding to 1/10th of the data) can achieve an accuracy of up to $\approx 95\%$ with the SVM-KNN-ensemble ENS*, and also with the SVM with linear kernel. The best recognition rate (error 3.89%) is achieved with KNN, and $K = 7$. For the SVM approach we can observe that a linear kernel is better than an RBF kernel, and better than KNN with $K = 5, 7$ for training sets smaller and equal to 6^{-1}. But the SVM with linear kernel takes significantly longer for training as of training set sizes larger than 5^{-1}, and did not terminate in case of $2 \cdot 3^{-1}$. Obviously, with the (four) *high-level* features KNN is sufficient, in particular, if the training set size is large.

Concerning the ensemble classifiers we can observe low error rates in the majority of the experiments. The ensemble classifier ENS* that employs all five classifiers turns out to be the algorithm with the most robust results. It is the best or second best classifier in nine of eleven cases, which is also reflected by the

highest sum of scores. Also the KNN ensemble classifier ENS-KNN achieves good results on the *field study* data. The results are similar to KNN with $K = 5, 7$ (also the failure in case of the *install* data set, which cannot be compensated by KNN with $K = 1$). The constantly good results of ENS* in comparison to most other classifiers motivate the employment of the SVM-KNN ensemble in practical scenarios. For small training sets the employment of SVMs with linear kernel is a good recommendation.

5.3 Neighborhood Sizes of KNN

The question arises how to choose the neighborhood size K for the KNN classifiers, and also for the ENS* ensemble. Figure 2 shows the influence of K on the error rate for the KNN classifier w.r.t. different training set sizes. In case of KNN, see Figure 2(a), we can observe that neighborhood sizes around $K = 4$ and $K = 5$ are optimal for small training sets. On larger training sets (2^{-1}) the influence of the neighborhood sizes is less significant, which can be explained as follows: If many patterns are available KNN can average over more training samples without a deterioration of the classification error. In case of the ensemble ENS* the neighborhood size of KNN is less important for both larger training sets 3^{-1}, and 2^{-1}. The SVM classifiers compensate the negative effect of too large neighborhoods. This is another motivation for the employment of ensembles: bad parameterizations can be compensated.



(a) (b)

Fig. 2. Study of neighborhood size K w.r.t. training set size 5^{-1}, 3^{-1}, and 2^{-1} for (a) KNN, and (b) ENS*. The neighborhood size has a significant influence on the classification error in case of the KNN classifiers, but the effect is compensated in the ensemble.

6 Conclusions

In this work we demonstrated how problems in load monitoring can be solved with an ensemble of multi-class SVM and KNN classifiers with various neighborhood sizes. Ensemble classifiers are well appropriate to solve this task as

practical NIALM data sets are often unbalanced, vary in training set sizes, and in the number of training samples. We compared the classifier ensembles to the state-of-the-art classifiers in machine learning (SVMs). The experiments with our ensemble classifiers on the electricity data of the experimental studies has shown satisfying recognition results. The results confirmed the theoretical expectations: SVMs are a good choice in case of small training sets, while KNN shows its strengths on large training sets. We recommend to combine both worlds: KNN that can adapt to any situation without assumptions about the data, but turns out to be unstable in many situations (high variance and low bias), and SVMs that are based on the assumption of linearity of the data, which is softened by kernel functions and slack variables (low variance and high bias).

The ensemble classifiers turned out to be very robust. In particular, ENS* that employs all five classifiers with a bagging majority vote achieves the best and second best error rates in most of the experiments. The achieved robustness and high recognition rates are important steps towards an efficient integrated real-time approach of label retrieval and appliance recognition. As future work, we plan to improve the recognition accuracy with semi-supervised support vector approaches [8] taking into account a large set of unlabeled electricity data, which is often available in practical scenarios.

A Test Sets

Table 2 shows the appliances that are part of the test data sets *install*, and of the *field study*.

Table 2. List of 15 appliances of *install* data set, and the field study data set

#	appliances	#	appliances	#	appliances
1	shelf light	6	table light, bedroom	11	ceiling lamp, bathroom
2	fridge	7	table light, TV	12	ceiling lamp, living room
3	bedside lamp	8	table light, door	13	ceiling lamp, corridor
4	desk lamp	9	kettle	14	ceiling lamp, bedroom
5	TV	10	mirror lamps	15	air conditioning

Acknowledgements. This work has been supported in part by funds of the *Federal Ministry of Economy and Technology* in the E-Energy project *eTelligence*, project number $01MR08007A$, and *The Lower Saxony Research Network Design of Environments for Ageing*.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Baranski, M., Voss, J.: Genetic algorithm for pattern detection in NIALM systems. In: IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 3462–3468 (2004)

3. Chang, H.-H., Yang, H.-T., Lin, C.-L.: Load Identification in Neural Networks for a Non-Intrusive Monitoring of Industrial Electrical Loads. In: Shen, W., Yong, J., Yang, Y., Barthès, J.-P.A., Luo, J. (eds.) CSCWD 2007. LNCS, vol. 5236, pp. 664–674. Springer, Heidelberg (2008)
4. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
5. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
6. Cover, T., Hart, P.: Nearest neighbor pattern classification 13, 21–27 (1967)
7. Fung, G.P.C., Yu, J.X., Wang, H., Cheung, D.W., Liu, H.: A Balanced Ensemble Approach to Weighting Classifiers for Text Classification. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 869–873. Springer, Heidelberg (2006)
8. Gieseke, F., Kramer, O., Airola, A., Pahikkala, T.: Speedy Local Search for Semi-Supervised Regularized Least-Squares. In: Bach, J., Edelkamp, S. (eds.) KI 2011. LNCS, vol. 7006, pp. 87–98. Springer, Heidelberg (2011)
9. Hart, G.W.: Nonintrusive appliance load monitoring 80(12), 1870–1891 (1992)
10. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, Berlin (2009)
11. Lin, Y.-H., Tsai, M.-S.: Applications of hierarchical support vector machines for identifying load operation in nonintrusive load monitoring systems. In: Intelligent Control and Automation (WCICA), pp. 688–693 (2011)
12. Opitz, D.W., Maclin, R.: Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research (JAIR) 11, 169–198 (1999)
13. Patel, S.N., Robertson, T., Kientz, J.A., Reynolds, M.S., Abowd, G.D.: At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line (Nominated for the Best Paper Award). In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 271–288. Springer, Heidelberg (2007)
14. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999)
16. Woods, K., Kegelmeyer, W.P., Bowyer, K.W.: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4), 405–410 (1997)

Lee Path Replanner for Partially-Known Environments

Maciej Polańczyk, Przemysław Barański, Michał Strzelecki, and Krzysztof Ślot

Institute of Electronics, Technical University of Lodz,
Wolczanska Street 211/215, 90-924 Lodz, Poland
maciej.polanczyk@gmail.com,
{przemysław.baranski,mstrzel,kslot}@p.lodz.pl

Abstract. This paper presents a new routing algorithm based on Lee's algorithm. The latter was developed with view to designing printed circuit boards (PCB). It can also be adapted for seeking collisionless routes for unmanned vehicles moving in an unknown environment. Such routes need to be updated on-the-fly, taking into account changes in the environment, e.g. moving obstacles, newly detected objects. The proposed algorithm uses results from the calculations carried out in the previous steps. Hence, computations in the following steps are only required for the areas that were subject to change. Tests showed that the presented route replanner is, on average, twice as fast as Lee's algorithm.

Keywords: path planning, map of obstacles, mobile platform motion system.

1 Introduction

One of the most important issues related to autonomous robotic mobile platformsis their ability of avoiding still and moving obstacles when navigating in an unknown environment. In the majority of such systems, e.g. mobile robots operating without an explicitly planned route, it is not possible to ensure obstacle map defined *a priori*. In such systems this information has to be discovered and updated in real time. To deal with problems of mobile platform operation in dynamically changing and usually unknown environments, the idea of path replanner was introduced.

The map of the platform under motion should be updated in two ways. First, every time the new obstacles are encountered, they should be marked on the map.At the same time, the position of obstacles already detected must also be updated, according to the new platform localization. This has to be done in order to preserve spatial relationships between objects. Both updates should be performed in a short time, to ensure the accurate platform localization. The necessary computational speed will depend on such parameters as maximum platform speed and number of acquired frames per second.

In this paper, a procedure for path planning, aimed for autonomous mobile platforms is presented. To provide a correct platform operation, the path should be recalculated every time a map is updated. Unfortunately, this approach is usually time-consuming. Therefore there is a need to improve the process of path planning,

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 332–342, 2012.

e.g. by consideration of results obtained in the previous step before new objects are detected. To implement this solution, a novel approach based on Lee's algorithm [1] is proposed. The experimental results show that the presented method is able to estimate a platform path in unknown indoor and outdoor environments such as corridors, office rooms or a parking lot. Moreover, the map calculation time is sufficiently short to ensure reliable and accurate platform localization. This work is the continuation of previously performed research, aimed at developing an obstacle avoidance procedure based on stereovision [2], including a fuzzy logic algorithm for motion control [3, 4].

2 Related Work

There are many publications covering the topic of collissionless route planning. Most of them assume that the deployment of obstacles is known before the unmanned vehicle sets off [5, 6, 7]. The A* algorithm [8], by means of heuristics, outperforms the latter in terms of the computation burden. Still, the aforementioned algorithms are not able to make use of the calculations carried out in the previous steps. This handicap is resolved by Lifelong Planning A* [9], which is a tuned version of A*. The next examples are the D* and D* lite algorithm [10, 11] which use propagation waves for cells update.Neural networks of different architectures, a proved tool in pattern recognition [15], are also employed for path planning[12, 13, 14].

The experiments we performed already demonstrated that the proposed algorithms are efficient, though the computation time largely depends on the number of changes in the map.

Lee's algorithm [1], which was originally designed to route copper paths for printed circuit boards, can also be harnessed to plan routes for autonomous vehicles. The algorithm is simple and the incurred computation burden is small. However, as it does not use heuristics, is slower than A*. Lee's algorithm can be improved by the possibility of using the calculation from the previous step as new objects are detected. This greatly reduces computation time.

3 Lee's Algorithm

The algorithm is made up of two stages – the forward and backward stage. Fig. 1. shows the subsequent steps of the algorithm.

In the forward stage, the map of obstacles is converted into a 2D table. Cells that represent the obstacles are marked with -1. The outset, i.e. the vehicle's location, is represented by a cell that stores a value of 1. All other cells are unlabeled meaning the vehicle can move through these areas. The algorithm visits all adjacent neighbours of the starting point and assigns them the value of the starting point incremented by one.These cells are stored on the L list. In the next step, the algorithm looks for the neighbours of the previously considered cells (retrieved from the L list)and in the same manner assigns them valuesand add them to the L list. As every cell can be visited several times (from different neighbouring cells), it is essential to store the

smallest possible value, bearing in mind that the algorithm should look for the shortest path. From then onwards, the algorithm iterates until the destination point is found, in which case the algorithm executes the backward search.

When the L list is empty and the destination point is not reached, it is not possible to find a path connecting the destination and the start point.

In the backward stage, the algorithm travels from the destination point through cells with the smallest possible values until the starting point is reached. These cells create the shortest path for the vehicle.

The algorithm needs to store the location of cells whose values have been changed. The list of such cells has to be sorted with respect to the cell's value from smallest to largest. It is crucial to notice, that during expansion, the algorithm starts the analysis from a cell of the smallest value. Therefore, the newly added cells will be at the end of the sorted list, thus the list is automatically sorted. This is a big advantage of the algorithm since the process of sorting a list is time-consuming. In this case a simple FIFO queue suits the need. New cells are appended to the list and cells to be analyzed are taken from the queue head.

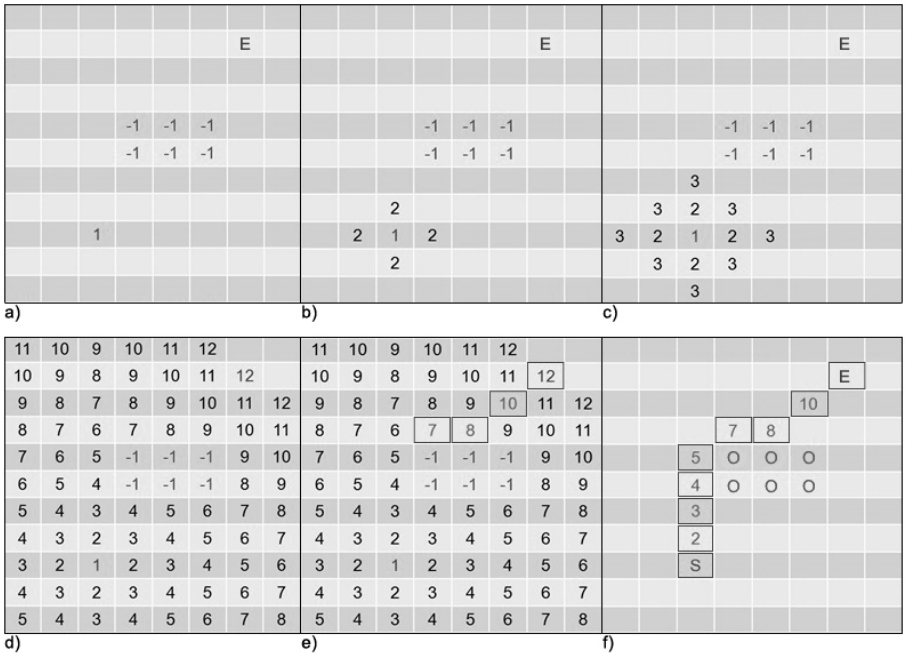


Fig. 1. Subsequent steps of Lee's algorithm. (a) Initialization. (b, c, d) Forward stage. (e, f) Backward stage.

4 Lee's Replanner

At the outset of the algorithm, the path is calculated from the original Lee's algorithm. When the environment evolves, the path is recalculated using the presented algorithm.

The proposed algorithm is made up of three stages (Fig. 2). In the first stage, the algorithm creates a list of cells that have changed. To check if a cell has been vacated or occupied by an obstacle, the state of the cell (i.e. the occupation flag) is compared at time instants *t* and *t-1*.

In the second stage, the analysis of those cells is performed. There are two possible situations: the area has been vacated or occupied by an obstacle at time instant t-1.

The former case is simpler. The cell representing this area is marked as unlabeled and can be reused. The algorithm looks for the neighbour that stores the smallest value. This neighbour is added to the list of expansion cells that will be analyzed by Lee's algorithm.

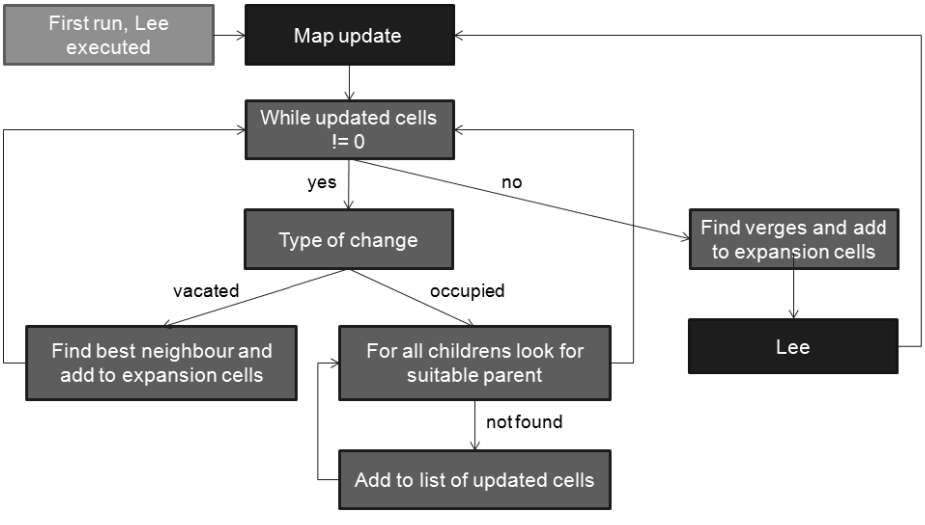


Fig. 2. Block diagram of Lee'sreplanner

The second case is more complicated. The cell is assigned -1 which stands for an obstacle. Next, all the adjacent cells that might be children of the cell in question are nalyzed. These children cells are checked if they have a neighbour of the same value as their parent. If so, nothing happens, as the integrity of the expansion is preserved. The rest of the cells are added to a list of cells that changed at time instant t-1. These cells will be analyzed at time instant t as was their parent. Hence, all cells that are not connected with the appropriate parent will be unlabeled. Finally, all the verges of the unlabeled areas are detected (Fig. 3).

Having analyzed all the cells that were changed, the last stage is executed. The updated list of the expansion cells is processed by Lee's algorithm in its basic form. The subsequent steps of this method are shown in Fig. 4.

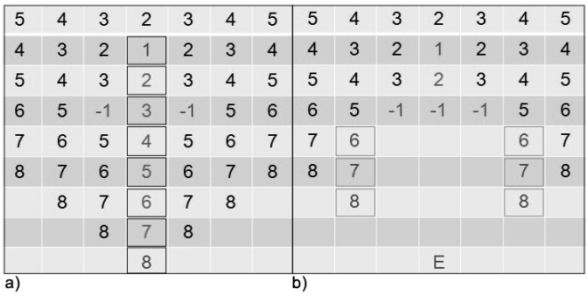


Fig. 3. Detection of verges of the unlabeled. (a) Before update. (b) Verges detected.

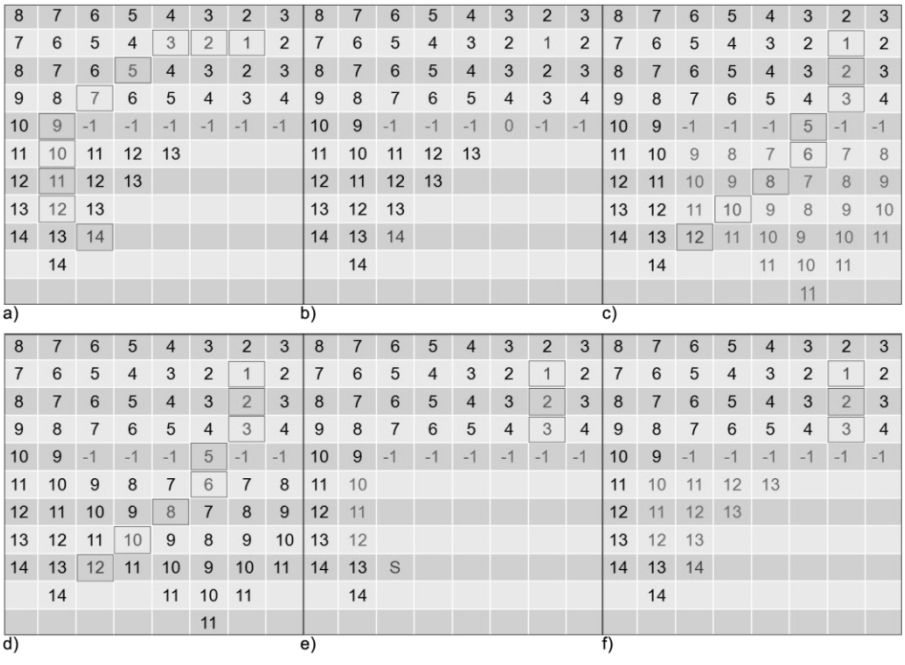


Fig. 4. Subsequent steps of the Lee replanner. (a) Map at time instant *t-1*. (b) Map at time instant *t*. (c) New path for time instant *t*. Case when a cell became free and a shorter path is possible (d) Map at time instant *t-1*. (e) Map at time instant *t*. (f) After expansion, before backward stage. Case when a cell was occupied and a new path had to be computed.

Steps 1 through 3 are repeated when recomputing the path. New cells are added to the expansion list. These cells need to be added at the appropriate position in the list according to their values. The list has to be sorted from smallest to largest values. Therefore a FIFO queue cannot be applied in this context. The implementation of a sorted list would slow down the execution time of the algorithm. The profit from the computations performed in the previous steps would be lost. The problem is solved by using a table of queues denoted with *q[x],* whereby *x* is the largest possible value of a cell (e.g. for a 100x100 map, *x* would equal 10000). New cells are added to the list in the following manner:

$$q[V-1].push(C) \ . \tag{1}$$

where: q – table of queues, C- cell to be appended, V – the value of a cell.

The algorithm iterates through every address of the *q[x]* table to find a queue of non-zero length and retrieve from them subsequent cells. Hence, sorting can be avoided. Another improvement is to store the minimum and maximum address of non-empty queues. Therefore Lee's algorithm does not need to waste time checking empty queues.

Another difference lies in the expansion direction. In the considered algorithm, the expansion is executed towards the cell representing the vehicle. Therefore, the position change of the vehicle does not affect the calculations.

5 Experimental Results

The algorithm was tested in an autonomous vehicle, which moved around the rooms of the university and the parking lot. The map of obstacles was built on the base of stereoscopic pictures recorded by a camera [16] mounted on the moving vehicle. The process of building the obstacle map was presented in [2].

The route calculations were performed on a 2.53GHz processor. The results are shown in Table 1, 2 and 3. The tables 1 and 2show a comparison,of computation time and number of cells that were processed, with the competitive Lifelong Planning A* algorithm [9].Table 3 shows average path length and average number of modified cells. The presented results concerns experiments in the university building, although similar results were observed in the parking lot.

The path determined by original Lee's and modified Lee's replanner algorithm at time instant t is the same. The path created by the Lifelong Planning A* algorithm is very similar to the one calculated by the aforementioned algorithms. The lengths of obtained paths differed by less than 5%. Thus pathsestimated by all tested algorithms were similar considering both their lengths and shapes as well.

Tests were repeated many times with different sizes of cells and various obstacles location.

Pictures 5a, 6a and 7a present photos captured by the stereo camera during the trials. Pictures 5b, 6b and 7b depict a map of obstacles includingvelocity vectors. The triangles stand for the field of view of the camera. The continuous lines denote the paths from the start to the destination points.

Table 1. Average times of computation

Map size [cells]	Computation time [ms]		
	Lee	Lee path replanner	Lifelong Planning A*
100x100	<1	<1[1]	<1 – 10[1]
500x500	8	2-6[1]	1-300[1]
1000x1000	35	15-30[1]	3-1000[1]
2000x2000	135	50-100[1]	10-3000[1]

[1] Dependent on the distance to the destination and the size of cells (in this case 10m in a straight line and 0.1x0.1m respectively). The path's length was subject to alternation at time instant t due to redeployment of the obstacles.

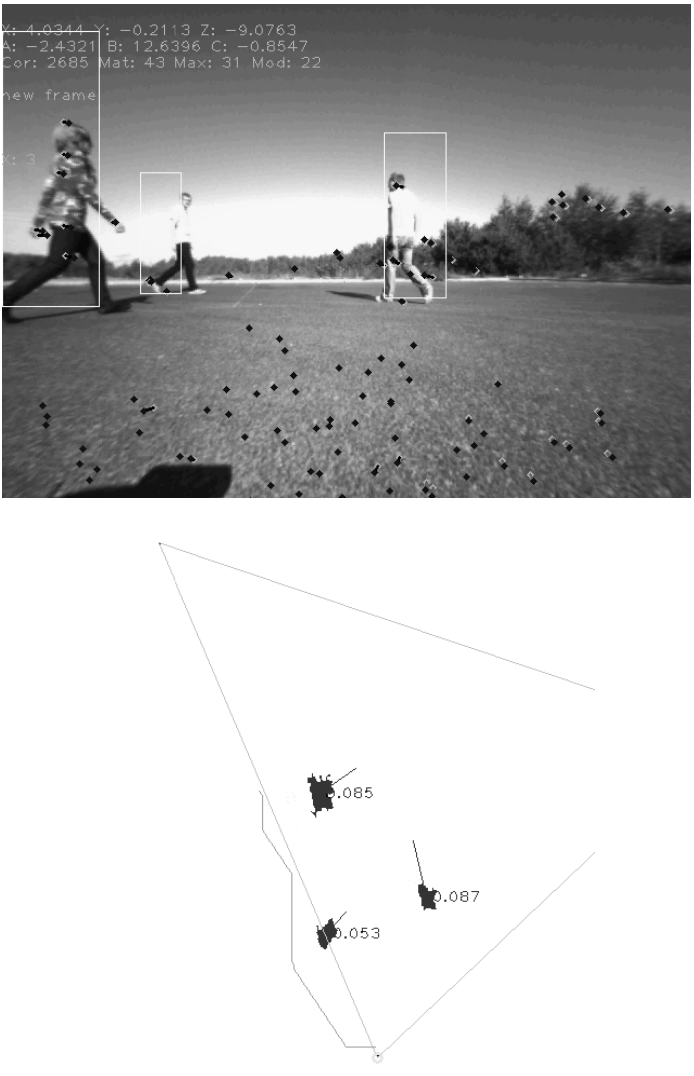


Fig. 5. Results for moving object detection. (a) The image from the stereo camera. (b) Obstacle map with marked objects velocities and found path.

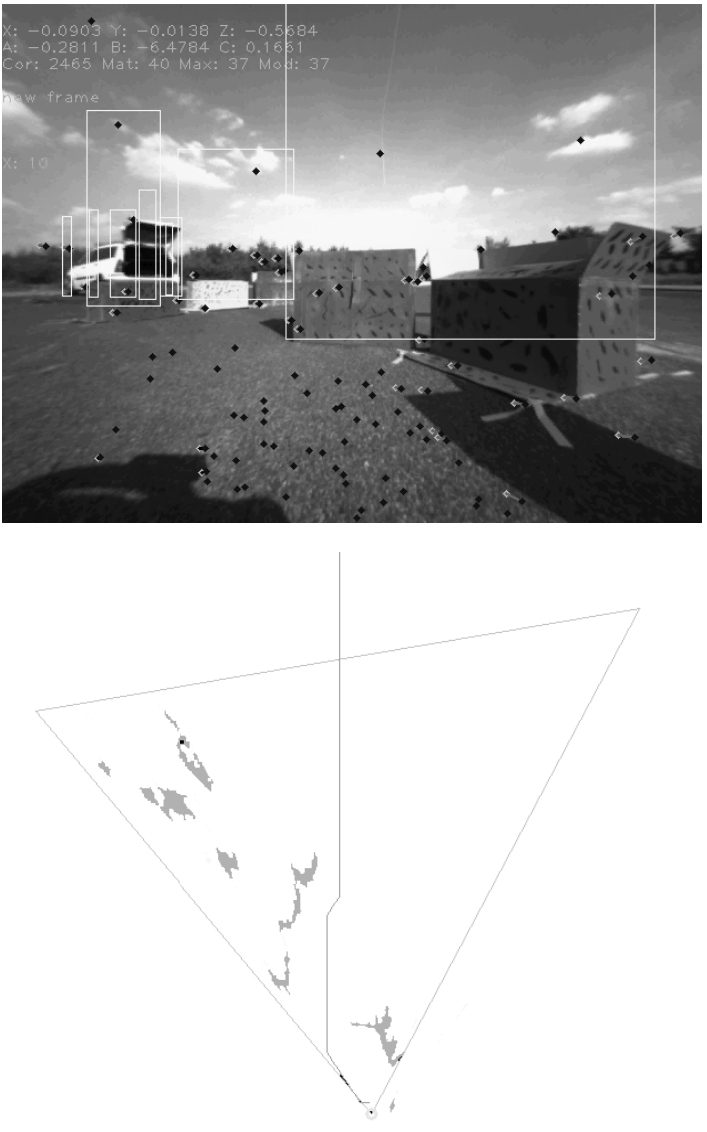


Fig. 6. Results for static object detection. (a) The image from the stereo camera. (b) Obstacle map with found path.

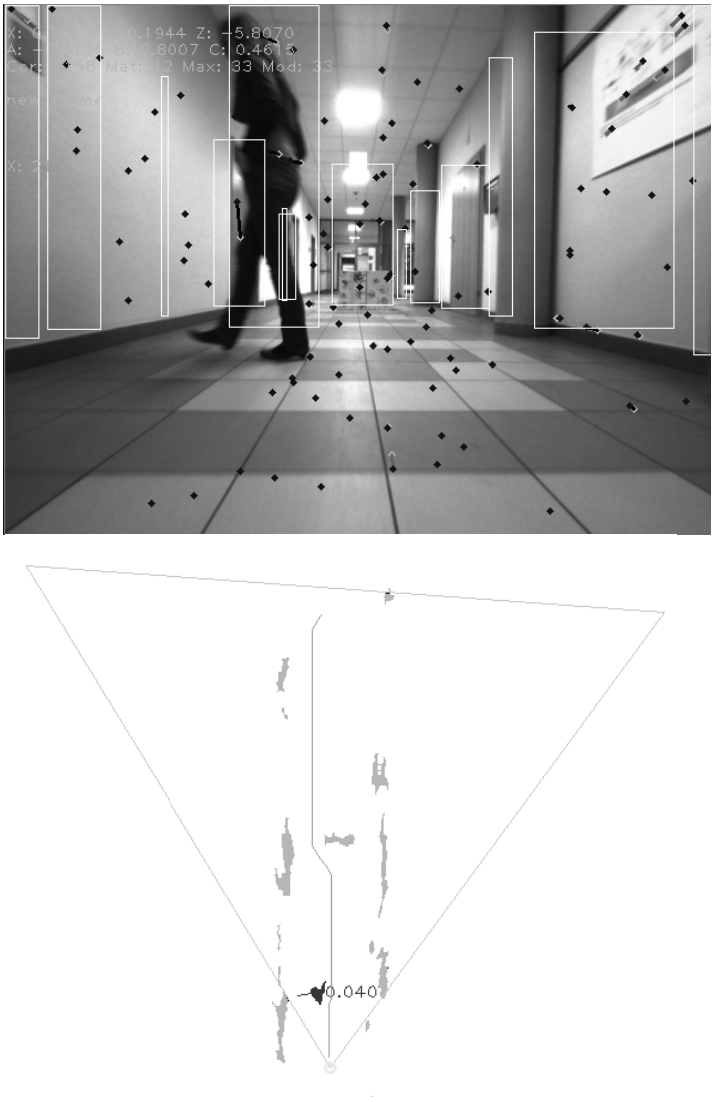


Fig. 7. Results for moving and static object detection. (a) The image from the stereo camera. (b) Obstacle map with found path.

Table 2. Average number of cells that were processed

Map size [cells]	Number of cells that were processed		
	Lee	Lee path replanner	Lifelong Planning A*
100x100	7.8k-8k	0.05k-1.9k	0.019k-2.2k
500x500	195k-200k	22k-50k	0.083k-60k
1000x1000	750k-820k	50k-300k	0.6k-240k
2000x2000	3000k-3200k	500k-800k	2.4k-1000k

Table 3. Additional information

Map size [cells]	Path length	Number of modified cells
100x100	95-150	100-150
500x500	470-750	1500-2700
1000x1000	950-1500	5000-10000
2000x2000	1900-3000	15000-40000

6 Conclusion

The presented algorithm for planning collisionless routes in an unknown environment proved to be faster than Lee's algorithm. The proposed algorithm behaved in a more stable way than Lifelong Planning A* in an evolving environment. The results show that the proposed algorithm is particularly effective in applications requiring run-time routing for unmanned moving vehicles.

Further research aims at estimating the direction and velocity of moving objects, so that a new path will account for the future location of obstacles.

Acknowledgements. The project is financed by the National Center of Research And Development in years 2010-2013 under the grant NR02-0083-10.

References

1. Lee, C.Y.: An algorithm for path connection and its applications. IRE Trans. on Electronic Computers EC-10(3), 346–365 (1961)
2. Polańczyk, M., Owczarek, A., Strzelecki, M., Ślot, K.: Stereovision-Based Obstacle Avoidance Procedure for Autonomous Mobile Platforms. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 206–213. Springer, Heidelberg (2011)
3. Corchado, E., Abraham, A.: André Carlos Ponce Leon Ferreira de Carvalho: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
4. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
5. Latombe, J.-C.: Robot Motion Planning. Kluwer Academic Publishers, Boston (1991)
6. Manikas, T.W., Ashenayi, K., Wainwright, R.L.: Generic Algorithms for Autonomous robot Navigation. IEEE Instrumenation & Measurement Magazine, 27–31 (December 2007)
7. Du, X., Chen, H.-H., Gu, W.-K.: Neural network and genetic algorithm based global path planning in a static environment. Journal of Zhejiang Univeristy Science 6, 549–554 (2005)
8. Hart, P., Nilsson, N., Rafael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4, 100–107 (1968)
9. Koenig, S., Likhachev, M., Furcy, D.: Lifelong Planning A*. Artificial Intelligence Journal 155(1-2), 93–146 (2004)

10. Stentz, A.: Optimal and efficient path planning for partially-known environment. In: Proc. Int. Conf. on Robotics and Automation, vol. 4, pp. 3310–3317 (1994)
11. Koenig, S., Likhachev, M.: D* Lite. In: Proc. of the AAAI Conference of Artificial Intelligence (AAAI), pp. 476–483 (2002)
12. Lebedev, D.V., Steil, J.J., Ritter, H.J.: The dynamic wave expansion neural network model for robot motion planning in time varying environments, vol. 18, pp. 267–285. Elsevier Science Ltd. (2005)
13. Li, H., Yang, S.X., Biletskiy, Y.: Neural network based path planning for a multi-robot system with moving obstacles. Automation Science and Engineering, 163–168 (2008)
14. Kroumov, V., Yu, J.: Neural Networks Based Path Planning and Navigation of Mobile Robots. Recent Advances in Mobile Robotics (2011)
15. Strzelecki, M., Materka, A., Drozdz, J., Krzeminska-Pakula, M., Kasprzak, J.D.: Classification and segmentation of intracardiac masses in cardiac tumor echocardiograms. Computerized Medical Imaging and Graphics 30(2), 95–107 (2006)
16. BumbleBee2, `http://www.ptgrey.com/products/bumblebee2/index.asp`

Stroke Based Handwritten Character Recognition

D. Álvarez, R. Fernández, and L. Sánchez

Department of Mechanical, Computer and Aerospace Engineerings,
University of León, Spain
etcdal00@estudiantes.unileon.es,
{ramon.fernandez,lidia.sanchez}@unileon.es
http://www.unileon.es

Abstract. This work proposes a new stroke based methodology for handwritten character recognition. After the pre-processing, several steps are involved to achieve the recognition. First, the character is segmented into its strokes. Then, we determine the maximum length of the longest horizontal segment that can be inscribed on a stroke. We also compute that for the vertical direction. So we decide whether the stroke must be tagged as horizontal or vertical. After that, we represent by a string the vertical stroke position and its relational-ship with its adjacent horizontal strokes. In that string, each vertical stroke is represented by a character followed by a set of numbers which means where the adjacent horizontal strokes join the vertical one. A formal language grammar has been set and a knowledge base developed with known characters written by a single writer. Finally, an inference engine allow us to recognize unknown characters written by that single user. This algorithm has been tested on different writers and provides a hit rate of 87,13%.

Keywords: Handwritten recognition, character recognition, stroke character representation, inference engine, knowledge base.

1 Introduction

When a document is written, it can be translated to ascii code to be used in personal computers. Commonly optical recognition is used and this kind of recognition is usually called ICR (Intelligent Character Recognition). Specifically, in Intelligent Character Recognition or Handwritten Recognition there are two different ways: On-Line and Off-Line recognition. The main difference between them is how data are acquired and processed.

On-Line recognition, is done while writing is performed. For example, there is a study [5] that takes as reference to obtain certain points and their direction when a pen is writing over PDA to get a character description. However, the method more commonly used is Hidden Markov Models (HMM). So Han Shu[7], calculates the probability of being a certain character from feature extraction

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 343–351, 2012.

made by GRID LCD. Another way to get handwritten character [2] is preprocessing the input data and finally, the obtained strokes are compared with precalculated models to guess the introduced word. Markov Models generally have very good results, taking in some cases an accuracy of 98%.

Off-Line recognition is done after writing the text and saving this text as an image. In Off-line recognition, the most common process is using a neural network. Mathur, Aggarwal, Joshi and Ahlawat use a handwritten document and after processing, words are segmented to obtain separated characters [11]. Finally, they introduce these characters in a neural network to obtain the result. They achieve a 71% of recognition rate.

Alphabet used for recognition influences in the results. Arabic handwritten recognition [8] has a 89,3% of accuracy, and an optical character recognition systems for handwritten Gujarati numbers [12] have achieved approximately 82% of hit rate. However, [3] handwritten Chinese Character Recognition has a recognition rate between 96.4% and 96.6%. Of course, all of them use neural networks. It should be noted that neural networks play an important role in the development of methodologies for handwritten recognition. It could be considered to use neuronal networks for fuzzy logic applied to this field [14]. Researching more on neuronal networks, it could be possible develop a hybrid neuronal network, for example using a combination of different transfer projection functions of neuronal network as proposed in HAIS'07 [15].

Form recognition is the most widely used. When a form with a personal bank account number and other data is filled out, these data can be introduced to the computer through a recognizer [9]. In this case, data are obtained as the amount, date or signature by analyzing the regions of the document form. A slant correction is applied over characters, then are normalized and finally, introduces results on a trained neural network. This method is applied to other types of checks from other banks [10] and obtain results of accuracy of 92.6%.

The pattern recognition is also widely used as show Pavlidis I., Singh R. and Papanikolopoulos N.P.[16]. In the 3rd International Workshop on Hybrid Artificial Intelligence Systems (HAIS'08) [13] was proposed many pattern recognition on hybrid intelligent systems to extract useful information from vast amount of data and discover meaningful patterns. It could be used for handwritten recognition applied on the input images to be recognized.

In section 2, we explain the proposed methodology. Section 3 shows the experiments an the obtained results. To finish, section 4, gathers conclusion and future works.

2 Methodology

In this section, we describe how the image is acquired and the slant correction procedure. Then we explain the morphological operations and the feature extraction. Finally, we present the method to get a string representation to be introduced in inference engine. All these steps are involved to achieve the recognition.

2.1 Image Acquisition and Preprocessing

First, a character is acquired through an scanner or similar device. Once the image is obtained, it is binarized into two levels(black or white), using the thresholding method.

Then a binary matrix $M_c = (x, y)$ is constructed whose dimension is the same as the image. Each pixel of this image is represented in this matrix by 0 for white pixels and 1 for the black ones.

2.2 Processing

Once matrix is filled with the pixel values, it is necessary to apply slant correction on the introduced character. We apply a nonuniform slant correction to the matrix [4] and, whit the study of vertical projection profile [6], we will obtain the unslanted character. To do this, the highest and lowest pixel of the matrix is calculated and each 1-valued pixel is moved using the following rules:

- The considered angle α satisfies $-45^o > \alpha > 45^o$.
- The height h is calculated for each pixel.
- The distance to move a pixel is calculated as described in Fig.1 by: $M = tg(\alpha \frac{\pi}{180})h$

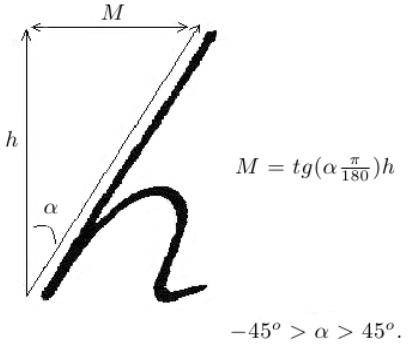


Fig. 1. Getting the distance to move the pixel

- The pixel in a position h takes value 0 and the pixel in a position $h \pm M$ takes value 1.
- We calculate a vertical projection profile of the resulting matrix.

Finally, all vertical projection profiles are compared and the one with the highest peak corresponds to the matrix that identifies the unslanted character. In Fig. 2 we can see an example.

To obtain an optimal image for further processing it is necessary to apply on the image certain morphological operations. It performs a dilation operation,

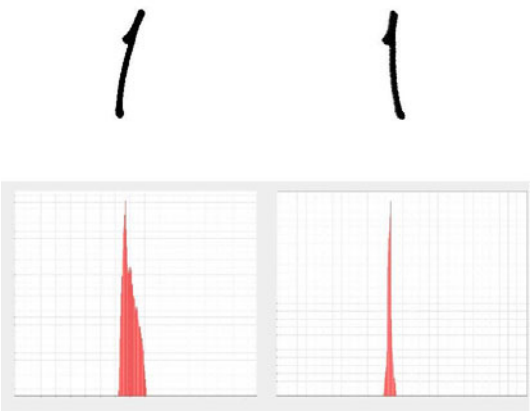


Fig. 2. Character '1' and its vertical projection before (left) and after (right) the slant correction

followed by erosion, which fills in small holes caused by moving pixels in the previous step.

The structuring element used is the same to both operations.

2.3 Stroke Segmentation

The feature extraction employed in this work was proposed at the V Congress of Hispalinux [1]. The image, which is itself a character, can be considered that it is composed by horizontal and vertical strokes. To segment a character into its strokes, the next rules must be followed:

- Each pixel belongs to one and only one stroke.
- The stroke can be quantized into two different values: vertical and horizontal. We can see an example of the character "r" in Fig. 3.
- The main strokes are vertical, so there is not any horizontal stroke laying over nor under a vertical one.

The first step to find the different strokes consists on assigning a stroke to each pixel. This can be accomplished by measuring the segments at 0 and 90. If the segment at 0 is the longest one, the involved pixel is labelled as a horizontal stroke. Else, the stroke is considered as vertical and so is marked the pixel.

Finally, if a pixel belonging to a horizontal stroke lays in the same column as pixels in a vertical stroke, then the pixel is also considered as part of a vertical stroke. Fig. 4 shows rule 3 applied on character "t".

2.4 Generating the Representative String Applying the Grammar

The representative string will be defined by the interconnection that exists between the vertical strokes and the horizontal ones. To do this, four regions of

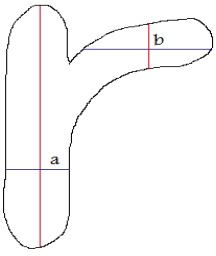


Fig. 3. Pixel at point "a" will be labelled as vertical. On the other hand, pixel at point "b" will be labelled as horizontal.

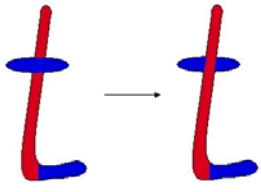


Fig. 4. Application of rule 3 over character "t": horizontal stroke lays under a vertical one

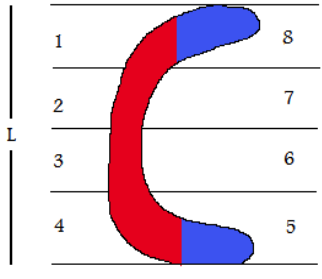


Fig. 5. Adjacency region at position "5" and "8". String representation: V58.

adjacency to the right and left vertical stroke is considered. Then, 8 regions are obtained and identified from "1" through "8" starting at the top left and ending at the top right. Thus, if there is a horizontal stroke located at the top right, it is tagged as in position "8" in the vertical stroke.

As shown in Fig. 5, for every vertical stroke and from left to right, the adjacencies with horizontal strokes are described. The grammar that is used to perform the representative string is a regular grammar $G_{cr} = (T, N, S, P)$ defined as:

T = V,1,2,3,4,5,6,7,8
N = V,1,2,3,4,5,6,7,8
S = V

P = {V → V|1|2|3|4|5|6|7|8,
1 → V|1|2|3|4|5|6|7|8, 2 → V|1|2|3|4|5|6|7|8,
3 → V|1|2|3|4|5|6|7|8, 4 → V|1|2|3|4|5|6|7|8,
5 → V|1|2|3|4|5|6|7|8, 6 → V|1|2|3|4|5|6|7|8,
7 → V|1|2|3|4|5|6|7|8, 8 → V|1|2|3|4|5|6|7|8}

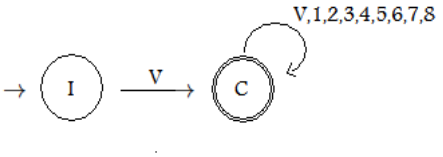


Fig. 6. State transition diagram representing the automata

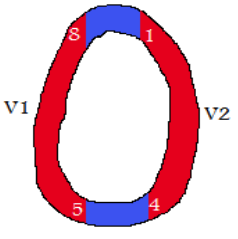


Fig. 7. Ambiguity case. Solution: repeat the strokes are connected. String representation: V5588V1144.

Therefore, it is possible to construct strings that always start with "V" and can be finished by any element defined in the alphabet of the grammar. A non-deterministic finite automata (AFDC) can be used to verify the validity of the constructed strings. This AFDC may be represented by (Fig. 6):

Sometimes, the ambiguity is possible. For example between 'e' and '6'. To solve this problem, the followed procedure consists in repeating the strokes that are connected. Fig. 7 shows the way to solve the ambiguity problem in the case of the character 'o'.

2.5 Knowledge Base and Inference Engine

The representation of the string is added to the knowledge base as a solution to represent the character. It is stored in a XML file with a tree structure using a XML-SCHEMA format.

Each node of the tree will be any variation in the searched character. Leaves define the character being sought in inference engine. This tree is constructed from the entries of a writer and will be used as inference engine. So when we

introduce a representative string, the search starts at root node and, each character of the string represents each node in the tree. The last character of the string will be a node which contain the character is searched. If the character is not found, the new variation on the tree is entered for future searches. Thus, the inference engine has the ability to learn new characters.

3 Experiments and Results

For experiments detailed in this section, two samples were collected from two different writers.

The first writer had to write 8 characters "a", then 8 characters "b" and so on until the "z". Only lower-case letters were considered. Then he did the same with the numbers: from 0 to 9. We got a sample of 296 images. Once obtained the sample, it is separated into two groups. The first group contains 6 characters of each type and the second the 2 remaining characters. The first group is used to generate three bases of knowledge and inference engine: one for numbers, one for letters and another for both. The second group, that contains the 2 remaining characters, is reserved for testing. The purpose of this second sample is to compare the results with other writers.

Second writer had to write 2 characters in the same way as the first writer thus obtaining a sample of 72 characters. This sample will be used only for testing.

In the first test, a number knowledge base and inference engine is used. Tests were made using the sample of each writer. Table 1 shows the results.

The second test, is similar to the previous one. The difference is that in this test had been used a knowledge base and inference engine developed with only letters (detailed in Table 1).

As we can see, results are best on numeric character. This is because numeric characters are simpler than alphabetic characters. On the other hand, the sample of character test is bigger than numeric, which gives more accurate results.

And finally, the third test uses a knowledge base and inference engine developed with both: alphabetical and numeric characters. Table 2 shows the results.

In the last test, results are best than other test above. It is possible because the sample in the last test, is greater than other two and consequently, gives more accurate result. This sample uses numeric and alphabetical characters to

Table 1. Hit, Miss and reject rates for numeric and alphabetic characters

Writer	Type	Hit (%)	Miss (%)	Reject (%)
Same writer	numeric	85,00	0	15,00
Different writer	numeric	70,00	5,00	25,00
Same writer	alphabetic	77,35	13,20	9,43
Different writer	alphabetic	62,26	22,64	15,09

Table 2. Hit, Miss and reject rates with numeric and alphabetical characters

Both Chars	Hit (%)	Miss (%)	Reject (%)
Same writer	85,13	14,86	0
Different writer	60,81	31,08	8,10

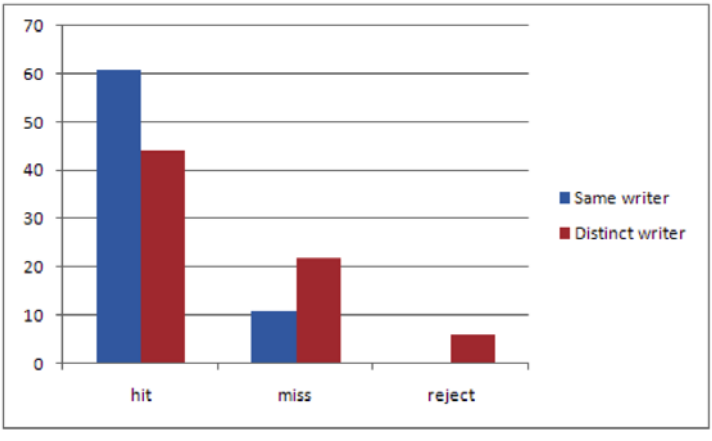


Fig. 8. Relation hit, miss and reject between same and distinct writer for numbers and characters. As we can see, there are more hit and lees miss rate from sample writer to other writer. This is because the sample is from first writer and is used as inference engine for second writer.

see how the inference engine works with bigger samples. The numbers of hits, misses and rejections for both writers are shown in Fig. 8.

4 Conclusions and Future Works

In this paper we propose a methodology to recognize handwritten characters.

The results indicate that it is possible to recognize handwritten characters from the identification of horizontal and vertical strokes and adjacency regions. The implementation of preprocessing, such as unslant and the use of morphological operations, reduce the ambiguity and improve outcomes.

Also, the use of an XML file to store the knowledge, reduces time that inference engine spent searching for the similar representative string to the handwritten character introduced, being practically zero. There are still ambiguities that complicate the recognition of characters such as between "5" and "S" or "0" and "O", which must be solved by considering the context of character.

It should also be noted that this algorithm can be used for all types of letters because anyone can create their own knowledge base by introducing a user-generated sample in the inference engine.

Finally, the algorithm provides satisfactory results around 87% of accuracy but it will be interesting to reduce time in preprocessing and even in processing to decrease the time spend in recognition.

The results obtained using this methodology are lower than other which use neuronal networks like Rafael Palacios and Amar Gupta[10] or Liana M. Lorigo and Venu Govindaraju[8], but the proposed methodology does not need to design a neural network.

As a future work, the use of the methodology presented in this document could be used on words recognition or even, in handwritten documents.

References

1. Alonso, A., Fernandez, R.-A., Garcia, I.: Recognition of Merged Characters Based on Vertical Strokes and Adjacency Regions (2005) (published at the V congress of Hispalinux)
2. Bharath, A., Madhvanath, S.: Hidden Markov Models for Online Handwritten Tamil Word Recognition, HPL-2007-108 (2007)
3. Shi, D., Damper, R.I., Gunn, S.R.: Offline Handwritten Chinese Character Recognition by Radical Decomposition. ACM Transactions on Asian Language Information Processing (2003)
4. Taira, E., Uchida, S., Sakoe, H.: Nonuniform Slant Correction for Handwritten Word Recognition. IEICE Trans. Inf. and Syst. (2004)
5. Tapia, E., Rojas, R.: A Survey on Recognition of On-Line Handwritten Mathematical Notation, Technical Report B-07-01 (2007)
6. de Zeeuw, F.: Slant Correction Using Histograms, Bachelor's Thesis in Artificial Intelligence (2006)
7. Shu, H.: On-line Handwritting Recognition Using Hidden Markov Models. Massachusetts Institute of Technology (2007)
8. Lorigo, L.M., Govindaraju, V.: Off-Line Arabic Handwriting Recognition. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2003) (to appear)
9. Khan, M.R.H., Hossain, G.: Bengali Handwritten Bank check Recognition Using Automatic Extraction of the User Entered Data. Islamic University of Technology (2005)
10. Palacios, R., Gupta, A.: Training Neuronal Networks for Reading Handwritten Amounts on Checks, Working Paper 4365-02 (2002)
11. Mathur, S., Aggarwal, V., Joshi, H., Ahlawat, A.: Offline Handwriting Recognition Using Genetic Algorithm. International Book Series "Information Science and Computing" (2008)
12. Desai, A.A.: Gujarati Handwritten Numeral Optical Character Reorganization Through Neural Network. Veer Narmad South Gujarat University (2010)
13. Corchado, E., Abraham, A., Carvalho, A.: Hybrid Intelligent Algorithms and Applications. Information Sciences 180(14), 2633–2634 (2010)
14. Pedrycz, W., Aliev, R.: Logic-Oriented Neural Networks for Fuzzy Neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
15. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. Neurocomputing 72(13-15), 2729–2730 (2009)
16. Pavlidis, I., Singh, R., Papanikolopoulos, N.P.: Recognition of On-Line Handwritten Patterns Through Shape Metamorphosis. University of Minnesota, Minneapolis (2002)

KETO: A Knowledge Editing Tool for Encoding Condition – Action Guidelines into Clinical DSSs

Aniello Minutolo[1,2], Massimo Esposito[1], and Giuseppe De Pietro[1]

[1] Institute for High Performance Computing and Networking, ICAR-CNR
Via P. Castellino, 111-80131, Napoli, Italy
[2] University of Naples "Parthenope", Department of Technology, Naples, Italy
{minutolo.a,esposito.m,depietro.g}@na.icar.cnr.it

Abstract. Clinical practice guidelines are expected to promote more consistent, effective, and efficient medical practices, especially if implemented in clinical Decision Support Systems (DSSs). One prerequisite for the broad acceptance of clinical DSSs and their efficient application to medical settings is the guarantee of a high level of upgradability and maintainability. In this respect, this paper proposes KETO (Knowledge Editing TOol), a user-friendly tool to guide and assist the editing and formalization of condition-action clinical recommendations into a hybrid Knowledge Base (KB), made of if-then rules built on the top of ontological vocabularies, to be then used in a clinical DSS. The tool aims at: i) synergistically combining multiple knowledge representation techniques for building efficient DSSs able to deal with different clinical problems; ii) reducing the complexity of the formalization process, by enabling the creation and automatic encoding into machine executable languages of hybrid KBs that could be functional in the context of clinical DSSs.

Keywords: Clinical DSS, Knowledge Base Editor, Ontology, Rules.

1 Introduction

In the last years, clinical practice guidelines have been more and more widely adopted since they are expected to promote more consistent, effective, and efficient medical practices and improve health outcomes when followed [1, 2]. Several recent studies have suggested that clinical practice guidelines without advanced Decision Support Systems (DSSs) (i.e. computer-based systems designed to help doctors to make decisions by providing motivated suggestions [3]) may not provide the promised improvements in patient safety or quality of care [4, 5]. Up to now, several DSSs have been widely studied [6, 7], by focusing on condition-action clinical rules rather than time-oriented guidelines. Condition-action clinical rules represent elementary, isolated care recommendations, which specify one or at most a few conditions which are linked to specific actions [8]. It is interesting to note that the most diagnostic and therapeutic clinical guidelines can be distilled in terms of a set of condition-action clinical rules, although this discards the control flow structure [9]. Current

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 352–364, 2012.

implementations of clinical DSSs, known as knowledge-based, encode condition-action clinical rules into a logical formalism for simulating the process followed by the physicians [10, 11]. The Knowledge Base (KB) is their key element, since it includes the corpus of relevant knowledge, coming from the clinical recommendations. They rely on inductive and deductive reasoning on the encoded knowledge and are strictly correlated to the knowledge representation formalism [12].

The task of building the KB of a DSS consists in the collection of the relevant condition-action clinical recommendations, its systematization and technical formalization. Typically, clinicians are not supposed to directly access the clinical recommendations encoded in the KB, but they can only ask for the assistance of the DSS, which can then decide to use the KB for its decision making process. It means that, the KB is not accessible and editable directly by clinicians and the cooperation of both clinical experts and experts in medical informatics is required to alter and update an existing KB [13]. Nevertheless, since this process of updating the KB can require a continuous intervention— clinical rules are often subject to change due to their evolution to implement medical progress in the treatment of individual diseases, or to the adaptation of generic, site-independent clinical rules to a patient to be treated [14] — it is unthinkable that it cannot be done directly by doctors when needed. Also, by supporting a direct access to the KB, doctors are encouraged to use clinical DSSs built on the top of it, since they are mostly entrusted with suggestions produced starting from their expertise, especially if inserted by them.

In contrast to the intensive efforts made to develop clinical knowledge-based DSSs, the issue of providing solutions for easily editing and upgrading the encoded knowledge has been widely neglected thus far. In this respect, this paper proposes an editing and visualization tool, named KETO (Knowledge Editing TOol), to guide and assist the creation and formalization of the KB to be used in a clinical knowledge-based DSS, with the aim of being mainly oriented to medical users. The key issue of KETO relies on graphical facilities offered for easily editing condition-action clinical recommendations into hybrid KBs, made of *if-then* rules built on the top of ontological vocabularies, with the remarkable goals of: i) synergistically combining different knowledge representation techniques for realizing efficient DSSs able to deal with different clinical problems; ii) reducing the complexity of the formalization process, by enabling the creation and automatic encoding into machine executable languages of hybrid KBs that could be functional in the context of clinical DSSs.

The rest of the paper is organized as follows. Section 2 introduces an overview of the state-of-the-art solutions for building KBs and addresses the motivations underlying the development of the proposed tool. Section 3 depicts design considerations for the construction of a DSS where the hybrid KB can be defined by means of KETO, while in Section 4 KETO editing functionalities and interfaces are described. Finally, Section 5 concludes the work.

2 Related Work and Motivations

Traditional methods to develop KBs for decision support implementations involve drawing out information from medical experts and range from informal or semi-structured interviews and observations between clinicians and technicians to more

structured methods, like the transcription and analyses of verbal reports, or conceptual techniques such as graph construction, etc. [15]. Thus, they result cumbersome and time consuming, generating the need for a simple and easy-to-use knowledge editing tool that clinical experts could directly use to encode their knowledge.

A number of studies has appeared in the last years, where different general-purpose tools for developing KBs have been analyzed and compared [16, 17]. These works have confirmed that existing solutions, which can be used for the KB creation, are typically general-purpose and provide many capabilities, at the price of being very complex, by typically speaking the language of knowledge representation formalisms which they support, rather than the language of the domain for which the KB has to be developed. This issue demands a deeper insight into the underlying formalisms, and thus, highlights the lack of a good usability [18]. It appears clear that doctors, who have knowledge and competencies to edit condition-action recommendations by creating the conceptualization of the domain and the decision making procedures, are not able to directly use these existing solutions to edit the KB.

Up to now, to the best of our knowledge, none of the existing tools is directly devised to medical applications and, in particular, mostly concerned with the editing of condition-action clinical guidelines, neither system-oriented researches appear to have been developed in that direction. As a result, the most important issue emerged is the need for user-friendly interfaces aimed at facilitating the use of knowledge editing tools by medical experts rather than by technical experts. This means that the advanced tools widely used in the knowledge engineering community are not intended to be replaced or outperformed. Differently, in order to simplify the knowledge editing in the medical setting, only the subset of facilities needed to build clinical rules on the top of a very simple collection of useful terms should be offered so as to reduce complexity at the cost of functionality, and, in addition, it should be exposed in a very simple and familiar fashion.

3 Clinical DSS Design Considerations

A clinical DSS solves the requests of monitoring and assessing the patients' health status by aiding different activities, i.e. diagnosis, prognosis, therapy and follow-up. It uses patients' heterogeneous information (i.e. anamnesis, diagnostic parameters, etc.) retrieved from a variety of data sources, such as user interfaces, clinical or hospital information systems, etc., and applies different types of reasoning, if needed.

Requirements for the broad acceptance of a clinical DSS and its efficient application to medical settings regard the capability of: i) modeling medical knowledge in a structured, coherent and flexible manner; ii) making useful inferences based on condition-action clinical guidelines and providing some level of transparency regarding the mechanisms for reaching such inferences; iii) providing specific explanations for these medical inferences; iv) providing some mechanisms to conveniently update and maintain the DSS with respect to medical progress or adaptation in the treatment of individual diseases.

Thus, the way the knowledge is represented is one of the most key facets for facing such requirements and, thus, having a successful clinical DSS. However, research in knowledge-based DSSs has recently dealt with the problem of overcoming the limitations imposed by a single knowledge representation language. According to the growing awareness that the combinations of intelligent techniques frequently perform better than the individual ones [19, 20], hybrid approaches have thus been proposed, with two or more subsystems dealing with specific portions of the knowledge base and using specific representation formalisms.

With such anawareness in mind, the method of representing medical knowledge in terms of a hybrid KB, made of *if-then* rules built on the top of ontological vocabularies, has appeared the more suitable way to model a very complex domain, such as the medical one. Moreover, this approach is also generally accepted in medical settings, since easily usable and understandable also by a non-technical audience [21]. Thus, the hybrid KB buildable by means of KETO, in the context of a clinical knowledge-based DSS, consists of two distinct portions, dealing with *declarative knowledge,* i.e. the structure of the domain knowledge involved in the guidelines, and *procedural knowledge,* i.e. the knowledge about the decision making procedures. In particular, the procedural knowledge consists in a set of *if-then* rules, each of them being a collection of conditional statements in the form of "*if antecedents then consequents*". The antecedent part consists in one or more statements, composed of ontological terms, concatenated by conjunctive or disjunctive connectives. Negation of a statement is not foreseen, whereas *negation-as-failure* can be used to infer negative answers based on the explicit absence of a statement. The consequent part of a rule consists in one or more statements, made of ontological terms, concatenated by conjunctive operators.

In order to guarantee the simplicity of the rule, the declarative knowledge, which can be defined by means of KETO in form of ontology, includes the basic terms composing the domain knowledge structure, the properties of the terms, i.e. attributes, and the relationships between these terms. The choice of using an ontology is due to the possibility of defining a vocabulary semantically, by specifying a set of modeling primitives, such as axioms about relations and attributes, which can be proficiently applied to guide the rule construction by reducing the possibility of editing errors. Moreover, an ontology can be understood as mean to quickly and simply share and reuse knowledge since intended to provide users with reusable pieces of declarative knowledge, which can be – together with procedural knowledge and reasoning services – assembled into high-quality DSSs in a timely and proficient fashion.

The knowledge representation formalisms adopted in KETO for implementing the hybrid KB are OWL and Jena rule language, respectively.

In more detail, OWL (Web Ontology Language) [22] has been chosen since it is the de-facto standard solution for providing 'semantics' and granting formality and expressivity. On the other hand, Jena rule language [23] has been identified as the most appropriate for writing rules on the top of terminological elements defined in OWL ontologies due to its syntax, which is not only concise and very expressive, but also simply understandable for non-technical users.

The reasoning engine used in a so built DSS is based on a forward chaining scheme, i.e. a data driven method that can be described logically as repeated application of the generalized modus ponens [24]. In other words, available data are

supplied to the engine as facts encoded as OWL statements and used to evaluate eligible rules and draw all possible actions/conclusions.

4 KETO Functionalities

The driving philosophy for KETO is to provide a simple and intuitive interface to the clinicians who do not have a deep technical expertise about hybrid KBs made of ontologies and rules. Additionally, more technical information can be also provided to ontology experts, if required. Following this idea, KETO has been devised as a more general working framework, where, for the sake of simplicity, a higher abstraction level of the ontology and rule constructs has been used by means of a visual interactive representation which reduces, where necessary and possible, complexity without losing completeness.

This simplicity of use contextually involves the correctness of the rules edited, by granting different levels of consistency in their development: from the syntactical composition of a rule, by supporting the correct encoding in terms of admissible rule structure and well-formed statements, to its verification in terms of logic integrity by ensuring it fires when it should. In the following, more details about the interfaces and facilities defined for encoding hybrid KBs are given. In order to better explain these facilities, some very simple examples are described, which regard some condition-action clinical recommendations defined for both monitoring patients suffering of cardiovascular diseases and detecting abnormal situations [25]. The set of terms involved in this case study and used in the following are described in Table 1.

Table 1. List of terms pertaining to the case study

Term	Description
Patient Summary	A summary of all the information pertaining to a patient.
HeartMonitoringInformation	The heart monitoring information regarding a patient.
PhysicalActivityInformation	The physical activity performed by a patient.
PostureInformation	The patient posture.
AlertInformation	The information needed to report an alert.
measuredHeartRate	Heart rate measurement acquired by a patient.
restingHeartRate(RHR)	The patient's heart rate at rest.
RHRmax	RHR+δ
RHRmin	RHR- δ
walking	It indicates if a patient is walking or not.
running	It indicates if a patient is running or not.
standingUp	It indicates if a patient is standing up or not.
lying	It indicates if a patient is lying or not.
alertExplanation	The explanation of the abnormal situation detected.
alertType	The severity of a generated alert.

4.1 The Knowledge Editing Interface

The overall KETO interface is shown in Figure 1, and it is organized as a Knowledge Tree (the left area) and a Rule Editing Interface (the right area).

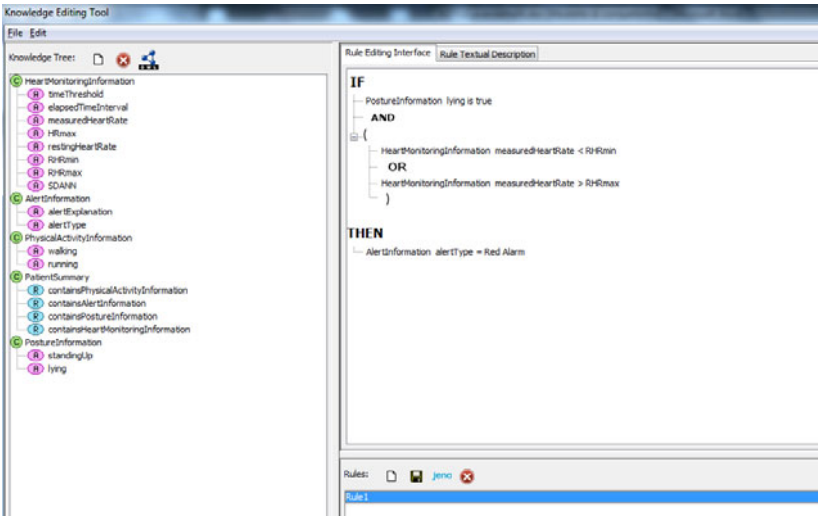


Fig. 1. The overall knowledge editing interface

First of all, the Knowledge Tree contains the vocabulary to be used in the rules, i.e., concepts, attributes, and relations, which can be either created brand new or imported from a source file. It has been designed with two main goals: i) to show all the concepts defined in the underlying ontology, by highlighting, for each concept, its attributes and inter-relations; ii) to lead the user in the process of composing well-formed rule statements by hiding formalisms and constraints.

On the other hand, the Rule Editing Interface is a user-friendly interface devised to assist the process of building condition-action rules by introducing an intuitive graphical representation of them, as illustrated in Figure 2.

As discussed in the previous section, each if-then rule is made of two parts, namely the antecedent and consequent parts. The antecedent part is graphically arranged as a nested tree, where its root is the node "*IF*" and the other nested nodes can be respectively a statement, a logical conjunctive/disjunctive connector or a couple of circle brackets. Statements included into a couple of circle brackets, which are associated to a higher evaluation priority, are arranged and visualized as nodes placed in a nested level with respect to the brackets including them.

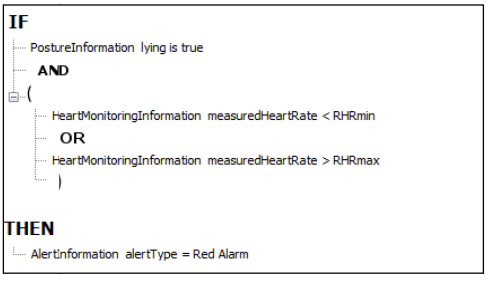


Fig. 2. Graphical representation of a clinical recommendation in the form of an *if-then* rule

Similarly, the consequent part is graphically arranged as a nested tree, where its root is the node "*THEN*" and the other nested nodes can be respectively a statement or a logical conjunctive connector. Since the consequent part of a rule consists in a conjunction of statements, as previously described, neither opened/closed circle brackets nor logical disjunctive connectors are admitted.

Figure 3 depicts the interfaces to create concepts, relations and attributes. In particular, the interface in the left part of Figure 3 illustrates how to create a new concept. The doctor has to only type the name of the concept. By default, each concept is automatically defined as specialization of the OWL concept *Thing;* otherwise the doctor can also indicate another concept as super-class by choosing between the concepts previously defined. Similarly, the creation of a relation or an attribute requires first the insertion of its name and, then, the specification of the associated domain and range, which are two concepts in the case of a relation and are respectively a concept and a data type in the case of an attribute (see the interfaces in the middle and right parts of Figure 3, respectively).

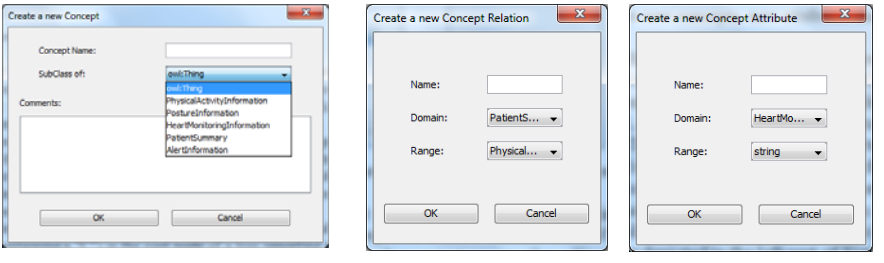


Fig. 3. Interfaces to create new concepts, new attributes and relations

It is worth noting that during the process for creating concepts, relations and attributes, the user is not required to know and use the OWL formalism. Moreover, in order to further support also expert users, the editor also let to visualize the description of the OWL ontology currently edited, automatically computed.

In the Knowledge Tree, the vocabulary is reported as a list of concepts and properties and arranged in the form of nested trees. Each available relation (attribute) is visualized under a concept only if the concept itself is contained in the domain restriction of the considered relation (attribute). In this way, the same relation (attribute) can be visualized under several concepts in the Knowledge Tree.

In particular, as better depicted in the left part of Figure 4, each concept (indicated with a small green circle containing a "C") is the root node of a tree reporting the list of properties, which can be both relations and attributes (denoted by small cyan circles containing a "R", and small pink circles containing an "A", respectively), whose domain is represented by the concept itself. Each property is arranged as a child node of this tree and visualized in a nested level with respect to its parent, i.e. the concept it is associated to. The OWL representation of this vocabulary, automatically generated, is shown in the right part of Figure 4.

The editing of a rule requires the use of this vocabulary to compose the statements to be inserted into the antecedent or consequent parts. Such statements, which can

involve instances, classes, relations and attributes, will be congruent and verifiable only if domain and range restrictionson relations and attributes are observed. In this respect, users are forced to always select either a relation or an attribute in the Knowledge Tree so as to grant the correctness of statements produced.

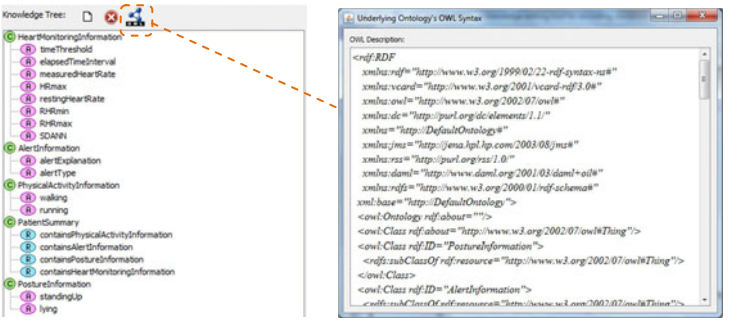


Fig. 4. The vocabulary arranged in the form of nested trees, and its corresponding OWL syntax

In particular, KETO implements a very simplified and intuitive drag-and-drop technique for adding respectively either a relation or an attribute by exploiting domain and range restrictions about relations and attributes stored into the ontology.

For example, the insertion of the statement "*PostureInformation lying is true*"can be carried out by selecting the attribute "*lying*" in the nested tree under the root concept "*PostureInformation"*, and, then, by dragging and dropping it in the rule editing area under the root "*IF*". Contextually, the interface in Figure 5 is visualized for completing the statement to be inserted. Since the attribute selected is associated to a concept through the specification of its domain, in the interface both the subject and the property of the statement appear already filled (and, thus, disabled). Moreover, according to the data type specifically expressed in the range associated to the attribute, KETO suggests the most appropriate list of operators to be used.

In the case reported in Figure 5, since the attribute "*lying*" can assume boolean values, the list of admissible operators which can be inserted by the doctor consists in *"isNot"*, *"is"* (in mathematical notation *"!="* and *"="*, respectively) and "*noValue*".

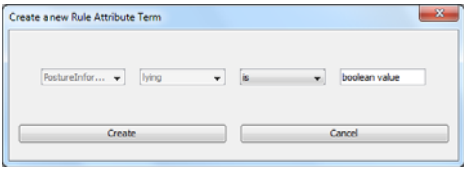


Fig. 5. The interface for adding a statement involving an attribute of a concept

Finally, the user can indicate the specific value the attribute has to assume in the statement by typing it into the last text area where the string "boolean value" is printed. In the case when the "*noValue*" operator is selected, no value must be

specified and the statement generated, i.e. "*noValue* (*PostureInformation lying)*" must be intended according to the *negation-as-failure* semantics. Differently, the addition of the statement "*HeartMonitoringInformationmeasuredHeartRate<RHRmin*", which involves the comparison between the values assumed by two attributes of a concept, can be done by selecting the attribute" *measuredHeartRate*" in the same way just described for the previous situation. Since the statement involves the comparison between values assumed by two attributes, the user must type no value into the last text area where the string "float value" is printed (the upper part of Figure 6).

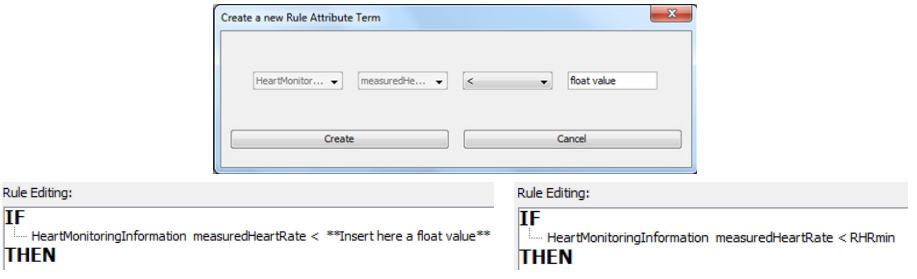


Fig. 6. The insertion of a statement involving the comparison between two attributes

This operation generates a partial insertion, as shown in the bottom left part of Figure 6, which can be completed by selecting the attribute "*RHRMin*" and, then, dragging and dropping it in the rule editing area over the string "Insert here a float value", as finally outlined in the bottom right part of Figure 6 (note that the comparison between the values assumed by two attributes is allowed only if their data types are the same). In the case when two statements regarding the same concept are put in the antecedent part of a rule (see the upper part of Figure 7), by default, the editor labels the second occurrence of the concept with a progressive id number appended at the end of its name for indicating that it refers to a different instance of the same concept. As reported in the lower part of Figure 7, a right-click on a selected statement allows to change the instance of the concept involved in it (note that if the same instance of a concept is involved into two different statements, no identification number is appended at the end of its name).

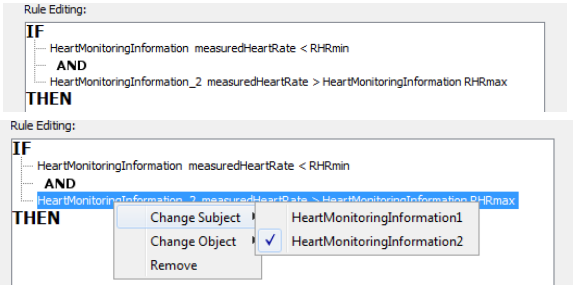


Fig. 7. The specification of different instances of a same concept in two different statements

Another kind of situation can occur when a statement involving a relation between concepts has to be added into a rule. For example, the statement "*PatientSummary containsPostureInformation PostureInformation*" can be entered by selecting the relation "*containsPostureInformation*" in the nested tree under the root concept "*PatientSummary"* and dragging and dropping it in the rule editing area under the root "*IF*". Contextually, the interface in Figure 8 is visualized for completing the procedure, where both the subject and the property of the statement appear already filled, since the relation selected is associated to a concept through the specification of its domain. Also, according to the range associated to the relation, the editor suggests the most appropriate list of concepts which can be used as object in the statement. In the example, the only admissible concept is "*PostureInformation*".

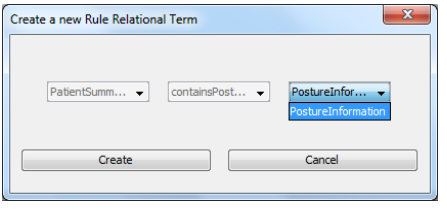


Fig. 8. The interface for entering a statement involving a relation between concepts

By default, a new statement added to the antecedent part of a rule, i.e. to the corresponding tree under the root *"IF",* is connected to the already existing ones by means of a conjunctive connector automatically inserted. By simply right-clicking on it, the connector can be changed. In the consequent part of a rule, as previously explained, only conjunctions are admitted and are added by default when a new statement is appended at the corresponding tree under the root *"THEN"*.

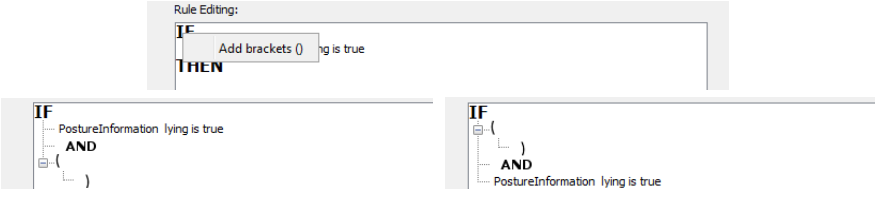


Fig. 9. The insertion of a couple of brackets in the antecedent part of a rule

Moreover, the addition of a couple of empty circle brackets in the antecedent part of a rule, in order to specify a different evaluation order for the statements, can be intuitively done by right-clicking on the root "IF" (see the upper part of Figure 9).

The couple of brackets, where the first open bracket is always followed by a closed one to avoid insertion errors, is generated empty and placed after the last existing child node (see the bottom left part of Figure 9). Successively, it can be dragged and dropped in order to be placed in a different position along the tree (see the bottom right part of Figure 9). Then, it can be filled by means of the appropriate group of

statements, which can be inserted ex-novo or moved from another position in the tree since already entered previously.

Indeed, it is important to note that statements added under the corresponding tree both in the antecedent or the consequent part of a rule can be moved and placed in other positions by means of drag-and-drop operations.

After inserting a rule, it can be saved, so as to appear in the bottom side of the Rule Editing Interface. Selecting and pressing on a rule in this list allows to change the rule currently opened. By clicking on the button named "Jena" placed in the bottom side of the Rule Editing Interface (see Figure 1), the encoding of the currently edited rule into the Jena rule language is automatically generated (see Figure 10), with the final aim of being then used in a clinical DSS for implementing the reasoning procedure.

Finally, KETO includes a reasoning engine, which simulates an actual DSS for testing rules inserted by means of KETO and automatically translated into the Jena rule language. A file containing clinical data encoded as OWL statements can be built to test the integrity of the logic and ensure rules fire when they should.

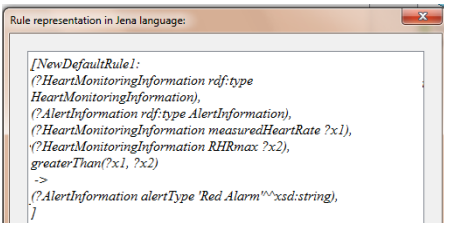


Fig. 10. The rule representation in Jena language

5 Conclusions

The paper described KETO an editing tool to guide and assist the creation and formalization of condition-action clinical recommendations. The key issue of KETO relies on graphical facilities offered for easily inserting and editing clinical recommendations into a hybrid KB, made of *if-then* rules built on the top of ontological vocabularies.

The basic design rationale used in the development was: i) to synergistically combine multiple knowledge representation techniques for building efficient DSSs able to deal with different clinical problems; ii) to reduce the complexity of the formalization process, by enabling the creation and automatic encoding into machine executable languages of hybrid KBs that could be functional in the context of clinical DSSs. KETO was entirely programmed in Java in accordance with the object-oriented paradigm, which makes it easy to maintain and extend.

Preliminary tests were performed with a team of volunteer students in medicine, with average computer skills, who were asked to develop rules for a simulated clinical DSS. The average satisfaction of the students in operating with KETO gave a first proof of its usability, suggesting that it could effectively support the simple insertion

of guidelines for actual clinical DSSs. Next step will regard the definition of a more extensive and punctual usability evaluation involving actual doctors.

Moreover, future work will also regard the improvement of KETO by means of new intuitive and user-friendly facilities to handle vagueness in condition-action clinical recommendations. Indeed, clinical rules the doctors have in mind are often vague in nature, since they are used to formulating their expertise at a high level of abstraction, in form of smooth linguistic labels, rather than as expressions with clear-cut boundaries. Next steps in this direction will regard the application of Fuzzy Logic to model vagueness in clinical recommendations, by focusing on the possibility of building fuzzy rules on the top of ontological concepts and properties.

References

1. Woolf, S.: Practice guidelines, a new reality in medicine: II. methods of developing guidelines. Archives of Internal Medicine 152(5), 946–952 (1992)
2. Scott, I.: What are the most effective strategies for improving quality and safety of health care? Internal Medicine Journal 39(6), 389–400 (2009)
3. Shortliffe, E., Cimino, J.: Biomedical informatics: computer applications in health care and biomedicine. Springer, New York (2006)
4. Linder, J.A., Ma, J., Bates, D.W., Middleton, B., Stafford, R.S.: Electronic health record use and the quality of ambulatory care in the United States. Archives of I. M, pp. 1400–1405 (2007)
5. Himmelstein, D.U., Wright, A., Woolhandler, S.: Hospital computing and the costs and quality of care: a national study. American Journal of Medicine 123(1), 40–46 (2010)
6. Niemi, K., Geary, S., Quinn, B., Larrabee, M., Brown, K.: Implementation and evaluation of electronic clinical decision support for compliance with pneumonia and heart failure quality indicators. American J. of Health-System Pharmacy 66(4), 389–397 (2009)
7. Brokel, J., Shaw, M., Nicholson, C.: Expert clinical rules automate steps in delivering evidence-based care in the electronic health record. C. Inf. Nursing 24(4), 196–205 (2006)
8. Shiffman, R.: Representation of clinical practice guidelines in conventional and augmented decision tables. J. of the American Medical Informatics Association 4(5), 382–393 (1997)
9. Peleg, M., Tu, S., Bury, J., et al.: Comparing computer-interpretable guideline models: a case-study approach. J. of the American Medical Informatics Assoc. 10(1), 52–68 (2003)
10. Colantonio, S., De Pietro, G., Esposito, M., Machì, A., Martinelli, M., Salvetti, O.: Decision Support for the Remote Management of Chronic Patients. In: Proc. of the 2nd Internat. ICST Conference on Wireless Mobile Communication and Healthcare, Kos Island, Greece (2011)
11. Colantonio, S., De Pietro, G., Esposito, M., Machì, A., Martinelli, M., Salvetti, O.: Knowledge Based Decision Support For The Management Of Chronic Patients. In: Proc. of the Int. Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Paris, France (2011)
12. Minutolo, A., Esposito, M., De Pietro, G.: A Mobile Reasoning System for Supporting the Monitoring of Chronic Diseases. In: Proc. of the 2nd International ICST Conference on Wireless Mobile Communication and Healthcare, Kos Island, Greece (2011)
13. Shahar, Y., Young, O., Shalom, E., Galperin, M., Mayaffit, A., Moskovitch, R., Hessing, A.: A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools. J. of Biomedical Informatics 37(5), 325–344 (2004)

14. Duftschmid, G., Miksch, S.: Knowledge-based verification of clinical guidelines by detection of anomalies. Artificial Intelligence in Medicine 22(1), 23–41 (2001)
15. Cooke, N.J.: Varieties of knowledge elicitation techniques. I. J. of HC Studies 41, 801–849
16. Escórcio, L., Cardoso, J.: Editing Tools for Ontology Construction. Semantic Web Services: Theory, Tools and Applications. Idea Group
17. Gómez-Pérez, A., Ortiz-Rodriguez, F., Villazón-Terrazas, B.: Ontological Engineering. Springer, Heidelberg (2004)
18. Garcia-Barriocanal, E., Sicilia, M.A., Sanchez-Alonso, S.: Usability evaluation of ontology editors. Knowledge Organization 32(1), 1–9 (2005)
19. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
20. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
21. Straszecka, E.: Medical Knowledge Representation in Terms of IF-THEN Rules and the Dempster-Shafer Theory. Artificial Intelligence and Soft Computing, 1056–1061 (2004)
22. OWL2 (Web Ontology Language), http://www.w3.org/TR/owl2-overview/
23. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: Proc. of the 13th Internat. World Wide Web Conference on Alternate Track Papers & Posters, New York, USA, pp. 74–83 (2004)
24. MacKay, D.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003)
25. Minutolo, A., Sannino, G., Esposito, M., De Pietro, G.: A rule-based mHealth system for cardiac monitoring. Biomedical Engineering and Sciences (IECBES), 144–149 (2010)

Integration of Intelligent Information Technologies Ensembles for Modeling and Classification

Andrey Shabalov, Eugene Semenkin, and Pavel Galushin

Institute of Computer Science and Telecommunications, Siberian State Aerospace University, Krasnoyarsk, Russia
shabalov-andrey@mail.ru, eugenesemenkin@yandex.ru, galushin@gmail.com

Abstract. Intelligent information technologies help us to solve complex data mining problems and therefore they are of particular interest. However, a generation of a specific technology structure demands high skills of a developer and this process is time-consuming as well. In this paper, we present an automated integration of intelligent information technologies for complex systems modeling and classification. We consider such popular techniques as neural networks, fuzzy rules based systems and neuro-fuzzy systems as well as evolutionary algorithms for automated generation. We also propose a new idea of genetic programming application to the design of intelligent information technologies ensembles for effectiveness and reliability improvement.

Keywords: Neural networks, fuzzy rules based systems, neuro-fuzzy systems, evolutionary algorithms, ensemble, modelling, forecasting.

1 Introduction

Complex systems control and design are impossible without appropriate models. However, a modeling of complex systems in a standard way is rather complicated if not possible. Systems simulating models can be helpful here. For many real world problems we can see a following situation. There is a big data base of the results of the complex system behavior observations but appropriate model of this system is not yet clear. Here we can use intelligent information technologies (IIT) to obtain the first stage model within short time in order to simulate the system and learn its properties that gives us a possibility to develop a full profile model of the system. However, the design of IIT can also be a problem. Currently, intelligent systems have got wide propagation in various fields of human activity connected with complex systems modeling and optimization. Artificial neural networks [1], fuzzy logic [2], neuro-fuzzy systems [3], evolutionary algorithms [4] and other techniques and technologies are popular tools for investigation due to their capability to solve complex intelligent problems that are difficult to solve with classic techniques [5].

In the most applications of intelligent methodologies, usual way is to use the most proper approach for each field of study. Within this approach, a successful application of an intelligent technique corresponds to the comparison of the performance of some

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 365–374, 2012.

competitive intelligence techniques in contrast to the proposed one. If the latter outperforms the other techniques then it proves the correctness of applied approach. The highly increasing computing power and technology made possible the use of more complex intelligent architectures, taking advantage of more than one intelligent technique, not in a competitive, but rather in a collaborative way. This is called a hybrid computational intelligence methodology and is an effective combination of intelligent techniques that outperform or compete to simple standard intelligent techniques.

The popularity of hybrid intelligent techniques is due to their extensive success in a wide range of real-world complex problems. The main reason for this success is assumed to be the synergy derived by the computational intelligent components, such as fuzzy logic, neural networks, genetic algorithms, or other intelligent heuristics. Each of these methodologies provides hybrid systems with complementary reasoning and searching methods that allow a usage of empirical data to solve complex problems [6].

In this paper, authors use genetic programming algorithm to determine effective way for combination of single intelligent techniques (neural networks, fuzzy or neuro-fuzzy systems) designed via evolutionary algorithms.

Intelligent information technologies design is a complex optimization problem whose structure doesn't allow solving it effectively with classic techniques. GAs are parallel, robust search procedure based on natural selection principles and evolution as well as genetics. The population of individuals representing solutions adapts to environment during evolution by means of genetic operators such as selection, recombination, crossover and mutation thus maximizing fitness (i.e. minimizing cost function) [7]. GAs have demonstrated high performance in the solution of practical problems with multimodal objective functions. A coding structure and the independence of quality measure properties make them an appropriate tool enabling to incorporate a priori knowledge of an investigated object or a process despite their capability to find suboptimal solutions in complex spaces [8].

A genetic programming algorithm operates computer programs expressed by trees. The required purpose is achieved by growing up trees population using both the principle of survival of the fittest and genetic operations (selection, recombination, mutation and others) [9]. To solve the specified problem it is necessary to define a functional set (a set of used functions) and a terminal set (a collection of function variables, constant types in use) that must have properties of closure and sufficiency.

Currently, ensemble techniques have been applied in many regression and classification tasks. It has been observed that diversity of members making up a "committee" plays an important role in an ensemble approach [10]. Different techniques have been proposed maintaining the diversity among members by running on different feature sets [11] or training sets (e.g. bagging [12] and boosting [13]). Some techniques as neural networks can be run on the same feature and training sets producing the diversity by different structures [14]. Simple averaging, weighted averaging, majority voting, and ranking are common methods usually applied to calculate the ensemble output.

In [15] a Mamdani fuzzy inference system was used to combine outputs of several techniques (Fuzzy KNN, Multi Layer Perceptron with Gradient Descent with Momentum Backpropagation, and Multi Layer Perceptron with Scaled Conjugate Gradient Backpropagation). Amorim Neto et. al. applied genetic algorithm for choosing

definite neural networks from pre-generated set according to the performance metrics [16]. Siwek et. al. [17] used 4 neural-like predictors (Multilayer Perceptrons (MLP), Support Vector Machines (SVM), Elman Networks, and Radial Basis Functions Networks). The obtained results post processed by SVM or MLP. In [18] a local zeroth, linear, quadratic, 3rd-order polynomial and MLP neural network were used as local predictors and MLP with Levenberg-Marquardt as global one. The simple averaging and dynamic averaging were applied for deriving the terminal result. Johansson et. al [19] used genetic programming for building an ensemble from predefined number of Artificial Neural Networks where functions were averaging and multiplying and terminals were models and a constant. In [20] a similar approach is proposed where a specified number of neural networks is generated. Then a genetic programming algorithm is applied to build an ensemble making up symbolic regression from partial decisions of the specific members. By that, an ensemble represents a homogeneous structure.

In this paper we propose a new idea of applying a genetic programming technique to building an ensemble of models of different kinds providing by that the diversity among members within the ensemble. Another peculiarity of our work consists in that these models are generated automatically by means of genetic algorithm.

The article is organized as follows. Section 2 describes an approach to IIT automated design, Section 3 presents an idea of IIT integration into ensembles using the genetic programming techniques, Section 4 shows practical results of applied problems, in the Conclusion outcomes of the work are done as well as future perspectives are discussed.

2 The Automated Design of Intelligent Information Technologies

Connectionist models. The multilayer perceptron is widely spread in different fields. As a rule the learning of this structure is carried out in terms of error back-propagation. The complexity of design consists in the initial choice of hidden neurons number and neurons number on each hidden layer whose structure is preliminary unknown as well as in neurons activation functions choice. The drawbacks of a back-propagation algorithm are low convergence speed, sensitivity to noise, the dependence of functioning quality on step heuristics, and the fact that a modeling error as a rule doesn't reach the global optimum due to its complexity [21].

To overcome such problems it is suggested to apply genetic algorithms for perceptron structure generation and weights coefficients tuning.

The structure of a chromosome is coded as follows. Initially the maximum number of hidden layers and the maximum number of neurons on each hidden layer are preset by a researcher. While coding the structure, the pass in the network is carried out downwards and from the left to the right for each neuron. The length of each chromosome is from 4 to 5 bits in the condition of using a set of 8 or 16 activation functions accordingly. The first bit shows the presence or absence of a neuron in the net, all the rest code the information of a number of an activation function. Thereby the chromosome length coding the neural network structure is equal to maximum neurons number multiplied by chromosome length of one neuron.

The weights coefficients are coded in the same way. The interval of weights coefficients changing and accuracy (digitization of a number) arepreset by a researcher and determines the bits number n for real number coding. The chromosome length coding the weights coefficients equals number n multiplied by the number of all coefficients of a current net.

Hereby, it is necessary to generate the population of individuals which represents the neural network structures and for each structure – a separate population of weights coefficients. The algorithm stops if there is apreset number of structure generations achieved or modeling error (usually a root-mean-square error) is small enough.

Fuzzy rule based systems. While developing a fuzzy system an expert faces the problem of initial fuzzy rules selection a set of which could be incomplete and contradictory. The selection of membership functions parameters describing the input and output object parameters is carried out subjectively and may represent the reality incorrectly [2].

Therefore, to increase the validity of decision making while designing fuzzy systems the GAs were applied. The Pittsburgh approach was used for generation of a fuzzy system structure[22], i.e. each individual presents the whole rule base. First it is necessary to preset the maximum rules number $N_{\max}$. During the learning process the following knowledge base parameters are adjusted: membership functions parameters determining the center and width of terms, number of rules, the overall length of the whole base due to the inclusion of a term– "don't care" [23]. Figures 1 and 2 show the Mamdani and Takagi-Sugeno (zero order) coding schemes accordingly, where the following notations a taken: R_k is the k -th rule, a_i , σ_i and $a_j^{'}$, $\sigma_j^{'}$ are parameters of Gaussian membership function (center and width), n is the number of input variables and m is the number of output variables, v_j is a singleton, $i=\overline{1,n}$, $j=\overline{1,m}$.

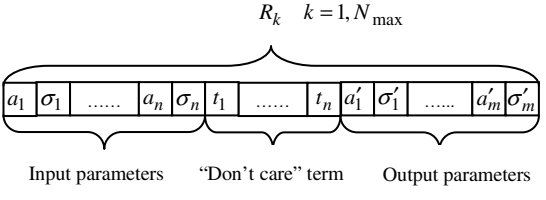


Fig. 1. Mamdani chromosome coding scheme

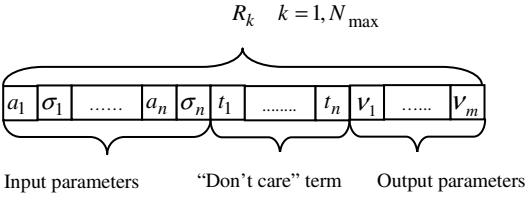


Fig. 2. Takagi-Sugeno chromosome coding scheme

Each parameter of center and width is presented by a bit-string. The coding procedure of these parameters is the same as described in "Connectionist models" for weight coefficients coding.

Neuro-fuzzy systems. The generation process of neuro-fuzzy systems consists of two phases [24, 25]. The first stage (unsupervised mode) represents the initial numerical data clustering based on competitive learning with rival-penalized method, adaptive resonance theory, etc. After that we get coarse fuzzy rules. The second stage (supervised mode) consists in accurate tuning of the rule base. Usually gradient algorithms are used here but their drawbacks are widely known and prevent effective use of neuro-fuzzy systems. Therefore, the GAs are applied instead of gradient algorithms. Their effectiveness in practical problems solving was shown in previous work and surpassed the steepest descent method in terms of modeling error [26]. The coding procedure of the parameters is the same as described in "Fuzzy rule based systems".

3 Genetic Programming Method For IIT Ensembles Designing

For effectiveness and reliability improvement of IIT it is suggested to apply the genetic programming method in order to form both IIT ensemble composition for complex problems solving and the way of cooperation of ensemble members in making the resultant decision based on particular decisions of individual technologies.

There exist two variants of IIT hybridization in this approach. The first one consists in mathematical expression built from decisions of individual members. Thus partial decisions of individual systems are terminal set elements of genetic programming method. The second variant is the formation of a hybrid multilayer system consisting of certain members of this ensemble. Here the terminal set is presented by IIT structure.

In figure 3 we can see the tree coding, figure 4 shows corresponding decisions for the first approach and for the second one. To denote particular objects on the figures, the following terms are used: ANN – artificial neural network, FIS – fuzzy inference system, NFS – neuro-fuzzy system.

In applying the first scheme it is necessary to generate and train in advance the specified number of terminal set elements which later will be used in the algorithm. In this scheme, there exist two modes of mutation realization in the genetic programming algorithm. It is possible either to choose randomlyan element from the terminal set or to generate an absolutely new intelligent system. In this paper the latter one is realized. In applying such an approach, a functional set includes mathematical expressions.

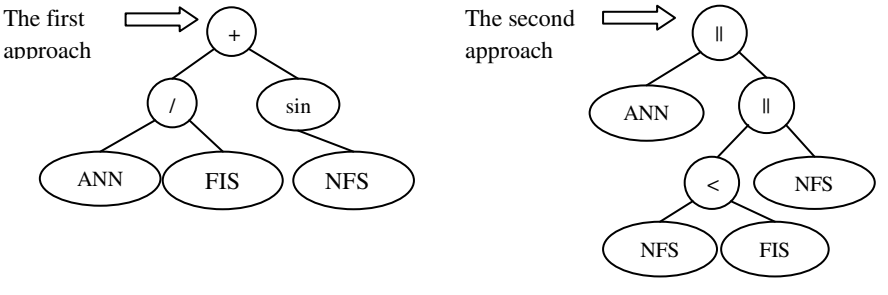


Fig. 3. Treecoding

The first approach

$$\frac{ANN}{FIS} + \sin(NFS)$$

The second approach

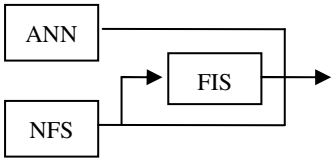


Fig. 4. Decisions representation

While implementing the second approach it is necessary to specify the number of cycles of learning for every technology. After generation of a new population the new structures arise as a result of the fact that every individual technology is designed anew. The elements of a functional set in this case are the rules of interconnections among certain technologies. For example, the symbol "||" means the systems are located parallel to each other. The symbols "<" and ">" aligns the models in accordance with a following order (e.g. "S1<S2" means S1 follows S2 or, in other words, S2 is located in front of S1). The aggregation of the partial decisions is carried out with one of the standard ensemble techniques (averaging, majority vote, etc.).

4 Application to Real-World Problems

For application of suggested algorithmic core generation schemes of intelligent information technologies there was developed a program system for solving modeling and classification tasks.

When implementing a genetic algorithm, there was taken a GA with a modified operator of multi-parent uniform recombination instead of standard GA.

The parameters of genetic programming algorithm were set as follows: 100 individuals, rank selection and maximum depth of the tree – 10. The stopping criterion was the maximum number of generations equaled 1000.

In the Table 1, the list of real-world problems used for approbation of submitted methods is presented. A part of them was taken from machine learning repository UCI[27].

Problems 1, 2, 4 present a classification task and the rest constitute a modeling task (approximation).

20 runs of the program were carried out for every type of IIT. Every run gave some operable systems.

In the Table 2, the best results in terms of error are shown. Following terms are used in table 2: L – learning sample, T – test sample. In classification problems the number of misclassified features is presented as an error, in all the other problems it is the mean relative error.

The modeling error obtained by a neural network model was much worse comparing to fuzzy- and neuro-fuzzy system and thereby is not displayed in the table.

Table 1. Approbation of suggestedtechniques

	Problem	Input dimension	Output dimension	Sample volume	
				Learning	Test
	Machine learning repository UCI				
1.	Iris classification	4	3	135	15
2.	Wine classification	13	3	163	15
3.	Forest fires forecasting	12	1	477	40
4.	Satellite image classification	36	6	4435	2000
	Applied problems				
5.	Turbine condition monitoring based on forecasting of vibration signals	11	12	1000	400
6.	Ore-thermal process modeling	9	1	47	10
7.	The degradation prediction of electrical characteristics of spacecraft's solar arrays	7	4	177	20
8.	Test-based characteristics forecasting of jet engine	5	1	20371	2263

From the Table 2 one can see that in many cases a neuro-fuzzy system has the best quality of modeling. Besides, the effectiveness of all the types of IIT is congruous to known results.

Table 2. The results of real-world problem solving

№	*Artificial neural net*		*Fuzzy system*		*Neuro-fuzzy system*	
	Error					
	L	T	L	T	L	T
1	5	1	2	0	2	0
2	1	1	0	0	0	0
3	-	-	16.87%	19.61%	15.67%	17.5%
4	1.78%	1.79%	1.11%	1.11%	1.45%	1.46%
5	9.11%	9.14%	8.07%	8.09%	7.99%	7.97%
6	4.86%	4.97%	2.99%	3.01%	2.81%	2.92%
7	-	-	5.66%	7.66%	5.05%	5.87%
8	-	-	4.97%	5.01%	0.93%	0.95%

The examples of ensembles formation on the basis of mathematical expressions from partial decisions of individual technologies are given below. On the initial phase 10 IIT of each type were generated and trained.

Following formula was obtained in terms of wine classification:

$$C = \sin\left(NFS_4 \cdot \sqrt{e^{NFS_{10}}} \right) \tag{1}$$

where C – is the class number. Moreover a recognition error constituted 0% of both a learning sample and a test sample that is better than for neural net models and fuzzy models and comparable with neuro-fuzzy models. Furthermore certain individual IITs which surpass the rest technologies in modelling quality (in terms of error) were not included in the committee.

In ore-thermal process modelling problem the following expression was got:

$$Ni(\%) = NFS_{10} \cdot e^{\frac{FIS_6 \cdot e^{\frac{FIS_6}{NFS_9}}}{FIS_{10}}} \tag{2}$$

that determines nickel percentage content in waste slag. The relative error is equal to 2.21% for the learning sample and 2.33%for the test one that is better than for every individual IIT.

In the process of numerical experiments it was found that certain technologies being superior to others in modelling quality are not always presented in the terminal formula. A set of technologies includes technologies with different modelling quality whose ensembles affords to improve the effectiveness and reliability on the whole.

5 Conclusion

Thus, the program system that realizes the developed approach enables to generate neural net models, fuzzy- and neuro-fuzzy models automatically, i.e. it allows obtaining computational models that are appropriate for learning complex systems properties from observations or experimental data base. Using these models we can develop full profile model that describes the system in explicit form, e.g. normal mathematical model. If our system gives us fairly simple fuzzy inference system then it can be used for data mining and seeking for hidden dependences that are not clear directly from data base records. Automatic IIT ensembles formation allows improving reliability and effectiveness of a system. The obtained results are approved by solving of some real-world problems.

Certainly, computational efforts for implementation of described approach and model complexity are severely increasing compared each single learning model. However it is usual drawback of any ensembling when one has to implement each member of ensemble. There are no additional problems with our approach here. Let us, also, remark that usually when evolutionary generating single model one cannot just generate one model and finish. One has to generate some models to avoid the impact of evolutionary algorithm randomness and then choose the best of them. That is why the difference in computation efforts is not as great as one could imagine it. Advantages of ensembling are better performance and reliability that essentially compensates extra efforts. In fact, real additional computational effort for our approach is necessity to run genetic programming algorithm that combines single models outputs into an output of ensemble. Our experiments showed that it is less than efforts for

evolutionary generation of one single model, i.e., could be considered as acceptable disadvantage.

As about the model complexity, again our approach does not bring much extra drawback comparing with any other ensemble technique. Of course, computational model given by genetic programming algorithm might be much more complicated compared to usual ensembling methods, like weighted sum of outputs or voting. However, our experiments show that genetic programming algorithm never includes all possible single models into ensemble taking usually a few of them. As the greater part of ensemble computational complexity is given by computational efforts needed to calculate the output for each model, our approach has advantage upon usual ensembling methods that include in ensemble all available single models.

The further development of the system is aimed to the expansion of its functionality by including the other types of IITs (dynamic neural networks, Kohonen and Hopfield networks, decision trees, multi objective selection etc.). Another direction is an improvement of system adaptability with automation of evolutionary algorithms tuning and improvement of the method of ensembles formation.

References

1. Rojas, R.: Neural networks: a systematic introduction. Springer, Berlin (1996)
2. Yager, R.R., Filev, D.P.: Essentials of fuzzy modelling and control. Wiley, New York (1994)
3. Jang, J.-S., Sun, S.-T., Mizutani, E.: Neuro Fuzzy and Soft Computing. Prentice-Hall (1997)
4. Eiben, A.E., Smith, J.E.: Introduction to evolutionary computing. Springer, Berlin (2003)
5. Konar, A.: Computational Intelligence: Principles, techniques and applications. Springer, Berlin (2005)
6. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
7. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. Wiley Interscience, New Jersey (2004)
8. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Boston (1989)
9. Koza, J.R.: Genetic programming. The MIT Press, London (1998)
10. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Machine Learning 40(2), 139–158 (2000)
11. Ho, T.K., Hull, J.J., Srihari, S.N.: Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(1), 66–75 (1994)
12. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
13. Friedman, J.H., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2), 337–374 (2000)
14. Navone, H.D., Granitto, P.M., Verdes, P.F., Ceccatto, H.A.: A learning algorithm for neural network ensembles. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial 12, 70–74 (2001)

15. Ramírez, E., Castillo, O., Soria, J.: Hybrid System for Cardiac Arrhythmia Classification with Fuzzy K-Nearest Neighbors and Multi Layer Perceptrons combined by a Fuzzy Inference System. In: WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain (2010)
16. Amorim Neto, M.C., Tavares, G., Alves, V.M.O., Cavalcanti, G.D.C., Ing Ren, T.: Improving Financial Time Series Prediction Using Exogenous Series and Neural Networks Committees. In: WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain (2010)
17. Siwek, K., Osowski, S., Sowinski, M.: Neural predictor ensemble for accurate forecasting of PM10 pollution. In: WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain (2010)
18. Siek, M., Solomatine, D.: Multi-model Ensemble Forecasting in High Dimensional Chaotic System. In: WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain (2010)
19. Johansson, U., Lofstrom, T., Konig, R., Niklasson, L.: Building Neural Network Ensembles using Genetic Programming. In: International Joint Conference on Neural Networks, IJCNN 2006 (2006)
20. Bukhtoyarov, V., Semenkina, O.: Comprehensive evolutionary approach for neural network ensemble automatic design. In: Proceedings of the IEEE World Congress on Computational Intelligence, Barcelona, Spain, pp. 1640–1645 (2010)
21. Wasserman, P.D.: Neural computing: theory and practice. Van Nostrand Reinhold Co., New York (1989)
22. Herrera, F., Magdalena, L.: Genetic Fuzzy Systems: a Tutorial. CICYT (1995)
23. Ishibuchi, H., Nojima, Y.: Analysis of interpretability-accuracy trade-off of fuzzy systems by multiobjective fuzzygenetics-based machine learning. International Journal of Approximate Reasoning 44(1), 4–31 (2007)
24. Castellano, G., Fanelli, A.M.: A self-organizing neural fuzzy inference network. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Como, Italy, vol. 5, pp. 14–19 (2000)
25. Castellano, G., Fanelli, A.M.: Information granulation via neural network based learning. In: IFSA World Congress and 20th NAFIPS International Conference, Vancouver, Canada, vol. 5, pp. 3059–3064 (2001)
26. Shabalov, A., Semenkin, E., Galushin, P.: Automatized Design Application Of Intelligent Information Technologies for Data Mining Problems. In: Joint IEEE Conference "The 7th International Conference on Natural Computation & The 8th International Conference on Fuzzy Systems and Knowledge Discovery", Shanghai, China, pp. 2659–2662 (2011)
27. UCI Machine Learning Repository, http://kdd.ics.uci.edu/

Fusion of Modular Bayesian Networks for Context-Aware Decision Making

Seung-Hyun Lee and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, Korea
e2sh83@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Ubiquitous computing brings various information and knowledge derived from different sources, under which Bayesian networks are widely used to cope with the uncertainty and imprecision. In this paper, we propose a modular Bayesian network system to extract context information by cooperative inference of multiple modules, which guarantees reliable inference compared to the monolithic Bayesian network without losing its strength like the ease of management of knowledge and scalability. Moreover, to provide a lightweight updating method for highly complicated environment, we propose a novel method of preserving inter-module dependencies by linking modules virtually, which extends d-separation to an inter-modular concept to control local information to be delivered only to relevant modules. Experimental results show that the proposed modular Bayesian networkscan keep inter-modular causalities in a time-saving manner. This paper implies that a context-aware system can be easily developed by exploiting Bayesian network fractions independently designed or learned in many domains.

Keywords: modular Bayesian network, selective inference, virtual linking, inter-modular d-separation, context-aware decision making.

1 Introduction

Solving real world decision problems often requires the fusion of information and knowledge derived from different sources of data and evidence. Unfortunately, however, the fusion accompanies uncertainty and imprecision which are likely to be unexpected in designing process. These may cause problems in decision making mainly due to, for instance, a lack of information, error in measurements, or ambiguous meanings in criteria and assessments. This gives rise to the demand for novel methods and efficient techniques for managing and integrating various types of uncertainty within a coherent framework, so as to ultimately improve decision analysis in complex decision situations. These research issues have recently received considerable attention, especially Bayesian probabilistic approach is widely adopted to deal with these innate problems of decision making.

Bayesian network (BN) is a robust tool for practical problems which involve high level of uncertainty. But utilizing it in the large scale domains is difficult because

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 375–384, 2012.

considerable effort is put on designing and maintaining the network. Besides, it is unable to entirely apply on ubiquitous devices since lots of computation power and resources are required in the inference process. For these reasons modular approach is applied in several studies such as landmark detection with mobile devices [1] and distributed multiple sensor networks [2].

There have been many studies on developing context-aware application based on a probabilistic approach. Li *et al*. used a probabilistic model for active affective state detection of a user [3]. They utilized dynamic Bayesian network and the utility theory in order to reason the user's states like fatigue, nervousness, and confusion. They showed the probabilistic approach could handle the information in uncertainty. Table 1 shows the comparison among some related works.

Table 1. Related works

Author	Modular approach	Structure modification	Time complexity	Description
U. Kjaerulff (1994) [7]	X	O	O	Removing weak dependenciesbefore inference
M. Marengononi et al.(2003) [5]	O	O	X	Hierarchicallystructured BN
B. Brandherm et al. (2005) [8]	X	X	O	Nodes with the value which are lowerthe threshold are not considered in DBN
A. Krause et al. (2006) [4]	O	X	X	Separated classifiermodules to infera user's activity
H. Tu et al. (2006) [6]	O	X	O	Hybridization of BN and HMM
Y. Xiang et al. (2007) [9]	O	O	X	Shafer-Shenoyalgorithm based lazypropagation method

This paper proposes a modular Bayesian network (MBN) system for facilitating context-aware decision making. Improved virtual linking method is devised to represent and preserve inter-modular dependencies regardless of a common node's properties. Selective inference algorithm is also presented as a way to reduce time complexity of MBN by limiting the range of modules to be evaluated. Proposed MBN system provides not only reliable decision making based on context-aware information but also easier way of representing and managing probabilistic knowledge. It also enables the integration of information on ubiquitous devices which have relatively limited computing power and available resources.

The organization of this paper is as follows: Section 2 introduces several definitions and proposed methods including inter-modular d-separation, virtual linking, and selective inference. Section 3 evaluates the accuracy and time of

inference in MBN compared to monolithic BN system to analyze whether MBN system fully reflects causalities and lessens the computation required. Section 4 concludes the paper with an overview and future work.

2 Modular Bayesian Network

Bayesian network is a popular tool widely used for statistical knowledge representation which machines can easily understand. Modular Bayesian networks are an extended version ofBayesian network. MBN has the basic BN's features such as d-separation[10] and increasing complexityas the growing number of parent nodes. Besides, it is possible to apply BN's inference or learningalgorithms to MBN without modification.In MBN, however, it is more difficult to keep dependencies between variables, especially inter-modularcausality due to its modular property. In addition, a cleverer method is required in inference sinceupdating the whole MBN given small change of observation would take considerable resources and time.To work out these problems, we propose a virtual linking method and selective inference.

2.1 Definitions

MBN consists of multiple BN units (BN modules) which are connected withother units according to their causality relationships. BN, BN modules, and MBN are defined as follows.

Definition 1 **(Bayesian network):** Bayesian networkis a probabilistic graphical model that represents casual relationships between random variables. BN consists of variables V, edges $E=(V_i, V_j)$, and conditional probability table $P(V)$. When evidence $e$is given the posterior probability $P(V|e)$ is calculated by applying the chain rule as (1):

$$P(V \mid e) = \prod P(V \mid Pa(V)) \cdot e = \prod P(V \mid Pa(V)) \prod_{e_i \in e} e_i \tag{1}$$

where$Pa(V)$ indicates the set of parent nodes and e means the evidence.

Definition 2 **(BN module):** A BN module $\Psi_i= (G_i, P_i)$ is a Bayesian network represented by$G_i= (V_i, E_i)$where V_i's are variables and $E_i= (V_i, V_j)$are directed edges from $V_i$to V_j, and P_i is the conditional probability. A BN module is a basic unit of a problem domain for perceiving contextual information.

Definition 3 **(Modular Bayesian networks):** MBN Ωconsists of a 2-tuple (M, R) where Mrepresents BN modules and Rindicatesthe causality between BN modules. Let two BN modules be defined as $\Psi_i= ((V_i, E_i), P_i)$ and $\Psi_j= ((V_j, E_j), P_j)$, and have an influence on each other. Then, a link $R= \{<\Psi_i,\Psi_i> \mid i\neq j, V_i\cap V_j\neq \emptyset\}$is created and able to affect or be affected by defining asharing node.

2.2 Inter-modular d-separation

A BN module can be a cause or result of the connected module by forming a hierarchical structure. Otherwise, two modules can affect sharing nodes with causative edges or share nodes as an input node with incoming edges to the modules. These relations between BN modules are defined by extending the concept of d-separation which has three types: serial, converging,and diverging connectionrepresented in Fig. 1.

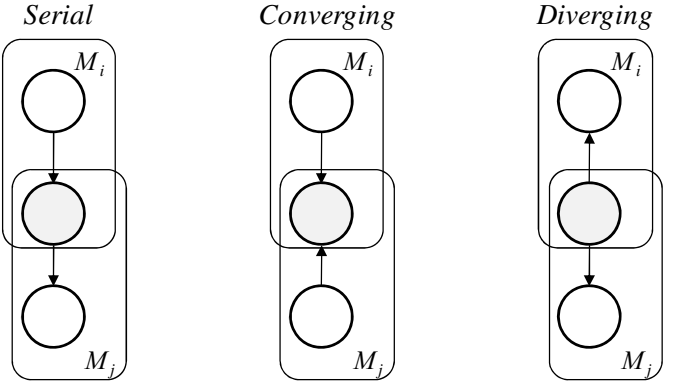


Fig. 1. Three connection types between BN modules

These connection types lead different aspects of information flow between modules according to the status of sharing nodes. When hard evidence is set on the sharing node S, two modules $\Psi_i$and $\Psi_j$are d-separated defined as $\Psi_i \perp_m \Psi_j | S$. d-separation between modules $\perp_m$means any changes in $\Psi_i$or $\Psi_j$cannot affect connected module. On the contrary, modules with converging edges have d-separationrelation when Shas no hard evidence, $\Psi_i \perp_m \Psi_j$. These relations are summarized as (2).

$$\begin{aligned} &\varphi_i \perp_m \varphi_j \mid S \quad (\textit{serial or diverging}) \\ &\varphi_i \perp_m \varphi_j \qquad (\textit{converging}) \end{aligned} \tag{2}$$

The concept of d-separation between BN modules is required to cover the case in which there is no direct connection between modules. This is defined as modular d-separation.

Definition 4 **(Modular d-separation):** Two modules, $\Psi_i$and Ψ_j, have modular d-separation relation when there is no connected path $\pi = <\Psi_i, ..., \Psi_j>$ given the evidence set. When there is more than one path, it is called as modular d-connection.

3 Inference with Modular Bayesian Networks

Evaluating MBN is the most important process. In order toperformthe whole inference process, it requires multiple steps controlled by components.First, MBN system picks the target nodes which are mostly needed to probe based on the context information collected so far. If a module contains a target node, it becomes target module. When target modules arefixed, each module is prioritized to decide precedence of evaluating.This is a crux of MBN system since a minor change in precedence between modules would cause huge difference in posterior probability even when the same evidence set is given.Then newly observed evidence is distributed to BN modulesso that evaluation is conducted.This process is summarized in Fig. 2.

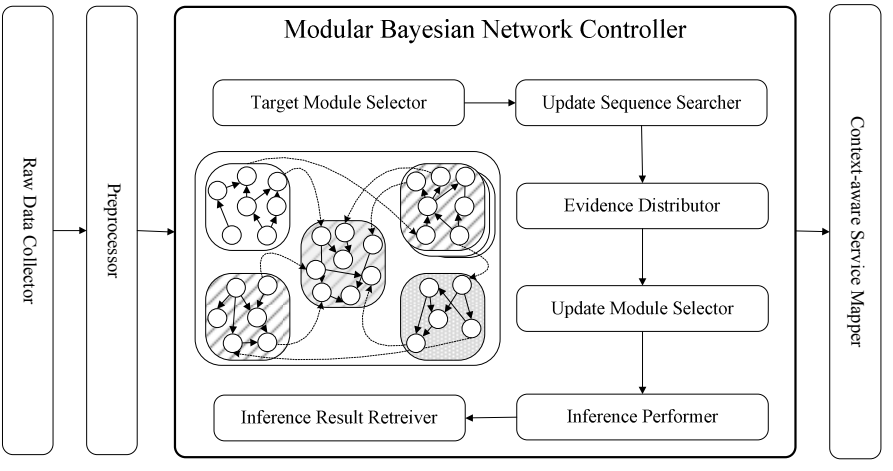


Fig. 2. Modular Bayesian network system components needed in inference

3.1 Preserving Causalities

It is a crucial problem to capture the correct causalities between variables in general Bayesian networks. The same property is also required in modular Bayesian network for reliable result. We devise a virtual linking method as a way of maintaining causality whichfacilitates communication between BN modules.

Definition 5 **(Virtual linking method):** Virtual linking method is a technique which enables each BN modules to convey its internal inference results to the other connected modules. BN modules are virtually linked when module $\Psi_i$and $\Psi_j$have at least a sharing node, S. The inference results of Ψ_i,$Bel(S_i)=\{p_1, p_2,..., p_n\}$, is propagated to $\Psi_j$by coercing the initial probability of a node $S_j$of $\Psi_j$in accordance with $Bel(S_i)$. The initial probability of a shared node in a moduleis controlled by utilizing a virtual node.

***Definition 6*(Virtual node):** A virtual node is an auxiliary node which is added on a network[7]. It sets virtual evidence which has uncertainty associated with it. Virtual node *V* is linked to a target node *T*as a child node; causality from *T* to *S*. Probability of *V*is set according to the target probability of *T*.

Local inference result of BN module can be passed to all other modules by utilizing a virtual node. So, it is a crux of virtual linking method to calculate the probability of virtual node given target probabilities for a shared node. This calculation process can be dealt with two cases: a shared node has two states, and more than two states.

When a sharing node *S*between two modules has only two states, the probability of a virtual node, which has only two states "*yes*" and "*no*", and hard evidence on "*yes*", $P(V=yes|s_i)=\{v_a, v_b\}$ is computed as

$$\frac{V_a}{V_b} = \frac{P(V = yes \mid S = s_a)}{P(V = yes \mid S = s_b)} \frac{P(S = s_a)}{P(S = s_b)} \tag{3}$$

where$P(V=yes|S=s_a)$ and $P(V=yes|S=s_b)$ are the target probabilities of *S*, and $P(S=s_a)$ and $P(S=s_b)$ are the initial probabilities of *S*. The result of (3) is a just ratio rather than probability value, so it needs to bemodulated by multiplying modulation coefficient αuntil both has the value ranging from 0 to 1.

When a shared node has more than two states, probabilities of a virtual node are computed differently. The calculation should be extended in accordance with thenumber of the states of a shared node. Let $PI=\{i_0, i_1, ..., i_n\}$ be initial probabilities of a shared node, $PT=\{t_0, t_1, ..., t_n\}$ be a target probability distribution of a shared node, and $PV=\{v_0, v_1, ..., v_n\}$be a probability distribution of a virtual node where *n*is the number of states of a shared node. Then we can get a proportional expression like (4)based on Bayes' theorem.

$$t_0 : t_1 : \ldots : t_n = i_0 \cdot v_0 : i_1 \cdot v_1 : \ldots : i_n \cdot v_n \tag{4}$$

We can get *the ratio of PV*for each states by solving (5).

$$v_i = \frac{t_i}{i_i}, \qquad PV_i = \frac{PT_i}{PI_i} \tag{5}$$

Actual probability distribution of a virtual node can be calculated by multiplying modulation coefficient which makes it appropriate probability value.

Every shared node in BN modules has corresponding virtual nodes, and through this channel BN modules can pass their messages to the neighbor modules and influence overall network. Virtual linking method enables the creation of networks over Bayesian network modules.

3.2 Selective Inference for Modular Bayesian Networks

In the updating process of BN modules in MBN, there are two factors we need to consider. The primary factor which has great impact on the accuracy of inference is the sequence of modules to evaluate. If the target module is updated in the early stage it may fail to reflect posterior belief fromconnected modules which causes inaccurate

results. For reliability of MBN system, a prioritizing algorithm is required which defers the target module's update until all the related modules are updated. Another is defining and controlling the range ofmodules to infer. When new evidences are observed in several BN modules, there is no need to evaluate the whole BN modules since not all of them are propagated to the other modules.

In order to guarantee reliable results with timely effective inference of MBN on ubiquitous devices such as mobile phones or embedded sensors, we update only relevant modules in the order of precedence. The precedence of BN modules is determined with a simple algorithm. Target modules have the top priority and are updated in the end. The second priority is given to the modules that have dependencies with the target modules, and modules which connect to the modules with second precedence get the next priority. When there are more than two candidate BN modules, a BN module with more sharing nodes precede others.Detailed algorithm is specified in Fig. 3.

```
So;       // Sois a stack which contains inference order
// Module with higher precedence is pushed earlier.
Qs;       // Qs is a queue which contains candidate modules to be searched
Ltemp; // Ltempis a list which contains multiple module IDs.

Ltemp= SortbyNumberofVirtualNode(Ψtarget);
push(Ltemp);      Enqueue(Ltemp);

while(So.Count() != MBN.CountModules()){
        if(Qs!= null){
                Ltemp= ConnectedModules(Dequeue());
                Ltemp= SortbyNumberofVirtualNode(Ltemp);
        }else
                Ltemp= SortbyNumberofVirtualNode(Ψremainders);
        push(Ltemp);     Enqueue(Ltemp);
  }
```

Fig. 3. Pseudo-code of the selective inference algorithm

4 Experimental Results

Experiments are comprised of two parts. The first part contains measurement of the MBN's accuracy with sequential order of updating described in the previous section. Next, we show that our selective inference method reduces the time complexity of MBN compared to monolithic BN.

At first, an experiment is designed to check the reliability of the proposed MBN. We measure the MBN's accuracy by comparing the posterior belief of a target node in MBN with that in monolithic BNwhen the same evidence set is given. Three well-known BN's, Alarm[11], Credit, and Hailfinder[12]are chosen as the monolithic BN, and we designed the MBN's with 6, 4, and 7 modules, respectively. Some engineering work would be needed to optimize the number of modules, but we did not put much effort to do it.

Table 2. Differences in posterior probabilities of target nodes between monolithic BN and MBN with different algorithms

OrderingMethod	Alarm	Credit	Hailfinder	Overall
1) Random order	0.92%	0.87%	1.95%	1.25%
2) Increasing order based on number of sharing nodes	0.73%	1.03%	3.32%	1.69%
3) Decreasing order based on number of sharing nodes	1.20%	0.74%	0.62%	0.85%
4) Increasing order based onnumber of nodes	0.72%	1.00%	3.32%	1.68%
5) Based on "3", and "5" as asecondary rule	0.67%	1.04%	3.32%	1.68%
6) Proposed method	0.05%	0.41%	0.07%	0.18%

Table 2 shows the average discrepancies between monolithic BN and MBN for three different models for 1000 inferences. The proposed method outperforms other methods and its average differences are close to 0%. This indicates that the proposed ordering algorithm successfully enables each BN modules to pass and get the information.

The second experiment is about the time complexity of MBN. We measured the time of the inference of monolithic BN and MBN on the smart phone whose resources and computing power are relatively limited. We create a BN model which is for extracting context information of a user in schedule management agent. For convenient management and scalability, it is modeled with 7 BN modules of MBN which represented in Fig. 4.

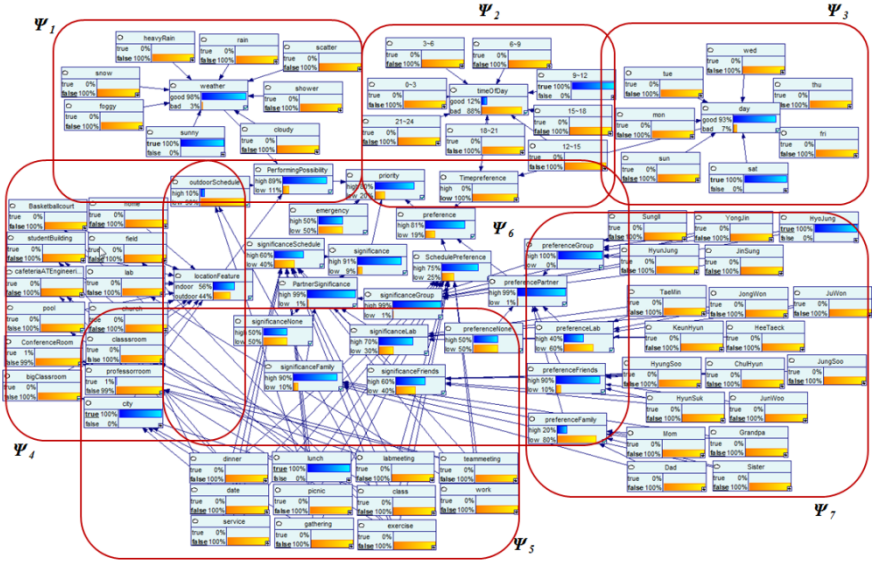


Fig. 4. A simple MBN model designed for personal schedule management

Fig. 5. The time taken to update MBN and monolithic BN

Average times of monolithic BN and MBN for inference are shown in Fig. 5. At the first inference,MBN takes slightly more time than monolithic BN because it loads more nodes and edges that are in separated multiple files. However, MBN saves time as inference is conducted repeatedly and it reduces significantly at last. This experimental result demonstrates the selective inference algorithm makes the inference of MBN more efficient and time-saving.

5 Conclusion

This paper presented modular Bayesian networks as a system. In order to preserve inter-modular dependencies we devised virtual linking method. For effective updating of MBN, each module is selectively inferred and passes the result to the neighborhood modules in the calculated order. With this MBN system, it becomes easier to design and manage probabilistic knowledge representation. Furthermore, it guarantees the extensibility so that context-aware system designer can adopt requisite partial BNs which are learned or designed previously in different domains, and put them together for its own purpose like mash-up. For the future research, inter-modular cycle problem should be handled in inference process because it would produce unintended posterior probability in MBN. The new method should not be very complex, nor increase inference efficiency.

Acknowledgement. This work was supported by the IT R&D program of MKE/KEIT (10033807, Development of context awareness based on self-learning for multiple sensors cooperation).

References

1. Hwang, K.-S., Cho, S.-B.: Landmark detection from mobile life log using a modular Bayesian network model. Expert Systems with Applications 36, 12065–12076 (2009)
2. Pavlin, G., et al.: A multi agent systems approach to distributed Bayesian information fusion. Information Fusion 11(3), 267–282 (2009)
3. Li, X., Ji, Q.: Active affective state detection and user assistance with dynamic Bayesian networks. IEEE Transactions on Systems, Man, and Cybernetics, Part A 35(6), 93–105 (2005)
4. Krause, A., Smailagic, A., Siewiorek, P.D.: Context-aware mobile computing: Learning context-dependent personal preferences from a wearable sensor array. IEEE Transactions on Mobile Computing 5(2), 113–127 (2006)
5. Marengoni, M., Hanson, A., Zilberstein, S., Riseman, E.: Decision making and uncertainty management in a 3D reconstruction system. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(7), 852–858 (2003)
6. Tu, H., Allanach, J., Singh, S., Pattipati, R.K., Willett, P.: Information integration via hierarchical and hybrid Bayesian networks. IEEE Transactions On Systems, Man, and Cybernetics. Part A: Systems and Humans 36(1), 19–33 (2006)
7. Kjaerulff, U.: Reduction of computational complexity in Bayesian networks through removal of weak dependences. In: Proceedings on Uncertainty in Artificial Intelligence, pp. 374–382 (1994)
8. Brandherm, B., Schwartz, T.: Geo Referenced Dynamic Bayesian Networks for User Positioning on Mobile Systems. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, pp. 223–234. Springer, Heidelberg (2005)
9. Xiang, Y., Jensen, V.F.: Lazy inference in multiply sectioned Bayesian networks using linked junction forests. Studies in Fuzziness and Soft Computing 213, 175–192 (2007)
10. Korb, K.: Bayesian Artificial Intelligence, pp. 29–68. Chapman & Hall/CRC (2004)
11. Beinlich, I., Suermondt, H., Chavez, M., Cooper, G.: The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In: Proceedings of the Second European Conference on Artificial Intelligence in Medicine, vol. 38, pp. 247–256 (1989)
12. Abramson, B., Brown, J., Edwards, W., Murphy, A., Winkler, R.: Hailfinder: A Bayesian system for forecasting severe weather. International Journal of Forecasting 12, 57–72 (1996)

Real-World Problem for Checking the Sensitiveness of Evolutionary Algorithms to the Choice of the Random Number Generator

Miguel Cárdenas-Montes[1], Miguel A. Vega-Rodríguez[2], and Antonio Gómez-Iglesias[3]

[1] Centro de Investigaciones Energéticas Medioambientales y Tecnológicas, Department of Fundamental Research, Madrid, Spain
miguel.cardenas@ciemat.es,
[2] University of Extremadura, ARCO Research Group, Dept. Technologies of Computers and Communications, Cáceres, Spain
mavega@unex.es,
[3] Centro de Investigaciones Energéticas Medioambientales y Tecnológicas, National Laboratory of Fusion, Madrid, Spain
antonio.gomez@ciemat.es

Abstract. This article presents an analysis of the sensitiveness of evolutionary algorithms to the change of the random number generator when using a real-world problem —the fitting of a theoretical curve to an experimental data set— as test. On the one hand, the evolutionary algorithms selected: particle swarm algorithm, differential evolution and genetic algorithm are widely used in optimization problems. And, on the other hand, the random number generator used: Mersenne Twister and GCC rand(), are the most frequently linked to evolutionary algorithms, as well as they are considered as high-quality. As a consequence of this work, an assessment is stated about the sensitiveness of the evolutionary algorithms studied to the choice of the random number generator.

Keywords: Random Number Generator, Particle Swarm Algorithm, Differential Evolution, Genetic Algorithm.

1 Introduction

There are numerous scientific and technical disciplines which use random number sequences in their simulations. These disciplines are concerned about the randomness of the Random Number Generator (RNGs) employed. Fortunately, RNGs have become so close to real random number sequences that certain computational experiments are unable to distinguish between real and computational-generated random number sequences [1].

Evolutionary algorithms (EAs) techniques rely heavily on the use of RNG. From initial population generation, through the specific canonical operators applied to create new temporary population, the use of randomness is pervasive through EAs. Therefore, it is reasonable to wonder how RNG quality affects EAs performance.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 385–396, 2012.

As soon as newer and longer period RNGs appear, articles studing the effect of its choice in the final performance of optimization problems are published. In spite of the continuous update of state-of-the-art, the majority of these publications use artificial problems.

These artificial problems are simplifications of more complex real problems. Therefore, the conclusions drawn in these studies should be put in quarantine and they should not be extrapolated to real problems.

In this study, a real problem —the fitting of experimental data sets to a theoretical curve— is used in order to check the sensitiveness of several EAs to the choice of the RNG. The experimental data set employed are the orbital velocity of stars for four spiral galaxies: NGC2460, NGC3370, NGC4800 and NGC5394.

The experimental data set provide a variety of scenarios: measures with big experimental errors and others with small ones, galaxies with a lot of experimental points and other with very few ones, and, galaxies where the two arms fit the same curve and others where strong differences between the two arms have been observed.

For the theoretical curve, diverse series expansions were tested. Finally, a Legendre-polynomial serial expansion was selected due to its better adjustment to the experimental data sets.

In this paper two RNGs (Mersenne Twister and GCC rand()) have been used to test their impact over the final performance of three EAs: Particle Swarm Algorithm (PSO), Differential Evolution (DE) and Genetic Algorithm (GA). The two RNG have been selected based on two criteria:

- The RNG has to be frequently used in research papers.
- The RNG has to be considered as high quality RNG.

RNGs —and the suites for testing their properties— have hardly evolved in the last years. The most stringent suites for checking the randomness allow separating good RNGs from others. However, the validation of these tests does not suffice to deduce that all the RNGs will produce similar performances when coupling to EAs.

Beyond of the initial scope of this work as it has been exposed, the conclusions drawn could be extrapolated toward other areas where the RNGs and the EA have a key role in the algorithm. Techniques where the EA are hybridized with other techniques, such as hybrid intelligent algorithms, will be concerned by the potential sensitiveness of the EAs. Therefore, some of the hybrid algorithm presented in previous editions of the HAIS congress and related publications: [2], [3] [4] could exhibit similar sensitiveness to the change of the RNG. As a consequence, the results could show small variations when employing different RNGs.

This paper is organized as follows: Section 2 summarizes the related work and previous efforts done. In Section 3.1, the Evolutionary Algorithms tested in this article are briefly described. In Section 3.2, a resume of the Random Number Generators used in the survey is introduced. Shortly, Section 3.3 describes the statistics used in the analysis process. In Section 4, the implementation details and the production set-up are shown. The results are displayed and analysed in Section 5. And finally, the conclusions and the future work are presented in Section 6.

2 Related Work

The first works in the assessment of the impact of the RNG over the final performance of EAs are characterised by studies restricted to GA as reference of the EA, the inclusion of well-recognized as poor-quality RNGs, and the use of a coarse-grained statistical for the analysis [5], [6]. All these factors lead, today, to put in quarantine the conclusions attained; and make relevant an update of this type of study. Later, a study with finer statistics [7] did not find correlation between goodness on the RNG tests —Diehard suite— and good performance obtained by the GA.

Other paper [1] has studied the sensitiveness of GA to the choice of RNG focusing on the components that are most affected by the RNG. The work presents an ablation experiment using two RNG and the true random number from an atmospheric noise source. The experiments showed that the RNG used to initialize the population had a critical impact over the final performance; whereas the RNG used as input to other operations —crossover and mutation— did not affect the performance significantly.

A special mention requires the work [8] where 11 artificial functions —extracted or inspired from *CEC 2010 and 2008 Special Sessions and Competition on Large-Scale Global Optimization (CEC competitions)* [9]— were used to build a scale of sensitiveness of EAs face to the change of RNG. In this work the EA evaluated were: PSO, DE, GA and firefly algorithm (FA); and the same RNGs: Mersenne Twister and GCC rand().

Other more recent work [10] presents a study of the effect of RNG on the performance of Differential Evolution. This work uses a set of benchmark functions similar to those used in [8]; however the number of tries and the statistical analysis of results is shorter and simpler —i.e. avoiding the employment of non-parametric statistical inference—. Other differences with this work rely on the analysis only over DE; whereas in [8] other three EAs are analysed: GA, PSO and FA. Finally, [10] includes some well-recognized bad and obsolete RNG, which are not considered here for this reason.

3 Methods and Materials

3.1 Evolutionary Algorithms

The sensitiveness of three EAs have been tested in this work: PSO [11], [12]; DE [13], [14] and GA [15], [16]. These EAs have been selected based on their wide applicability in optimization problems; these algorithms being very representative in the community. In-depth description of the EAs tested can be found in the mentioned articles, this being this purpose out of the scope of the present paper.

In all EAs used in this work, the population structure is panmictic. Thus, the intrinsic operations to each EA —mutation, reproduction, selection, replacement, etc.— take place globally over the whole population. Furthermore, in all cases the EAs follow a generational model, in which a whole new population of individuals replaces the old one [17].

3.2 Random Number Generators

It may seem to be a conceptual impossibility to produce "random" numbers with computers which are completely deterministic machines. Nevertheless, Random Number Generators (RNGs) are in common use, especially in science and engineering disciplines.

Sometimes the term pseudorandom is used for computer-generated sequences, while the word random is reserved for intrinsically random physical process. Such fine distinctions are not made in this paper.

In general, the random number sequences produced by RNG ought to be uniform, uncorrelated and of extremely long period. It is clear that for optimum performance and accuracy of EA, the RNG used ought to have these good properties. Some of the most common tests to examine the quality of RNGs are:

Uniformity test: Break up the interval between zero and one into a large number of small bins and after generating a large number of random numbers, check for uniformity in the number of entries in each bin.

Overlapping M-tuple test: Check the statistical properties of the number of times that M-tuples of digits appear in the sequence of random numbers.

Parking lot test: Plot points in an m-dimensional space where the m-coordinates of each point are determined by m-successive calls to the RNG. Then look for regular structures.

The two RNG tested in this paper are Mersenne Twister and GCC RAND. Both RNGs fit the criteria previously exposed: high-quality and widely used in optimization problems with EAs.

Mersenne Twister RNG. The Mersenne Twister RNG was developed in 1997 by Matsumoto and Nishimura [18]. It is a high-quality RNG designed specifically to rectify many of the flaws found in older RNGs. Its name derives from the fact that period length is chosen to be a Mersenne prime number.

The main features of the Mersenne Twister RNG are:

- It has a very long period of $2^{19937} - 1 \simeq 10^{6000}$. While a long period is not a guarantee of quality in a random number generator, short periods —common in many software packages— are problematic.
- It is k-distributed to 32-bit accuracy for every $1 \leq k \leq 623$ (Overlapping M-tuple test).
- It passes numerous tests for statistical randomness, including the *Diehard* tests. It passes most, but not all, of the even more stringent *TestU01 Crush* randomness tests.

GCC RAND RNG. In C and C++, the default RNG is the rand() function. Many platforms have poor-quality versions of the rand() function, however GNU platforms —*glibc*— implement a version with higher quality and broadly accepted as good quality RNG.

The main features of the GCC RNG are:

- The implementation of *glibc* corresponds to the category of Linear Congruential Generator [19].

– It has a period of $\simeq 2^{31} - 1$. This period is accepted in general as long but it is clearly shorter than the Mersenne Twister RNG.

Spite of the difference of period between both RNGs, the GNU rand() is accepted as good quality RNG, it being used in many scientific and technical works. The question is if these differences affect to the final results of the optimization process based on EAs. The intent of this work is to unveil this question.

In our implementations, the seed of the RNG is reinitialised for each new execution in order to keep the study as fair as possible.

3.3 Statistical Inference

Statistical hypothesis testing is a fundamental method used at the data analysis stage of a comparative experiment. For this comparison, two kind of tests can be used: parametric and non-parametric. The main difference between parametric and non-parametric tests rely on the assumption of a distribution underlying the sample data. Given that non-parametric tests do not require explicit conditions on the underlying sample data, they are recommended when the statistical model of data is unknown [20].

The Wilcoxon signed-rank test belongs to the category of non-parametric test. It is a pairwise test that aims to detect significant differences between to sample means [20], [21], that is in our study, the sensitiveness of EA face to the change of the RNG.

In order to assess if the EA algorithm performs better when implementing one of the RNG the sign test can be used. The sign test is also a non-parametric test. For this test the differences between the results obtained with both RNG are calculated. Next, by counting the number of plus signs or minus signs, it can be stated if the EA performs better with a particular RNG [22]. For this scenario, the number of plus signs or minus signs must be lower than or equal to a critical value depending of the sample size.

4 Production Setup

In order to check the sensitiveness of EAs to the choice of the RNG, the adjustment of experimental data to a polynomial series has been used. The experimental data correspond to the orbital velocity of stars around the centre of the galaxy.

As polynomial series, diverse series expansions were initially tested: Legendre, Bessel and normal series expansion. Legendre polynomial obtained the best results reproducing better the experimental data sets. For the series expansion two degree of expansion —10 and 20— were employed. This parameter marks the dimensionality of the problem. The rest of the configuration used in this work is: 10 particles/individuals as population size and 1,000 cycles/generations.

In the PSO implementation, the c_1 and c_2 constants were established as $c_1 = c_2 = 1$ and the maximum velocity of particles $V_{max} = 2$. In the DE implementation, the mutation rate was established as $\mu = 0.5$ and the recombination rate as $C_r = 0.5$. In the GA implementation, the mutation rate was established as $p_m = 0.01$, without elitism.

According to the usual practice in adjustment of experimental data to theoretical curve, the chi-square test —χ^2—- has been chosen as fitness function within this work

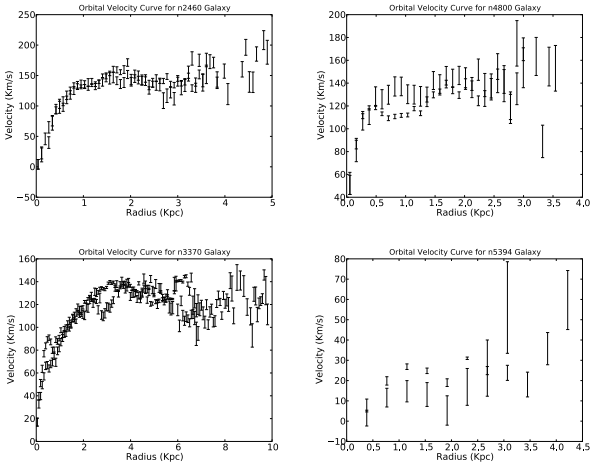


Fig. 1. Galaxy rotation curves —experimental data sets— used in this survey: NGC2460, NGC4800, NGC3370 and NGC5394

[23]. The lower the χ^2 value is, the closer the solution is to the objective —the fitter the experimental data is to the theoretical curve. Therefore, the task becomes minimizing χ^2.

Considering the standard fitting problem, where one is given a discrete set of N data points with associated measured errors σ, and is asked to construct the best possible fit to these data using a specific functional form, the most appropriated fitness function is the merit function χ^2, Eq. 1 [19]. Therefore, independently of the specific functional form chosen, the fitness function used in this work is χ^2, Eq. 1.

$$\chi^2 = \sum_{all\ points} \left(\frac{y_{simulated} - y_{observed}}{\sigma}\right)^2 \tag{1}$$

For each case —each EA, galaxy rotation curve and polynomial degree— a total of 25 tries were executed in order to reach the statistical relevance desired.

In Fig. 1, the four galaxy rotation curves used in this work are shown. Particularly, these galaxy rotation curves were extracted from a larger astronomical data, covering approximately 60 galaxies. The criteria to select these curves were: the largest and the smallest data set, and two more randomly selected.

As can be appreciated the experimental data are very diverse, providing different scenarios: curve with many points —more than 200— and with few points —less than 20—, big and small errors, duplicated values for the same x-axis coordinate —generally corresponding to the two arms of the spiral galaxy—. This diversity makes the task of adjustment very challenging and stressing; and providing different scenarios to test the EAs coupled to the RNGs.

In order to fit the experimental points —see graphics in Fig. 1— to a theoretical curve, a series expansion of Legendre Polynomial was used (Eq. 2). In this series expansion the unknown terms a_i have to be calculated to obtain the better adjust of function $F(x)$ to the experimental points.

$$F(x) = \sum_{i=0}^{N} a_i \cdot LP_i(x) \tag{2}$$

5 Analysis and Results

The most appropriated statistical test to assess the impact of the choice of the RNG over final performance of the EA is the Wilcoxon signed-rank test. This is a non-parametric test used for statistical inference. In Table 1, the p-value of Wilcoxon signed-rank test for each EA, galaxy and polynomial degree is presented. In our study, the analysis of the sensitiveness of the EAs is based on these values. The significance level used has been $\alpha = 0.05$, which is the most usual in this kind of analysis.

When changing the RNG in the PSO algorithm, *null hypothesis* —$H_0 : \mu_1 = \mu_2$— can be rejected only in one case —NGC2460 and degree 10—. This is the single case where the PSO algorithm shows sensitiveness to the change of the RNG. By contrast, for the rest of the cases —total of 7— the null hypothesis can not be rejected. For these cases the change of RNG has not any impact over the final performance of PSO.

Regarding the results of Wilcoxon signed-rank test for DE, the *null hypothesis* can be rejected also in a single case —NGC2460 and degree 20—. For the rest of the cases —total of 7— the *null hypothesis* can not be rejected.

Finally for GA, the *null hypothesis* can not be rejected in any case. Therefore, the statement $H_0 : \mu_1 = \mu_2$ has to be accepted as true; leading to the conclusion that the change of the RNG does not have any impact over the final performance of the GA.

Based on the non-parametric analysis performed, it can be concluded that the choice of the RNG has not any impact —for GA— or a very small impact —for PSO and DE— over the final performance of the problems treated in this work.

Unfortunately, the Wilcoxon signed-rank test does not allow to establish conclusions about which particular RNG produces best performance when it is coupled to a EA, only if the results differ or not.

The analysis of the results allows building a scale of sensitiveness for the EAs tested: $DE = PSO > GA$. This scale coincides partially with the scale created with the

Table 1. p-value from Wilcoxon signed-rank test for non-parametric hypothesis testing for each evolutionary algorithm, galaxy and expansion degree

Evolutionary Algorithm	Galaxy and Polynomial Degree							
	NGC2460		NGC3370		NGC4800		NGC5394	
	10	20	10	20	10	20	10	20
PSO	**0.026**	0.367	0.288	0.925	0.757	0.657	0.158	0.276
DE	0.443	**0.040**	0.098	0.638	0.619	0.135	0.058	0.946
GA	0.065	0.619	0.638	0.861	0.545	0.638	0.946	0.109

Table 2. Comparison of the best adjustment obtained with the RNG tested for galaxy NGC2460

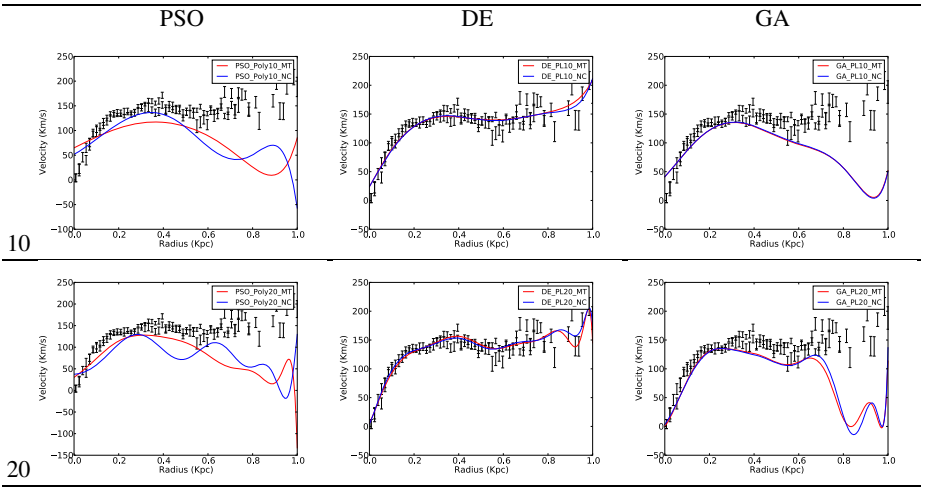
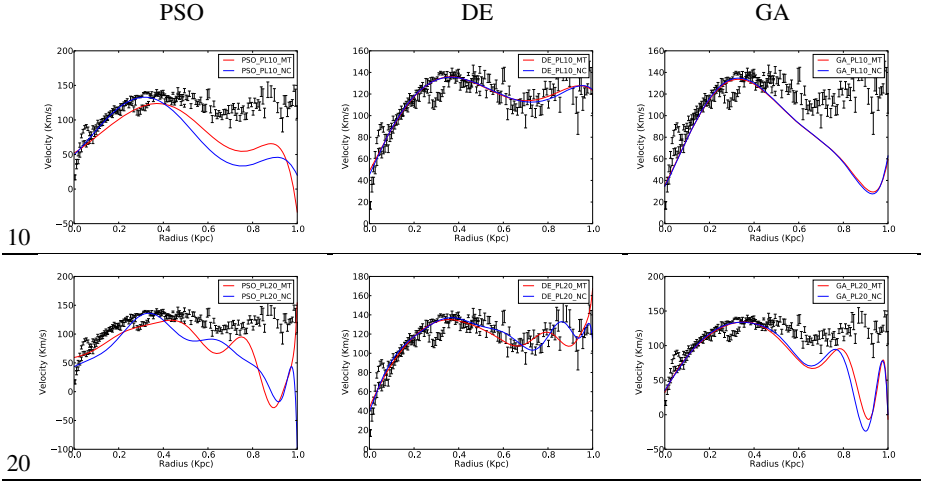


Table 3. Comparison of the best adjustment obtained with the RNG tested for galaxy NGC3370



work [8], in which only artificial functions were employed. These functions —a total of 11, including multimodal and monomodal, separable and non-separable— were extracted or inspired from *CEC 2010 and 2008 Special Sessions and Competition on Large-Scale Global Optimization (CEC competitions)* [9]. In order to test the sensitiveness a large production was performed using four EAs: PSO, DE, GA and FA, and the same RNG: Mersenne Twister and GCC rand(); as well as an in-depth analysis —also based on the Wilcoxon signed-rank test— of the data obtained. In this work the scale of sensitiveness obtained was: $DE > GA > PSO > FA$.

Table 4. Comparison of the best adjustment obtained with the RNG tested for galaxy NGC4800

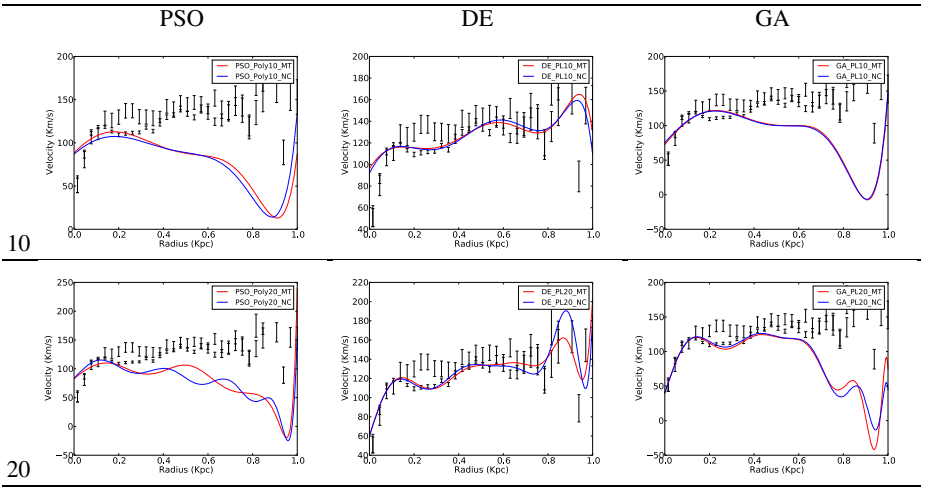
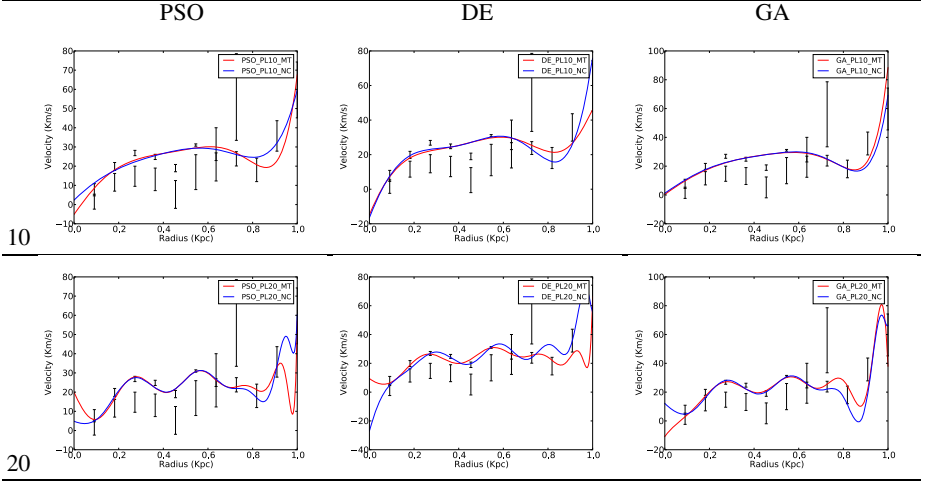


Table 5. Comparison of the best adjustment obtained with the RNG tested for galaxy NGC5394



As can be appreciated, in both studies DE is the most sensitive algorithm to the choice of the RNG. Oppositely, both surveys alternate the positions in the scale of GA and PSO.

In order to evaluate if the EAs perform better when using a particular RNG, the sign test has been employed. For testing $H_0 : M_{NC} \geq M_{MT}$ against $H_1 : M_{NC} < M_{MT}$, reject H_0 if the number of plus signs is less than or equal to the critical value for this test. Taking into account that the number of tests is 25, the critical value is 7. When applying this to the experimental data, only one rejection is produced —PSO, NGC2460 and

degree 10—. Therefore, only for this case, the algorithm performs better when using MT rather than NC can be stated. For the rest of cases, the null hypothesis can not be rejected, therefore the statement that both RNGs perform similarly can not be rejected.

Alternatively, the opposite null hypothesis can also be tested, $H_0 : M_{MT} \geq M_{NC}$ against $H_1 : M_{MT} < M_{NC}$. If the number of minus signs is less than or equal to the critical value, then the null hypothesis can be rejected. In this case, the null hypothesis can be rejected in two test, both for DE: NGC2460 and degree 20; and NGC3370 and degree 10. Therefore, for these two cases, the DE algorithm performs worse when implementing MT. For the rest of case the null hypothesis can not be rejected.

Considering the cases where the null hypothesis can be rejected by both tests: Wilcoxon signed-rank test and sign test, it exists coincidence in two cases —the two cases where sensitiveness to the change of RNG is detected by Wilcoxon signed-rank test are also detected by sign test. In a third cases —DE, NGC3370 and degree 10—, the null hypothesis is only rejected by the sign test, whereas the sign test does not detect differences.

In Tables 2, 3, 4 and 5 the best adjustments obtained for each case are presented. As observed in these figures, in most of the cases both curves well-conform the experimental data, so the simple observation of them does not allow to discern if the RNGs differ in performance when coupling to different EAs. This endorses the results obtained by the Wilcoxon signed-rank test disabling to discern between both RNGs.

6 Conclusion and Future Work

This work analyses the sensitiveness of diverse EAs to the choice of RNG using real-world problems: the fitting of experimental data sets to a theoretical curve. Concerning the EAs, three widely employed EA have been used: PSO, DE and GA. In relation to the RNG, two high quality generators and frequently used in scientific papers —Mersenne Twister and GCC rand()— have been considered.

Unlike other previous studies where artificial problems are usually employed, in this one a real-world problem —the fitting of experimental data sets to a theoretical curve— has been considered as criterion to check the sensitiveness of the EAs.

From the results obtained, it can be observed that the variation of results for PSO and DE are scarce, and therefore, these algorithms show a low sensitiveness to the choice of the RNG. Regarding GA, variations in the results can not be detected, and as a consequence it shows a null sensitiveness to the change of RNG. These results roughly coincide with similar studies which used artificial functions.

More comparative works using other real-world problems, finer-grained statistics and particularly examples of hybrid algorithms are proposed as future work.

Acknowledgement. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) via the project EGI-InSPIRE under the grant agreement number RI-261323. The author thanks the Spanish Network for e-Science (CAC-2007-52) for their support when using the NGI resources.

The authors acknowledge to the reviewers and to the publication coordinators the comments and improvements suggested. This has allowed a more clear explanation of the results obtained as well as the conclusions stated, and therefore producing a net improvement in the work's comprehension.

References

1. Cantú-Paz, E.: On random numbers and the performance of genetic algorithms. In: GECCO, pp. 311–318. Morgan Kaufmann (2002)
2. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
3. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. Inf. Sci. 180(14), 2633–2634 (2010)
4. Corchado, E., Graña, M., Wozniak, M.: Editorial: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
5. Meysenburg, M.M., Foster, J., Saghi, G., Dickinson, J., Jacobsen, R.T., Shreeve, J.M.: The effect of pseudo-random number generator quality on the performance of a simple genetic algorithm. Master's thesis, University of Idaho, Idaho (1997)
6. Meysenburg, M.M., Foster, J.A.: The quality of pseudo-random number generations and simple genetic algorithm performance. In: Bäck, T. (ed.) ICGA, pp. 276–282. Morgan Kaufmann (1997)
7. Meysenburg, M.M., Foster, J.A.: Randomness and GA performance, revisited. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, vol. 1, pp. 425–432. Morgan Kaufmann, Orlando (1999)
8. Cárdenas-Montes, M., Vega-Rodríguez, M.A., Gómez-Iglesias, A.: Sensitiveness of Evolutionary Algorithms to the Random Number Generator. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part I. LNCS, vol. 6593, pp. 371–380. Springer, Heidelberg (2011)
9. Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T.: Benchmark functions for the cec'2010 special session and competition on large-scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China (USTC), Electric Building No. 2, Room 504, West Campus, Huangshan Road, Hefei 230027, Anhui, China (2009)
10. Tirronen, V., Äyrämö, S., Weber, M.: Study on the Effects of Pseudorandom Generation Quality on the Performance of Differential Evolution. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part I. LNCS, vol. 6593, pp. 361–370. Springer, Heidelberg (2011)
11. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948 (1995)
12. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm Intelligence (The Morgan Kaufmann Series in Artificial Intelligence), 1st edn. Morgan Kaufmann (April 2001)
13. Price, K.V., Storn, R., Lampinen, J.: Differential Evolution: A practical Approach to Global Optimization. Springer, Berlin (2005)
14. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization 11(4), 341–359 (1997)
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag New York, Inc. (1994)
16. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1998)
17. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE Trans. Evolutionary Computation 6(5), 443–462 (2002)

18. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Transactions on Modeling and Computer Simulation 8(1), 3–30 (1999)
19. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press (1992)
20. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. J. Heuristics 15(6), 617–644 (2009)
21. García, S., Fernández, A., Luengo, J., Herrera, F.: A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput. 13(10), 959–977 (2009)
22. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inf. Sci. 180(10), 2044–2064 (2010)
23. Montgomery, D., Runger, G.: Applied Statistics and Probability for Engineers. John Wiley and Sons Ltd., New York (2002)
24. Tang, K., Yao, X., Suganthan, P.N., MacNish, C., Chen, Y.P., Chen, C.M., Yang, Z.: Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China (2007)

Hybrid Multi-objective Machine Learning Classification in Liver Transplantation

M. Pérez-Ortiz*, M. Cruz-Ramírez, J.C. Fernández-Caballero, and C. Hervás-Martínez

Department of Computer Science and Numerical Analysis
University of Córdoba, Córdoba, Spain
{i82perom,mcruz,jcfernandez,chervas}@uco.es

Abstract. This paper constructs a hybrid, multi-objective and evolutionary algorithm based on Differential Evolutions using neural network models and q-Gaussian basis units in order to develop an efficient and complete system for donor-recipient assignment in liver transplantation. The algorithm is used for the classification of a binary dataset and will predict graft survival at 15 and 90 days after the transplantation. Other hybrid approaches combining artificial neural networks with evolutionary computation and well-known algorithms are presented in order to compare the obtained performance of both mono and multi-objective methods, using other methods such as Support Vector Machines and Discriminant Analysis. Some supervised attribute selection methods were previously applied, in order to extract the most discriminant variables in the problem presented. The models obtained allowed medical experts to predict survival rates and to come to a fair decision based on the principles of justice, efficiency and equity.

Keywords: multiobjective, evolutionary computation, neural networks, liver transplantation, differential evolution.

1 Introduction

During the last decades, new trends in biomedicine have considered some machine learning techniques as classification methods, where the predicted decisions are based on the recognition of some determinant patterns within the data. This kind of method has worked well in a great number of problems.

Liver transplantation is nowadays a widely-accepted treatment for patients who have terminal liver disease. Despite this, transplantation is greatly hampered by the un-availability of suitable liver donors. Several methods have been developed and applied to find a better system to prioritize recipients on the waiting list. Based on this concept, Feng [6] proposed a donor risk index (DRI), aimed at establishing the quantitative risk associated with the sole use of combinations of donor characteristics.

* Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 397–408, 2012.

Many reasons have motivated this study, for example, methods developed to handle this problem have separately considered the characteristics of donors, recipients and transplants. Also, developing a matching donor-recipient system offers the possibility of predicting the outcome when a specific donor liver is allocated to a specific recipient. This paper considers a liver transplant binary dataset obtained from eleven Spanish hospitals, including the graft survival or rejection in liver transplantation three months after the operation in order to develop a complete system for donor-recipient matching.

Several machine learning techniques (especially binary classifiers) have been implemented and tested for the prediction of graft survival. Our analysis will focus on comparing mono versus multi-objective techniques taking into account two well-known measures, i.e. analysing which method obtains a better relation between these metrics (which will be defined later) using both mono and multi-objective algorithms. For this purpose, we have implemented a hybrid multi-objective algorithm combining Artificial Neural Networks and Evolutionary Computation, using a local search.

The paper is organized as follows: Section 2 shows a description of the methods used, both for attribute selection and classification (mono-objective and hybrid multi-objective ones); Section 3 describes the specific characteristics of the dataset and the experimental study; Section 4 describes the results obtained from a mono and multi-objective point of view; and finally, Section 5 outlines some conclusions and future work.

2 Method

2.1 Data Pre-processing

Feature extraction is one of the most important pre-processing steps in pattern recognition, due to the fact that it is an effective dimensionality reduction technique and also an essential method to remove noise features.

Predicting the first days post-transplantation graft survival may lead us to understand and improve the usual donor-recipient matching procedure. But organ transplantation procedures involve a large number of variables that may have an insignificant impact on the survival of the graft and/or the patient. The omission of the vast majority of variables may hinder the discovery of underlying relationships between survival and related factors. So, in this paper, due to the great number of characteristics in the database we have applied a hybrid supervised attribute filter for the selection of the more discriminant variables. For this purpose the evaluators CfsSubsetEval and FilteredSubsetEval [10] were used, both with the BestFirst search method.

The final selected attributes are a combination of those selected by both of the feature selection techniques. As can be seen in the Results section, the application of attribute selection not only decreased the execution costs, but also improved classification results in the majority of the methods selected.

2.2 Multi-objective Classification Using Memetic Pareto Evolutionary Neural Networks

This paper constructs a hybrid, evolutionary and multi-objective algorithm using neural network models with q-Gaussian basis units for classification, MPDENN (Memetic Pareto Differential Evolutionary Neural Network algorithm); we compare its efficiency case with another multi-objective method (Memetic Pareto Evolutionary NSGA2) and other well-known mono-objective algorithms.

Evolutionary artificial neural-networks [14] have been widely used in investigation fields like computer science or statistics, especially in machine learning or classification problems showing good performance, and present a good alternative to some conventional classification methods [4].

Multi-objective Evolutionary Algorithms are a type of solution-searching method based on principles of biological evolution. These methods analyse and mix some possible solutions for the problem and best solutions prevail over time. Often, a number of objectives must be processed to obtain a viable solution for a given problem. We use the evolutionary algorithm in combination with a Pareto front optimization methodology [1] is used to deal with this problem.

Fig. 1 shows the solution chosen by these multi-objective algorithms. The non-feasible region for solutions is the one outside of the triangles. Minimum sensitivity (MS) is represented on the horizontal axis and accuracy (CCR) on the vertical axis. One point in space dominates another if it is more accurate and has equal or greater sensitivity, or if it has greater sensitivity and equal or better accuracy. The use of evolutionary Pareto-based algorithms when training an Artificial Neural Network may lead to an improvement in the accuracy of the model, helping it to escape local optima. The two algorithms explained in this section are both hybridised with a Local Search Algorithm, and are based on a two dimensional measure associated with the confusion matrix: CCR and MS, i.e. the global performance for the dataset and the performance in each class.

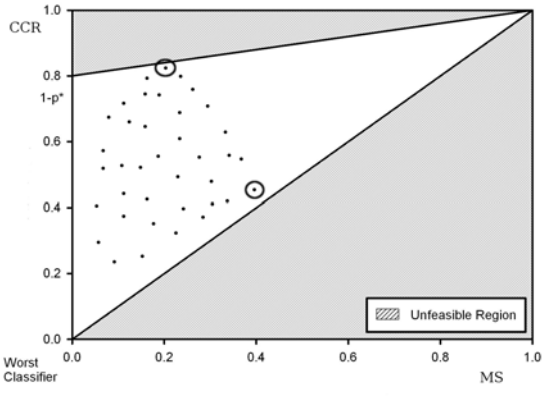


Fig. 1. Solutions chosen by the Pareto front. In this case the first objective is MS and the second one CCR.

The latter objective is not usually optimized in classification, but it is considered here given the need to obtain high precision in each class of real problems. The pair composed by the CCR and MS measures tries to find a point between the scalar accuracy metric and the multidimensional equivalent one (MS) which is based on missclassification rates. These measures verify that:

$$MS \leq CCR \leq 1 - (1 - MS)p^*,$$

where p^* shows the minimum estimated prior probabilities.

Once the first Pareto front is calculated using the training patterns, the best individual taking into account the Correct Classification Rate measure is chosen for the algorithm, and the best individual in terms of Minimum Sensitivity is also selected. Once this is done, the values of CCR and MS are obtained using the testing set.

MPDENN. This section presents the constructed hybrid, multi-objective algorithm for classification in liver transplantation. It is based on Differential Evolution (DE) algorithm developed by R. Storn and K. Price in [13] and modified by H. Abbass to train neural networks [1]. DE is adapted by analysing the trade-off between the CCR and MS in [7]. The fundamental bases of the MPDENN algorithm are Differential Evolution and the concept of Pareto dominance. So, in order to solve this complex real-world problem of donor-recipient allocation, we use the MPDENN algorithm [5], training ANNs and using a specific type of basis functions called q-Gaussian Radial Basis Functions [9].

The steps of the algorithm are explained in Fig. 2. As can be seen, MPDENN generates a random population P_0 of N elements. The population is sorted according to the concept of non-domination, so dominated individuals are deleted from the population. Then, a procedure of selection, crossover and mutation is applied and finally the population is completed with new offspring generated from the selected individuals in the population. In some generations a method combining a local search with the K-means algorithm is applied for more representative individuals. The algorithm ends when the maximum number of generations is reached.

Evolutionary Algorithms carry out a global search inside the solution space, the solutions approaching the global optimum. So, in this case, the local procedure can quickly and efficiently find the best solution.

The local search is only applied in three different generations: at the beginning, in the middle of the execution and at the end once the population is completed. In this paper, the local search algorithm used is the *iRprop*$^+$ algorithm [12]. Thus, the iRprop$^+$ algorithm is not applied to all individuals, only to the most representative ones. This procedure can be seen in Figure 3.

The process for selecting these representative individuals depends on the number of individuals in the first Pareto front. If this number is lower than or equal to the desired number of clusters (num), a local search is applied to all

```
Create a random population P_0
while no stop condition do
      Evaluate and adjust population
      while population not complete do
            Selection
            Crossover
            Mutation
            Evaluation
            Add children in the population according to dominance relationships with the main parent
      end while
      if local search do
        if number of individuals in the first Pareto front of the population < num do
          apply iRprop+
        else
          generates num cluster with K-means
          apply iRprop+
        end if
      end if
end while
```

Fig. 2. Procedure of MPDENN algorithm

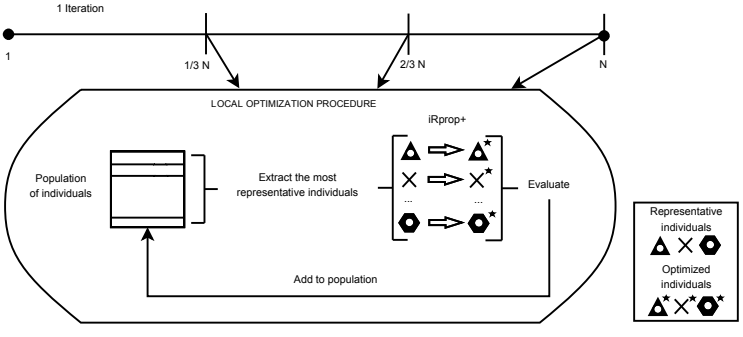


Fig. 3. Local search procedure Scheme

individuals of the Pareto front without using the K-means algorithm. On the other hand, if the number of individuals is greater than *num*, the K-means procedure is applied to get the most representative *num* individuals, who will then be the object of a local search.

MPENSGA2. This section briefly explain a another hybrid multi-objective algorithm which uses the multilayer perceptron based on the NSGA2 evolutionary algorithm (MPENSGA2) proposed in [8]. This algorithm will be used to compare different hybrid, evolutionary and multi-objective algorithms. This algorithm seems to be an interesting method because it applies a pruning procedure which makes the resulting model more interpretable.

This method designs artificial neural network models, simultaneously optimising the structure and parameters of each network. It also introduces an algorithm for local search in order to improve the generalisability of multi-classifiers and

the learning process. The MPENSGA2 algorithm obtains different sets of non-dominated classifiers which are well-balanced taking into account CCR and MS, the two objectives to optimize.

2.3 Mono-Objective Algorithms

Mono-objective optimization aims to optimize a single objective, i.e., a well-defined target, to the detriment of all other possible targets. The hybrid multi-objective methods developed are compared with some mono-objectives ones, like Support Vector Machines, Kernel Discriminant Analysis or Simple Logistic, methods explained in this section.

We have compared the MPDENN algorithm to eight state-of-the-art methods well known in the literature, five of which have been configured and run in WEKA [11]. The LibSVM method is available on a website as a continuous updated software library for Support Vector Machines and Linear Discriminant Analysis, and its kernel approach has been especially implemented in MATLAB Software for the specific problem.

The methods used for comparison are:

- MLogistic: Classifier for building a multinomial logistic regression model with a ridge estimator.
- Simple Logistic: Classifier for building linear logistic regression models.
- C4.5: Classifier for generating a pruned or unpruned C4.5 decision tree.
- Multilayer Perceptron: A neural network classifier that uses backpropagation to adjust the weights. The nodes in this network are all sigmoid.
- LMT: Classifier for building logistic model trees, which are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values.
- Support Vector Machines: Software package for the optimization of Support Vector Machines (SVM). This library contains a script for automatically adjusting the hyperparameters associated with this kind of model, including the cost parameter and the width of the Gaussian kernels. The library searches for the best hyper-parameter values using a grid search and choosing the best configuration by a 10-fold cross-validation process.
- Linear Discriminant Analysis: Discriminant Analysis is a well-known method used in statistics, machine learning, and pattern recognition to decrease data dimensionality and choose the best linear combination of characteristics to best separate the data. It was proposed by Fisher in 1934. The objective of this technique is to choose the optimal projection of the data, i.e. the projection which best separates this data.
- Kernel Discriminant Analysis: it is to map the original finite dimensional space in a higher dimensional one, where a lineal separation of the patterns may be carried out. This can be done using a kernel function (in this case, the Gaussian one). For more information about Linear Discriminant Analysis and its kernel version, see [2].

3 Experimental Study

3.1 Dataset Description

A multi-centered retrospective analysis was made of 11 Spanish units of liver transplantation. Recipient and donor characteristics were reported at the time of transplant. Patients undergoing partial, split or living-donor liver transplantation and patients undergoing combined or multi-visceral transplants were excluded from the study. All patients were followed from the date of transplant until either death or graft loss. Units of liver transplantation were homogeneously distributed throughout Spain.

So, a database with 114 patterns (Donor-Recipient pairs) corresponding to the years 2007 and 2008 in eleven Spanish hospitals is used to perform the classification problem. 16 recipient characteristics, 16 donor characteristics and another 9 operative factors were reported for each pair, as for example the donor and recipient age and sex or the presence of different diseases among others.

To solve this donor-recipient matching problem, the dependent variable is chosen whether there is graft survival after 15 days or not. This is a binary variable equal to 0 when representing graft rejection up to the first 15 days after the transplantation and equal to 1 if the rejection occurs between 15 days and 3 months. The choice of the first fifteen days for the dependent variable is not arbitrary. The data showed that the fifteen days after the transplantation is a critical point for the survival or rejection of the graft, which is supported by the fact that more than the 50% of rejections occurr in the first fifteen days. Fig. 4 shows the cumulative frequency of the rejection of the graft where the slope of the line strongly changes somewhere around the first fifteen days.

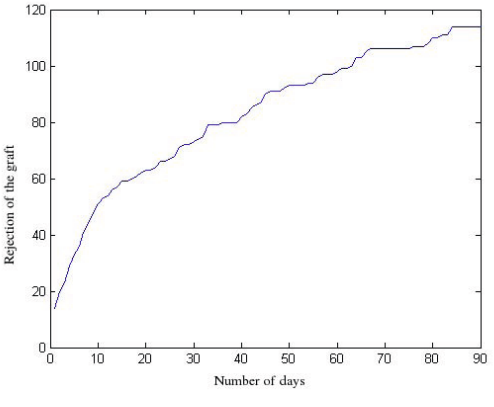


Fig. 4. Graphic showing the cumulative frequency of the rejection of the graft

3.2 Experimental Design

Two different experimental designs based on cross-validation have been used to estimate the performance of each method.

- We used a 30 holdout method for the deterministic methods, i.e. they do not depend on a stochastic variable, like Logistic, Simple Logistic, SVM, C4.5, LDA and KDA.
- However, a 10 holdout with 3 repetitions was used for the non-deterministic methods as the Multilayer Perceptron, MPDENN and MPENSGA2.

The split percentage was 75% for training and 25% for testing. Also, a cross-validation technique was used based on a mesh model for optimizing the parameters of each method and each holdout.

3.3 Statistical Analysis

Four metrics are used to analyse the efficiency of the different methods [3]: CCR, MS, $RMSE$ and AUC. CCR and MS represent threshold metrics, $RMSE$ a rank metric and AUC is a probability metric.

CCR **or Correct Classification Rate:**

$$CCR = \left(\frac{1}{N}\right) \sum_{n=1}^{N} (I(\mathbf{x}_n = y_n)),$$

where $I(\cdot)$ is the zero-one loss function, y_n is the desired output for pattern n and N is the total number of patterns in the dataset. A good classifier tries to achieve the highest possible CCR in a given problem.

MS **or minimum sensitivity:** This metric can be defined as the minimum value of the sensitivities for each class,

$$MS = \min\{S_i;\ i = 0, \ldots, K\},$$

where S_i is the sensitivity for the ith class. Sensitivity for class i corresponds to the correct classification rate for this specific class.

$RMSE$ **or Root Mean Square Error:** It measures to what degree predictions deviate from the true targets.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2},$$

where y_n is the real value and $\hat{y}_n$ is the estimated value.

AUC **or Area Under Curve:** The AUC of a binary classifier is equivalent to the probability of the classifier will ranking a randomly chosen positive instance higher than a randomly chosen negative instance

$$AUC = \frac{1}{N_0 N_1} \sum_{n=1}^{N_0} \sum_{m=1}^{N_1} c(\mathbf{p}_n, \mathbf{p}_m),$$

where N_0 is the number of examples for class GS and N_1 for class GNS, $\mathbf{p}_n = [p_{0n}, p_{1n}]$ the estimated probability vector for members of class GS and $\mathbf{p}_m = [p_{0m}, p_{1m}]$ the estimated probability vector for members of class GNS, being $c(\mathbf{p}_n, \mathbf{p}_m) = 1$ if $p_{0n} > p_{0m}$ and 0 otherwise.

In [8] there is a study of sensitivity versus accuracy, which demonstrates the fact that these metrics occur after certain levels, in conflict in the optimization process. This specific case shows that they are quite related, i.e. a relation between them can be shown.

4 Results

Table 1 shows the results of different classification methods for the complete dataset, that is, the dataset without carrying out a selection of characteristics. These results can be compared with the ones obtained in Tables 2 and 3 (where the attribute selection is applied). The results show that the attribute selection improves the classification rates in almost all the selected methods.

Table 1. CCR, MS, $RMSE$ and AUC from different methods for the complete dataset

	CCR	MS	$RMSE$	AUC
J48	**55.21 ± 8.59**	*45.98* ± 10.02	**0.659 ± 0.064**	**0.563 ± 0.085**
Logistic	*54.48* ± 8.72	45.13 ± 10.60	*0.672* ± 0.064	*0.546* ± 0.087
LMT	54.29 ± 7.83	44.88 ± 10.98	0.674 ± 0.057	0.543 ± 0.078
SimpleLogistic	54.05 ± 7.44	44.76 ± 11.12	0.676 ± 0.054	0.541 ± 0.074
MultilayerPerceptron	53.54 ± 7.88	44.51 ± 10.85	0.679 ± 0.057	0.535 ± 0.079
LibSVM	51.84 ± 7.60	37.24 ± 16.53	0.692 ± 0.055	0.518 ± 0.077
KDA	51.38 ± 7.96	**49.91 ± 7.89**	0.695 ± 0.056	0.511 ± 0.079
LDA	49.65 ± 8.55	47.65 ± 9.19	0.707 ± 0.059	0.496 ± 0.085

Table 2 shows the performance of the mono-objective and non-hybrid methods and Table 3 shows instead the performance obtained in the multi-objective and hybrid methods, for both sides of the Pareto front (the ones chosen are represented in Fig. 1).

Table 2. CCR, MS, $RMSE$, and AUC of all the selected mono-objective methods

	CCR	MS	$RMSE$	AUC
J48	55.17 ± 9.75	44.52 ± 14.53	0.665 ± 0.074	0.553 ± 0.096
Logistic	54.31 ± 8.60	44.24 ± 12.16	0.672 ± 0.065	0.544 ± 0.085
LMT	55.17 ± 8.71	45.06 ± 11.71	0.666 ± 0.066	0.552 ± 0.086
SimpleLogistic	55.77 ± 8.75	45.73 ± 11.58	0.662 ± 0.067	0.558 ± 0.086
MultilayerPerceptron	*56.04* ± 8.91	46.55 ± 12.03	*0.659* ± 0.069	*0.562* ± 0.088
LibSVM	50.46 ± 5.69	35.81 ± 12.15	0.703 ± 0.041	0.503 ± 0.056
KDA	**56.20 ± 6.64**	**50.54 ± 6.08**	**0.655 ± 0.052**	**0.584 ± 0.079**
LDA	52.64 ± 7.78	*49.69* ± 7.79	0.686 ± 0.056	0.527 ± 0.077

For $MPDENN$ and $MPENSGA2$ the population size is established at 50 and the number of generations at 300. The number of hidden neurons is between 2 and 6. Other parameters to each method can be found respectively in [5] [8].

Table 3. CCR, MS, $RMSE$, and AUC of all the multi-objective methods selected

	Pareto	CCR	MS	$RMSE$	AUC
MPENSGA2	CCR	51.26 ± 8.99	43.73 ± 10.94	0.565 ± 0.045	0.525 ± 0.092
MPENSGA2	MS	54.14 ± 7.85	46.09 ± 9.71	0.549 ± 0.042	0.553 ± 0.089
MPDENN	CCR	**59.16 ± 9.43**	**49.65 ± 13.02**	*0.529* ± 0.058	**0.604 ± 0.117**
MPDENN	MS	*56.89* ± 9.49	*47.19* ± 11.55	**0.528 ± 0.057**	*0.601* ± 0.119

The method which generally obtains better performance is MPDENN, followed by KDA and Multilayer Perceptron. As seen in the following Pareto front (Fig. 6), there is some kind of linear correlation between these metrics, which means that the optimization of them is related.

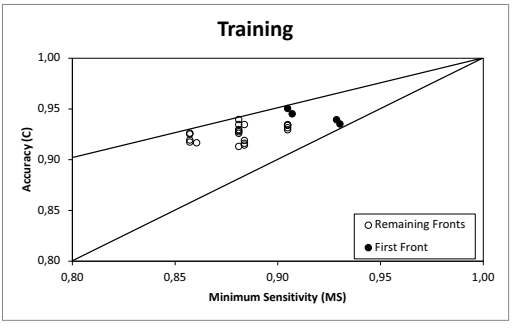


Fig. 5. Graphic showing the Pareto front for training

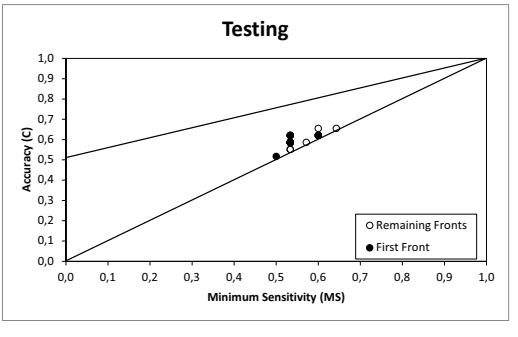


Fig. 6. Graphic showing the Pareto front for testing

To study the significance of the mean differences obtained, a parametric test has been applied (Student's t in this case) comparing KDA versus the MPDENN method, the algorithms which have obtained the best performance for the specific problem. First of all, we checked that the conditions for the parametric test are

satisfied: independence of the sample, same variances (using the Levene test), and normal distribution of the sample (Kolmogorov-Smirnov test). The results of the Student's t test show the fact that, in the case of CCR and $RMSE$ metrics, the means are significantly different because the associated p-values are respectively 0.070 and 0.001, greater than 0.10 (significance level). On the other hand, there are not significant differences taking MS and AUC measures into account. So, analysing the results of the algorithms for the four selected metrics we can conclude that there are some significant differences in mean between MPDENN and KDA for CCR and $RMSE$, favouring in this case MPDENN. To sum up, this hybrid algorithm shows better performance although the mean result for MS is higher with KDA.

In conclusion, the application of a multi-objective point of view can be interesting even when the two selected objectives are quite related. The results evidence that a multi-objective approach lead us to improve the solutions.

5 Conclusions

We can conclude for this case that although the Correct Classification Rate and Minimum Sensitivity are correlated objectives, i.e. some kind of linear relation can be seen between the solutions in the Pareto fronts, a multi-objective approach using Artificial Neural Networks is preferred in order to explore the complete set of solutions for the classification. The MPDENN algorithm has shown good performance for this specific database, improving the results of some state-of-the-art methods. Also, the conclusion drawn may be that in some really complex problems, as in this case, it could be interesting to apply some feature selection to remove noise from the data. It is important to analyse the problem we are dealing with before applying a mono or a multi-objective methodology, and clarify which are the metrics we want to maximize, in this specific case, we want a good balance between Correct Classification Rate and Minimum Sensitivity. As future works the database could be analysed taking into account two different classes: graft failure before and after three months after transplantation. Patterns classified in first class could be analysed with the obtained models in this paper.

Acknowledgments. This work has been partially subsidized by the Spanish Inter-Ministerial Commission of Science and Technology under Project TIN2011-22794, by the European Regional Development fund, by the "Junta de Andalucía" (Spain) under Project P2011-TIC-7508. Manuel Cruz-Ramírez's research has been subsidized by the FPU Predoctoral Program (Spanish Ministry of Education and Science), grant reference AP2009-0487, and by Astellas Pharma.

References

1. Abbass, H.A., Sarker, R., Newton, C.: Pde: A pareto frontier differential evolution approach for multiobjective optimization problems. In: Proceedings of the Congress on Evolutionary Computation, pp. 971–978. IEEE (2001)

2. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus (2006)
3. Caruana, R., Niculescu-Mizil, A.: Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004, pp. 69–78. ACM (2004)
4. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, 1st edn. Cambridge University (2000)
5. Cruz-Ramírez, M., Sánchez-Monedero, J., Fernández-Navarro, F., Fernández-Caballero, J.C., Hervás-Martínez, C.: Memetic pareto differential evolutionary artificial neural networks to determine growth multi-classes in predictive microbiology. Evolutionary Intelligence 3(3-4), 187–199 (2010)
6. Feng, S., Goodrich, N.P., Bragg-Gresham, J.L., Dykstra, D.M., Punch, J.D., DebRoy, M.A., Greenstein, S.M., Merion, R.M.: Characteristics associated with liver graft failure: the concept of a donor risk index. Am. J. Trans. 6(4), 783–790 (2006)
7. Fernández, J.C., Hervás, C., Martínez, F.J., Gutiérrez, P.A., Cruz, M.: Memetic Pareto Differential Evolution for Designing Artificial Neural Networks in Multiclassification Problems Using Cross-Entropy Versus Sensitivity. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 433–441. Springer, Heidelberg (2009)
8. Fernández-Caballero, J.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., Gutiérrez-Peña, P.A.: Sensitivity Versus Accuracy in Multiclass Problems Using Memetic Pareto Evolutionary Neural Networks. IEEE Transacctions on Neural Networks 21(5), 750–770 (2010)
9. Fernández-Navarro, F., Hervás-Martínez, C., Gutiérrez-Peña, P.A., Carbonero-Ruz, M.: Evolutionary q-Gaussian Radial Basis Functions Neural Networks for Multi-Classification. Neural Networks (2011)
10. Guyon, I.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newsletter 11, 10–18 (2009)
12. Igel, C., Husken, M.: Empirical evaluation of the improved rprop learning algorithm. Neurocomputing 50 (2003)
13. Storn, R., Price, K.: D. evolution: A simple and efficient heuristic for global optimization over continuous spaces. J. of G. Optimization 11, 341–359 (1997)
14. Yao, X.: A review of evolutionary artificial neural networks. International Journal of Intelligent Systems 4, 203–222 (1993)

Evolutionary Optimized Forest of Regression Trees: Application in Metallurgy

Mirosław Kordos[1], Jerzy Piotrowski[1], Szymon Bialka[1],
Marcin Blachnik[2], Slawomir Golak[2], and Tadeusz Wieczorek[2]

[1] University of Bielsko-Biala, Department of Mathematics and Computer Science,
Bielsko-Biala, Willowa 2, Poland
mkordos@ath.bielsko.pl
[2] Silesian University of Technology, Department of Management and Informatics,
Katowice, Krasinskiego 8, Poland
marcin.blachnik@polsl.pl

Abstract. A forest of regression trees is generated, with each tree using a different randomly chosen subset of data. Then the forest is optimized in two ways. First each tree independently by shifting the split points to the left or to the right to compensate for the fact, that the original split points were set up as being optimal only for the given node and not for the whole tree. Then evolutionary algorithms are used to exchange particular tree subnodes between different trees in the forest. This leads to the best single tree, which although may produce not better results than the forest, but can generate comprehensive logical rules that are very important in some practical applications. The system is currently being applied in the optimization of metallurgical processes.

Keywords: Decision tree, regression, evolutionary optimization, logical rules.

1 Introduction

In is not an easy task to find an appropriate trade-off between the system performance and comprehensibility. Complex hybrid systems can perform better than simple decision trees, but it is very hard to extract comprehensive logical rules from them [2,3,4]. In our paper presented at the last HAIS conference in 2011 we considered optimization of regression tree parameters [1]. Although the logical rules were very clear and allowed for better understanding of the process, we were able to obtain much higher accuracy for the same task using an ensemble of neural networks with evolutionary optimization [5]. In [6] a heterogeneous forests of decision trees for classification tasks was presented to address this issue. The purpose of our current work is to join the two features (comprehensibility and accuracy) for regression tasks in one model as far as possible.

The introduction outlines the problems associated with regression trees and their committee optimization and with temperature control in electric arc steel-making. The second section ("Methodology") presents the methodology of costructing and training the regression model, including data preprocessing, decision tree construction algorithm and evolutionary methods of optimizing the decision tree and the decision tree forest.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 409–420, 2012.

The third section ("Experimental Evaluation") presents the way the numerical experiments were performed and the results obtained on several real-word datasets. The last section ("Conclusion") concludes the paper.

1.1 Regression Tree and Their Committee Optimization

Decision trees produce very comprehensive rules although the accuracy of neural networks accuracy can be higher. Rudy Setiono in his works presented an approach to logical rule extraction from standard multilayer perceptrons (MLP) for regression problems [8]. Although the accuracy of the rules obtained from his approach is only very little below the accuracy of the underlying neural network, the complexity of the rules is so high, that it may prevent the practical application of the system in the industry.

The idea behind building a committee of regression models is to improve the accuracy of single models by weighting the outputs of particular models, where the sum of weights equals one. That works, because the bias of the whole committee is of the same level as biases of particular models, while the variance is smaller or at least not larger than variances of particular networks [9].

Bagging [10] provides decorrelation between the predictions of committee members by training each one on a different dataset, obtained by randomly drawing with replacement n data points from the original data set. In AdaBoost [10] the probability of selecting a vector in a next model is higher if the previous model made higher error on that vector.

Barbosa [11] used a population of feed-forward neural networks, which was evolved using the Clonal Selection Algorithm and the final ensemble was composed of the selected subset of the neural network population.

Chen and Yao [12] proposed a Negative Correlation Learning, which introduces a correlation penalty term to the cost function of each network so that each network minimizes not only its mean square error (MSE) but also the correlation with other networks. They used and evolutionary multi-objective algorithm to optimize the networks in the ensemble.

Baruque and Corchado [13] instead of combining directly the results, formulated a single resultant model of the SOM networks, where the neurons composing the map were obtained as the fusion of neurons from different maps that are considered close enough to be fused into one.

The M5 [14] algorithm is a greedy top-down induction method with the following tree structure: univariate splits and multivariate linear models at the leaves.

Malebra et. al. [15] presented a regression tree model, which at the same time optimizes the split points and the linear regression models not only at the leaves but also at the upper parts of the tree. It improves the prediction accuracy of the tree but makes the extracted rules much more complex.

Czajkowski and Kretowski [16] presented an evolutionary induction of decision trees, where trees are represented as typical univariate trees and each test in a non-terminal node concerns only one attribute. They initially created a set of decision trees, each of the only on a part of the whole dataset. Then using genetic algorithms where randomly selected pairs of trees exchanged some nodes with the whole branch starting from the node as long as the performance of the best tree was improving.

In our previous system we used a single regression tree with appropriate data transformations [1] to obtain logical rules depicting the process and a forest of regression trees

Because of the greedy nature, decision tree induction algorithms may not generate the smallest possible trees and thus the smallest possible number of rules for a given problem [17] and that in turns decreases their comprehensibility. For that reason a global approach can lead to better prediction and smaller size of the tree [16]. And therefore we use the shifting of split points and evolutionary optimization to obtain more efficient models.

1.2 Temperature Control in the Electric Arc Furnace

In the electric arc furnace the steel scrap is melted using the electric arc to generate most of the heat. Additional heat is obtained from gas that is inserted and burnt in the furnace. The optimal temperature of the melted steel that is to be tapped out from the furnace is about 1900K, however it must be kept at proper temperature enough long so that all the solid metal gets melted. If the heating lasts too long, unnecessary time and energy is wasted and additional wear of the furnace is caused. Modern EAFs have the melt times of 30 minutes, older ones up to one hour.

The temperature is measured a few times during every melt by special lances with thermocouple that are inserted into the liquid steel. Every measurement takes about one minute and in this time the arc has to be turn off and the process suspended. Waste of time and energy for three or even more measurements is thus quite significant. There are many problems with the continuous measurement of the steel temperature. The temperatures are very high and the radiant heat and the electro-magnetic radiation from the furnace creates major problems for the measuring equipment.

Therefore there was a need to build a temperature prediction system that would allow us to limit the number of temperature measurements and thus shorten the EAF process. We previously built a system based on a single regression model [1], as a part of the whole intelligent system for steel production optimization [18]. However, as our recent experiments showed, a significant improvement can be obtained with a forest of decision trees where all the member trees underwent post-training optimization of the split points or even with w single regression tree, which is build by genetic algorithm-based optimization of the forest.

2 Methodology

Although it takes some time to build and optimize the model (up to several hours on a single CPU), the time of training the model is not important in our application, because the model has to be created only once. When the model allows to shorten the steel production cycle on average by half a percent, the yearly gain of using it at one electric arc furnace can be quite significant. Thus what we need is a model that works fast, is accurate and produces comprehensive logical rules, but which does not necessarily learn fast.

Two models that we have built are presented in this section: a single tree with genetic optimization and split point optimization and a forest with split point optimization and evolutionary optimization of the voting.

The system works in the following way:

1. Perform data transformations
2. Build tree forest
3. Optimize each tree by shifting the split points -> Ensemble Optimization
4. Perform the ensemble output optimization
5. Perform inter-tree optimization by exchanging the branches -> Genetic Algorithms, single tree
6. Finally optimize the best tree by shifting the split points
7. Use the best tree for rule prediction and continuous value prediction

2.1 Data Transformation

The presence of outliers and wrong data may dramatically reduce generalization abilities and limit the convergence of training process of any learning algorithm. Proper data preprocessing is helpful not only to deal with outliers but also to achieve variable sensitivity of the model in different ranges of input variables. First we standardize the data before the training, according to the following formula:

$$x_{std} = \frac{x - \bar{x}}{\sigma} \quad \sigma = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(x - \bar{x})^2} \tag{1}$$

and then we transform the data be the hyperbolic tangent function

$$y2 = \frac{1 - exp\,(-\beta \cdot y + \theta)}{1 + exp\,(-\beta \cdot y + \theta)} \tag{2}$$

where y is the output value before the transformation and $y2$ - after. Although one univariate regression trees do not need the standardization of inputs, because they consider only a single attribute at a time, we perform the standardization, because it is easier to understand the process and compare the influence of various attributes if the data is standardized. In practical problems, it is frequently desired to obtain a model with higher sensitivity in the intervals with more dense data (as it was in our case) or in other intervals of special interests. To address his issue, we transfer the data through a hyperbolic tangent function. The other advantage of the transformation is the automatic reduction of the outliers' influence on the model. Because if the complexity of our data and lot of error in particular value measurements, before the values were recorded into the database it is very difficult to properly apply known outliers removal methods, such as Edited Nearest Neighbor or others. For that reason we do not do not reject the outliers [1], but rather reduce their influence on the final model, because it is frequently not clear whether a given value is already an outlier or wrong value or is still correct. The hyperbolic tangent transformation allows for a smooth reduction of the outliers, because even very big values after the transformation will never be grater than one or smaller than minus one. However, in the case of multimodal data distribution the data should be first divided into several single-mode distribution datasets or a more complex transformation function should be used.

2.2 Tree Construction and Split Criteria

We use decision trees with univariate non-terminal nodes and at each leaf a multivariate linear model is constructed using standard regression technique. However, to keep the model simple, the regression model is limited to only these three attributes, which have the greatest correlation coefficient with the output value in the node. The non-terminal nodes are split always into two child notes in a way that maximizes the the purpose o the parameterization function. To achieve this we had to find such an attribute (feature) f_0 over all possible attributes f_i and such a split point s_0 of that attribute, which maximize the variance reduction v for each tree node. In the equation below v_0 is the node variance, i.e. the variance of all vector output values Y in the node. We search for the optimal split point (s_0) iterating over each input feature f_j and each value of that feature j. For that reason the vectors must be sorted in the increasing order of each feature separately, before the search for the optimal split is attempted. v_L is the variance of the left side of the node (the potential left child) and v_R of the right side. After performing some experiments to find optimal split function (see [1]), we determined that in most cased the following split criteria is sufficient:

$$v = v_0 - (p_L \cdot v_L - p_R \cdot v_R) \tag{3}$$

where p is the number of vectors in the given node, p_L is the ratio of the number of vectors in the left child node and the total number of vectors in that node (p) and respectively p_R is the ratio in the right mode. The split points determined in this way do not maximize the variance reduction at the level of a single node, but additionally enforce rather symmetrical splits such maximizing the performance of the whole tree. For more see [1]. Nerveless, because of the local search only (at the level of single nodes) they are not guarantied to be optimal split points and for that reason after the construction of the decision tree is finished, we will further perform optimization of the split points, taking into account the global performance of the whole tree.

2.3 Forest Construction

There are ways to improve the prediction ability of a single tree. One of the methods is to create a forest of trees. In our experiments we used a forest of N=64 trees. The whole training set consisted each time of a 90% of the total set and the whole model was tested in 10-fold crossvalidation, each time on the remaining 10% of vectors. From 10 to 30 percent of training vectors (depending on the dataset size) were randomly chosen to build each of thee trees. Each tree was then tested on the remaining training vectors and the inverse of MSE achieved on this set was the quality measure of the tree. In the simplest case the final decision y is taken based a weighted average of the values predicted by all the trees:

$$y = \left(\sum_{i=1}^{N} \frac{1}{MSE_i} \right)^{-1} \sum_{i=1}^{N} \frac{y_i}{MSE_i} \tag{4}$$

In the following sections we present how the decision weighting scheme can be further optimized.

2.4 Optimization of Split Points

The idea behind the optimization of the split points is to find a minimum of the error function, where the arguments of the functions are the split points at every node and the value of the function is the MSE on the training set. Where the tree was originally created, the split points at each node minimized the the variance at one pair of nodes (the child nodes of the current node) but not a global variance on the whole tree. Thus, we try to minimize the MSE on the whole tree, while keeping its size constant and keeping constant the attributes that split each node. The split point at one node is shifted to the next or previous value of the split attribute and then all the subtree that originates from a given node is recreated. If the MSE decreases - the change is kept, otherwise it is rejected and a change in an opposite direction is attempted. If an improvement was achieved by shifting the split points by one distinct attribute value, then another attempt is made to change the shift about one more value in the same direction. This is done recursively from the top to the bottom of the the tree.

2.5 Evolutionary Optimization of Forest Decision

The purpose ot the optimization is to find the optimal weights w_i by which the output of particular trees is multiplied in order to obtain the lowest possible error of the value predicted by the forest y:

$$y = \left(\sum_{i=1}^{N} \frac{1}{w_i} \right)^{-1} \sum_{i=1}^{N} \frac{y_i}{w_i} \tag{5}$$

The methodology used in this work is based on the solution presented in [5] to optimize the decision of neural network ensemble. All the trees in the forest are constructed only once. Only determining the weights w_i is an iterative procedure. That makes the process relatively fast, because it does not require either re-training or re-testing the decision trees. To find the weights w_i, the following algorithm is used:

1. Create a population of N individuals of randomly generated array of weights from the interval (0,1). Eeach position in an induvidual represents one weight w_i by which the output of the corresponding decision tree is multiplied while determining the final decision of the forest.
2. When summing the weighted outputs of particular trees in the forest, multiply each weight w_i by a value inversely proportional to MSE error made by the tree on the test set (as in eq. 3.). Though this step may be omitted, it accelerates the convergence of the algorithm.
3. Calculate the quality of each forest over all test vectors using as the inverse of the MSE.
4. Four best solutions are preserved and included in the next iteration.
5. The winning forest is determined and in a space of weights w_i a randomly chosen half of the other forests are placed in the minimum along the straight lines connecting the forests with the winner. This idea is partially based on the Self Organizing Migrating Algorithm presented in [21,22].

6. The genetic operations of crossover and mutation are performed on the other half of the forests to generate the next population, as in typical evolutionary algorithms. A random number of parents (from two to four) are used and a random number of crossover points (from the number of parents minus one to six) to generate each child. Multiply crossover frequently performs better than single crossover, however an optimal number of crossings exists as well as optimal number of parents [23,24]. The value of the fitness function ot the i-th forest f_i is expressed by the inverse of the MSE_i for that forest and transformed by the following formula:

$$f_i = max(MSE_i - c \cdot MSE_{avg}, 0) \tag{6}$$

where MSE_{avg} - is the average MSE in the current population, c starts with 0 and is gradually increased from iteration to iteration up to 0.9. That allows for including each forest in the reproduction at the beginning thus now narrowing the population diversity and promoting the best individuals at the end of the iterations, where most individuals tend to have comparable fitness and the diversity of the population is very narrow. The value at a given position in the child individual is the value of the appropriate parent with a randomly added real number tx_d where

$$x_d = r \cdot exp{-c} * x \tag{7}$$

where x is a random real number from 2 to 100 and r is a random number that can take three values: -1,0,1. In this way new values can appear in a similar way as in mutation.
7. Again four best solutions are preserved and included in the next iteration.
8. Repeat steps 3 to 7 as long as a noticeable improvement occurs. In the experimental section the noticable improvement is defined as the improvement in prediction accuracy of 0.03

2.6 Construction of the Best Tree with Genetic Algorithms

When the forest is used as a committee of predictive models, no further optimization of the trees with evolutionary algorithm is performed. The optimization improves the predictive abilities of the trees in the forest, but it also makes the differences between particular trees gradually disappear. And the difference between particular committee members is one of the crucial issues that makes the committee effective. Thus, we use the evolutionary optimization only to create the best single tree, which is the best way to obtain comprehensive logical rules.

The optimization method is a modification of the methodology proposed in [16]. The construction of the forest is the same as presented in the above section. After the forest is created, we begin the process of evolutionary optimization. First two trees are chosen randomly with the probability proportional to their fitness function. Then in each of two trees one node is chosen randomly and subtrees starting in the selected nodes are exchanged or the tests performed at the nodes are exchanged. Two mutation operators are used: shifting the splitting threshold (as discussed above) and changing the non-terminal node into leaf.

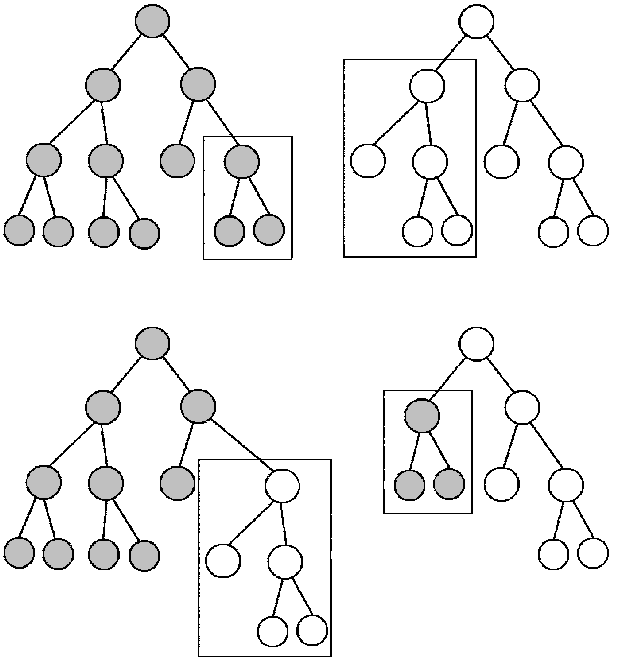


Fig. 1. The idea of exchanging nodes with subtress between two trees

The fitness function value increases if the accuracy of the tree on the test set increases and it also increases if the size (number of nodes) of the tree decreases. In this way the trees which are more accurate and smaller are preferred.

$$fitness = \frac{accuracy^p}{treeSize^r} \tag{8}$$

where the exponent p starts with 1 at the beginning of the optimization in order to keep high variability in the population and is gradually increased to speed up the algorithm convergence at the final stage. Tree size has self-optimization properties to a certain degree, because too big trees have poor generalization and therefore their accuracy on the test set falls down. However, we use this parameter to prefer a smaller tree from two trees of the same accuracy. r is between 0 and 1. If more accurate trees are desired r should be closer to 0 if simpler then r should be close to 1. The split point optimization of the best tree is performed using the same approach as for any tree in the forest optimization.

2.7 Combining the Final Decision

In the terms of accuracy obtained in the experiments, the single best tree after evolutionary optimization performed only slightly poorer than the forest, but because of the comprehensibility of the logical rules it maybe the preferred model. Actually we present

the user with the value predicted by the forest, the value and the rules predicted by the single best tree and suggest that the final prediction of the model that should be used is the average of the forest predicted value and the single best tree predicted value.

3 Experimental Evaluation

3.1 Experimental Methodology

In this section we compare our methods (called in Table 1 "optimized tree" and "optimmized forest") to a single regression tree without optimization, to a forest of regression trees without optimization, an MLP network, and a hierarchical committee of MLP networks.

The regression tree and forest were constructed as described in our paper presented at the previous HAIS conference [1]. The single MLP neural network had the structure $n-n-1$, where n is the number of attributes and was trained with the VSS algorithm [25]. In the hierarchical MLP Committee, the whole dataset was split into several clusters, with a hierarchy of clusters, where the clusters of higher levels contained the clusters of lower levels. Several neural networks were created and trained for each cluster, using bagging to select the vectors. Then an evolutionary algorithm was used to decide how a final decision of the committee must be evaluated based on the test vector properties and on particular neural network responses [5].

We created software that implements all the methods in C#, Java and Delphi languages and made it available together with the datasets used in the experiments from our web page at [19]. All the methods were run in a 10-fold crossvalidation.

The experiments were performed with the following parameters that limited the tree size:

- minimum node variance: 0.002
- minimum number of instances in the current node: 2% of the number of instances in the training dataset
- minimum number of instances in a child node: 0.5% of the number of instances in the training dataset
- maximum number of levels in the tree: 24

3.2 Datasets

We used three datasets depicting the metallurgical problem; one of them refers to the temperature prediction in the EAF process as described in the introduction, one refers to predicting the amount of carbon and one of silicon to be added in the steel refinement process. The other datasets are the Crime, Concrete and Parkinson datasets from the UCI Machine Learning Repository.

The UCI Datasets. We used in the experiment three datasets from the UCI Machine Learning Repository: Concrete Compressive Strength, Parkinsons Telemonitoring, and Crime and Communities [26].

Steel-Carbon and Steel-Silicon. The datasets come from a real metallurgical process at the phase of refining the melted steel in a ladle arc furnace to achieve desired steel

Table 1. Experimental results; mean square error (MSE) and standard deviation in 10-fold cross-validation

dataset		single tree	tree forest	opt. tree	opt. forest	single MLP	committee of MLP
Concrete	MSE	0.153	0.121	0.133	0.120	0.142	0.120
	std. dev.	0.020	0.018	0.015	0.015	0.015	0.014
Crime	MSE	0.35	0.30	0.32	0.26	0.30	0.28
	std. dev.	0.04	0.04	0.03	0.03	0.03	0.03
Parkinsons	MSE	0.105	0.092	0.094	0.088	0.088	0.084
	std. dev.	0.025	0.022	0.025	0.018	0.028	0.018
Steel-C	MSE	0.140	0.122	0.122	0.110	0.118	0.110
	std. dev.	0.017	0.015	0.015	0.012	0.015	0.011
Steel-Si	MSE	0.098	0.072	0.072	0.074	0.095	0.078
	std. dev.	0.012	0.014	0.011	0.013	0.015	0.012
Steel-Temp	MSE	0.70	0.60	0.58	0.54	0.57	0.51
	std. dev.	0.08	0.06	0.06	0.05	0.09	0.05

properties. The inputs variables represent various measured parameters, such as temperature, energy, amount of particular elements in the steel etc. The amount of carbon or silicon that should be added to the steel refinement process is the output variable. The datasets was standardized, only 12 attributes in the Steel-Carbon dataset and 26 attributes in the Steel-Silicon data were left from the original dataset with over 100 attributes and the names of 12(26) input attributes were changed to $x_1 \ldots x_{12(26)}$. There are 1440 instances in both datasets. The datasets are available from [19].

Steel-Temperature. The dataset comes from a real metallurgical process at the phase of melting the steel scrap in the electric arc furnace. The inputs variables represent various measured parameters, such as temperature, energy, amount of gases, etc. The temperature of the liquid steel is the output variable. The data was standardized, only 14 attributes were left from the original dataset with over 100 attributes and the names of 14 input attributes were changed to $x_1 \ldots x_{14}$. There are 7400 instances in the dataset. The dataset is available from [19].

3.3 Experimental Results

In regression and classification problems, there is no single model which is best for all datasets and models must be chosen for a given task. However, an advantage of decision trees is that it is easy to understand and follow the process of decision making. And that is an important factor is safety-critical applications. Although the hierarchical committee of MLP networks usually performed slightly better than the optimized tree, it did not allowed for extracting logical rules, in the same way as the forest of decision trees did not allow for extracting logical rules. Thus, it cannot be arbitrarily stated that a model with a lower MSE error is always better and in our method to construct the optimized tree we tried to obtain the best possible accuracy with the maximum comprehensibility of the rules.

4 Conclusions

This paper presents an approach to decision tree optimization for regression tasks with a practical application in metallurgical industry. The application of the system requires from it to provide at the same time high prediction accuracy and comprehensive logical rules which explain the decision made by the system. We used the split criteria and the data transformations, which we already described in our previous work, presented at the HAIS 2011 conference. However, previously we used the tree as a part of complex hybrid system together with other predictive models. In order to simplify the final model and increase their comprehensibility, we tried to design such an effective optimization of the trees, so that the decision trees could be used as the only models. This is achieved by a post-training optimization of the decision tree obtained with a combination of local search (while shifting the split points) and global search with evolutionary methods (exchanging tree branches and node tests). The leaves of the tree implement linear regression models with limited number of attributes to simplify the equations. In this way we attempted to avoid local optima, which are really what tree induction algorithms optimize. That allowed improving the tree accuracy and generalization while together with reducing the tree size and thus the transparency of the obtained logical rules. Although the accuracy for the metallurgical problems is still lower than the accuracy obtained with a complex hierarchical system of neural networks, it is much higher than that obtain with the basic decision tree and it still allows for simple logical rule extraction. The system can also be used as a part of a hybrid system with the neural network committee which had the highest accuracy.

The system is being applied in production environment to predict the temperature of steel in the electric arc furnace and thus shorten steel production cycle and decrease the costs at one of the steelworks in our country.

Acknowledgment. The work was sponsored by the Polish Ministry of Science and Higher Education, projects No. 4866/B/T02/2010/38 and 4421/B/T02/2010/38.

References

1. Kordos, M., Blachnik, M., Perzyk, M., Kozłowski, J., Bystrzycki, O., Gródek, M., Byrdziak, A., Motyka, Z.: A Hybrid System with Regression Trees in Steel-Making Process. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 222–230. Springer, Heidelberg (2011)
2. Duch, W., Setiono, R., Zurada, J.: Computational intelligence methods for understanding of data. Proceedings of the IEEE 92(5), 771–805 (2008)
3. Jankowski, N., Grabczewski, K.: Heterogenous Committees with Competence Analysis. In: Fifth International Conference on Hybrid Intelligent Systems, Rio de Janeiro, Brasil, pp. 417–422 (2005)
4. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
5. Kordos, M., Blachnik, M., Wieczorek, T., Golak, S.: Neural Network Committees Optimized with Evolutionary Methods for Steel Temperature Control. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS, vol. 6922, pp. 42–51. Springer, Heidelberg (2011)

6. Grąbczewski, K., Duch, W.: Heterogeneous Forests of Decision Trees. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 504–509. Springer, Heidelberg (2002)
7. Kordos, M., Blachnik, M., Wieczorek, T.: Evolutionary Optimization of Regression Model Ensembles in Steel-Making Process. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 369–376. Springer, Heidelberg (2011)
8. Setiono, R., Thong, J.: An approach to generate rules from neural networks for regression problems. European Journal of Operational Research 155(1) (2004)
9. Tresp, V.: Committee Machines. Handbook for Neural Network Signal Processing. CRC Press (2001)
10. Breiman, L.: Combining predictors. In: Sharkey, A.J.C. (ed.) Combining Artificial Neural Nets. Springer, Heidelberg (1999)
11. Barbosa, B.H.G., Bui, L.T., Abbass, H.A., Aguirre, L.A., Braga, A.P.: Evolving an Ensemble of Neural Networks Using Artificial Immune Systems. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 121–130. Springer, Heidelberg (2008)
12. Chen, H., Yao, X.: Multiobjective Neural Network Ensembles Based on Regularized Negative Correlation Learning. IEEE Trans. on Knowledge and Data Engineering 22, 1738–1751 (2010)
13. Corchado, E., et al.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
14. Quinlan, J.: Learning with Continuous Classes. In: Proc. of AI 1992, pp. 343–348. World Scientific, Singapore (1992)
15. Malerba, D., Esposito, F., Ceci, M., Appice, A.: Top-down Induction of Model Trees with Regression and Splitting Nodes. IEEE Transactions on PAMI 26(5), 612–625 (2004)
16. Czajkowski, M., Kretowski, M.: An Evolutionary Algorithm for Global Induction of Regression Trees with Multivariate Linear Models. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS(LNAI), vol. 6804, pp. 230–239. Springer, Heidelberg (2011)
17. Potts, D., Sammut, C.: Incremental Learning of Linear Model Trees. Machine Learning 62, 5–48 (2005)
18. Blachnik, M., Mączka, K., Wieczorek, T.: A Model for Temperature Prediction of Melted Steel in the Electric Arc Furnace (EAF). In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6114, pp. 371–378. Springer, Heidelberg (2010)
19. http://www.kordos.com/his.html
20. Landwehr, N., Hall, M., Frank, E.: Logistic Model Trees. Machine Learning 95 (2005)
21. Zelinka, I., et al.: Evolutionary Algorithms and Chaotic systems. Springer, Heidelberg (2010)
22. Zelinka, I., Senkerik, R., Oplatkova, Z.: Evolutionary Scanning and Neural Network Optimization. In: 19th International Conference on Database and Expert Systems Application, DEXA, pp. 576–582 (2008)
23. Tsutsui, S., et al.: Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms. In: The 1999 Genetic and Evolutionary Computation Conference, pp. 657–664 (1999)
24. Semya, E., et al.: Multiple crossover genetic algorithm for the multiobjective traveling salesman problem. Electronic Notes in Discrete Mathematics 36, 939–946 (2010)
25. Kordos, M., Duch, W.: Variable Step Search Algorithm for Feedforward Networks. Neurocomputing 71(13-15), 2470–2480 (2008)
26. Blake, C., Keogh, E., Merz, C.: UCI Repository of Machine Learning Databases (1998-2011), http://archive.ics.uci.edu/ml/datasets/

Evolutionary Neural Networks for Product Design Tasks

Angela Bernardini, Javier Asensio, José Luis Olazagoitia, and Jorge Biera

CITEAN - The Automotive Technological Innovation Centre of Navarre,
Calle Tajonar 20, 31006 Pamplona, Spain
{abernardini,jasensio,jolazagoitia,jbiera}@citean.com
http://www.citean.com

Abstract. Standard development process of a product involves several CAE and CAD analyses in order to determine parameter values satisfying technical product specifications. In case of nonlinear behavior of the system, the computational time may quickly increase. In the current study, a new methodology that integrates Neural Networks (NN) and Genetic Algorithms (AG) is introduced to analyse virtual models. The proposed tool is based on different computational , mathematical and experimental methods that are combined together to distil a single tool that permits to evaluate in a few seconds how behaviors of certain product vary when any design parameter is altered. As example, the methodology is applied to adjust design parameters of an exhaust system, showing same accuracy range than FEA, but strongly reducing the simulation time.

Keywords: Virtual Engineering, Neural Networks, Genetic Algorithms.

1 Introduction

Product design is a complex decision-making process usually requiring intense interaction between designers and the designed product. According to the traditional iterated design process, linking high fidelity models, numerical optimization and human interaction is expensive and time consuming. In recent years many efforts are under way to determine a very effective virtual engineering and planning process. Computer simulations and analysis enable to save time developing products and processes while reducing global validation costs and increasing quality. Organizations involved in product design and manufacturing are progressively moving from the traditional "trial and error" cycle approach to virtual simulation, mainly supported by Computer Aided Engineering (CAE) and Computer Aided Design (CAD) tools.

In this context Neural Networks (NN) offer an attractive solution to complex engineering design problems. Parra et al. [1,2] proposed a tool based on NN to reduce the probability to suffer squeal noises, due to coalescence of natural frequencies among different brake components, at the design stage of the product. Indeed, after the teaching - learning process, the NN is capable of offering design

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 421–428, 2012.

decisions oriented to influence the avoidance of certain "dangerous" frequency modes.

Despite the great activity and investigations during last decades, the design of NN for specific applications is still a test and error process, depending mainly on previous experience in similar applications [3,4]. Given a set of training examples, there is probably an infinite number of different networks that can learn to map input patterns into output patterns.

In general, network topology affects network training, generalization and learning time. Experience has shown that larger networks tend to overfit the training data, producing a poor generalization, while overly small neural networks lose the capacity to learn [5,6]. A large neural network requires, generally, more computational time than a smaller one. Furthermore, a smaller network it is usually more desirable for model understanding.

Currently there are no formal methods for directly selecting networks architecture, which remains a very complex task.

Evolutionary computation is a set of global optimization techniques that have been widely used in the last few years for training and/or automatically designing neural networks [7,8]. Historically, Genetic Algorithms (GA) [9] and evolutionary programming [10] are two broad categories of evolutionary computation. The main difference between these two categories is based on the representation of the populations and on the generation to generation alterations.

During the last decades GA has been introduced for solving complex engineering design problems. Jenkins [11] and Hajela [12] solved structure optimum problem using GA; Rao [13] solved optimum problem of actively controlled structures; Deb [14] applied GA in optimal design of a welded beam; Hajela and Lin [15] introduces GA to multi-criterion optimal design.

At this paper a new methodology, based on the integration of GA with NN, is introduced for solving complex engineering design problems. The aim of this development is to provide designers with a tool that can be used to select the best possible solution for a given product in a short period of time.

The structure of the paper is as follows. In the next section, the proposed methodology is presented. In section 3 the method is used to adjust design parameters of an exhaust system as study case. The objective is to provide results that differ no more than 5% in accuracy compared to FEA models. Conclusions and further works are drawn in the last section.

2 Experimental Design

The proposed methodology can be divided into the following steps:

a) Parametric FEM model creation

Data sets for NN training can be obtained by tests or FEA simulations. In order to reduce products development time, FEA cases are used for training. A parametric model including FE model (elements, nodes, material properties, boundary conditions, load cases ...) and model variants has been created. Depending on model complexity, the number of cases may become large, then it

is useful to program automatic analysis input process and data post processing. A MSC PATRAN PCL program has been designed. It permits to automatically create the analysis input files which, in turn, reduces the processing time through a loop process:

- Import original CAD model file. The original geometry is imported into MSC PATRAN pre-postprocessor.
- FE model creation.
- Material model and properties creation. The material models and parameters are defined. After this, PCL applies the properties to each part of the model.
- Load case definition.
- Analysis files creation. The PCL creates an analysis input file in each cycle.

b) Producing training samples

To make the Neural Network the most efficient as possible, DOE (Design Of Experiments) techniques are used for a proper distribution of study cases. Different values are assigned to each of the selected parameters through the parametric FEM model. Thus, the analysis strats in batch mode using MSC NASTRAN 2008r2. As a result of this process the PCL permits to obtain the NN analysis cases. The PCL creates each model in few seconds and generates all files in some hours.

c) NN design

Such network must sutisfy some requirements: it must learn the input data, it must generalize and it must have the minimum size allowed to accomplish the first two tasks. The hybrid algorithm employed for the automatic generation of NN develops the following steps:

1. Create an initial population of neural networks with random topologies. A standard population size of twenty is used. Backpropagation NN type has been selected. Each neural network has hidden layers of sigmoid neurons and one linear output layer. Once the number of hidden layers has been fixed, the number of nodes of each of them is determined as a random number bigger than zero and less than a defined maximum. Initial weights range is randomly determined.
2. Train each individual. As previously mentioned, it is possible that the network ends up overfitting the training data, if the training session is not stopped at the right time. It is possible to identify overfitting using crossed validation [16]: the training cases are split into training subset and validation subset. The training subset is used in the usual way, but the training session is periodically stopped to evaluate the network with the validation set.
3. Select parents from the population. This algorithm applies a roulette wheel selection. The fitness function integrates the validation error: the aptitude function is given by the number of correlated validation cases penalized with the ratio of the number of hidden neurons and the maximum number of hidden neurons.

4. Recombine both parents to obtain two children. A one point crossover which operates on chains of genes representing hidden nodes is implemented.
5. Mutate each child randomly. The mutation operator maintains genetic diversity in the population. The mutation rate decreases with the number of generations, allowing to explore the whole search space at the beginning and to favour convergence at the end.
6. Train each child.
7. Replace childrens into the population.
8. Repeat from step 2 for a given number of generations. Hundred generations are carried out.

In order to evaluate the generalization ability of the resulting architecture, the NN is tested with a new set of data. The algorithm is implemented in MATLAB code. MATLAB neural network toolbox is used.

3 Experimental Results

In order to assess the efficiency of the proposed methodology, the algorithm is applied for adjusting design parameters of a vehicle exhaust system, replacing the traditional CAD and CAE calculations cycle. The exhaust system behavior is strongly dependent on rubber hangers. Its design needs to have into account variables as hanger's position, shape and material that might influence vibration transmission from the engine to the vehicle body. In product development process, these variables must be defined to cope with technical specifications, including displacements and vibration damping.
As a first step, hanger parameters related to material and shape are determined in case of static non linear responses (displacements, stress, strain and constraints reaction forces in the presented application).
In the present approach, the NN training data has been obtained by FEA simulations. Exhaust system consists of two main parts: the exhaust, whose behaviors are lineal, and rubber hangers, which introduce non linearity at the system. In order to model rubber hangers non linearity,a packet of three coaxial springs is considered, each one caracterized by the following parameters:

- linear stiffness,
- offset between spring and point of force application,
- preload.

The packet's behavior remains reflected in a non linear Force-Displacement curve, also called non linear stiffness curve. The offset and preload parameters permit to define different non linear stiffness curves.
In this paper two non linear stiffness curves have been considered, see Fig. 1, according to the parameters values showed in Table 1. Each non linear stiffness curve is obtained with the formula

$$\text{Force} = \text{Stiffness} \cdot (\text{Displacement} + \text{Preload} - \text{Offset}). \tag{1}$$

Table 1. Springs parameters

Spring	*Stiffness* (N/mm)	*Offset* (mm)		*Preload* (mm)	
		Case 1	Case 2	Case 1	Case 2
1	12	0	0	0	0
2	25	4	8	5	5
3	38	6	10	5	5

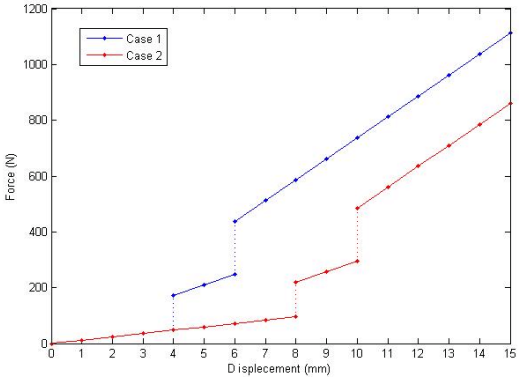


Fig. 1. Two non linear stiffness curves

As mentioned above, the exhaust system vibration might cause discomfort. For this reason vertical load at strategic points are considered. Besides, boundary conditions are selected to make the system sensitive to the applied forces: at the entry of the exhaust pipe, movements are restricted by a cylindrical hinge with rotation axis along Y, so that displacement in the three directions and rotation in the X and Z direction are zero; the free end (opposite to that is attached to the exhaust pipe) of the hangers is fitted, so that displacement and rotation in all directions are zero.

The exhaust pipe and the hangers are modeled using linear SHELL QUAD4 and linear CBAR 1D BEAM type elements respectively, as showed in Fig. 2. The element size objective is 4 mm. These mesh options define good quality models for the analysis included in the project.

To assure accuracy of the results of the NN, the experimental validation of the FE model becomes necessary. As previous step, upper and lower Force-Displacemente curves were definined, as well as maximum and minimum applied forces. Thus, several specimens' tests permitted to corroborate the model under intermediate and extreme parameters values.

To make the NN as efficient as possible DOE techniques are used for a proper distribution of study cases. Full factorial design includes one of possible combination that takes into account the influence of all variables in model response. Training data are built considering the stiffness of rubber hangers described by

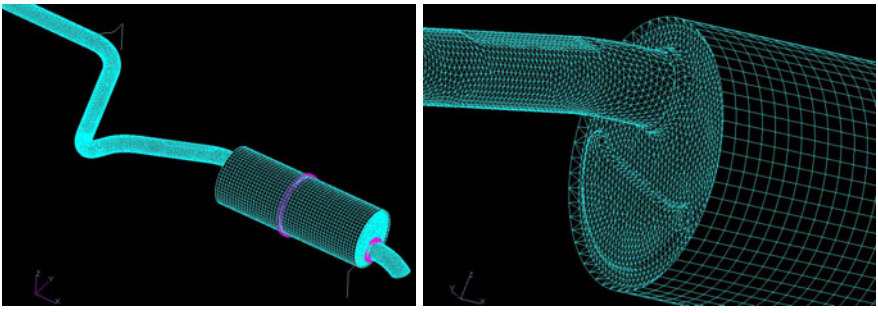


Fig. 2. FEM model details

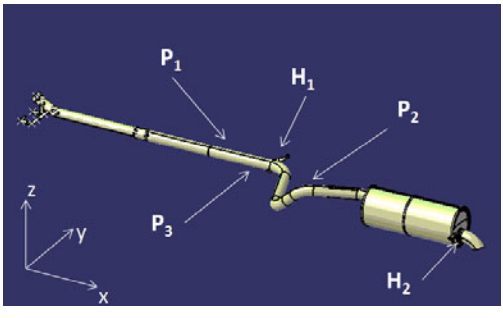


Fig. 3. Applied forces location (P_1 and P_2), rubber hangers position (H_1 and H_2) and point location for comparison results (P_3)

the same curve or by the two different curves and the load cases varying into the range $[0, 500]$ N. In this case the input layer is defined by six neurons (non linear stiffness of rubber hangers (H_1 and H_2) characterized by two parameters that define force-displacement curves for each of them; applied forces at two different points of the exhaust (P_1 and P_2), Fig. 3). The output layer, on the other hand, is determined by ten neurons (constraints reaction forces in hangers (H_1 and H_2); displacements (in Z direction) of hangers (H_1 and H_2); maximum local stress, strain and displacements at a point (P_3), Fig. 3).

Four hidden layers networks are considered. The maximum number of neurons is 60. The network population is trained with the Levenberg-Marquardt algorithm [17].

The resulting structure is $(10, 10, 10, 10)$ network. Table 3 shows testing errors comparison obtained with FEA and network analysis.

The comparison is done taking into account stress, σ_x, and displacement, d_z, values at the point $P3$. For all cases the relative error is below 5%, so that the evolutionary NN is able to predict the exhaust system response. Furthermore the computational time is strongly reduced: FEA needs 1.25 hours to solve each case, while the designed NN can simulate the same model in just two seconds.

Table 2. FEA and NN analysis results comparison

TEST DATA	*INPUTS*				*FEA*		*NN*		*Δ(FEA-NN)*	
N°	H1 − 2 points (mm)	H2 − 2 points (mm)	F1 (N)	F2 (N)	d_z (mm)	σ_x (MPa)	d_z (mm)	σ_x (MPa)	d_z (%)	σ_x (%)
1	8, 10	8 , 10	45.00	180.00	9.74	39.36	10.06	40.96	-3.29	-4.08
2	8, 10	8 , 10	67.50	270.00	11	50.9	11.33	52.65	-3.00	-3.42
3	8, 10	8 , 10	90.00	360.00	12.23	61.02	12.56	62.80	-2.70	-2.92
4	8, 10	8 , 10	112.50	450.00	13.57	68.22	13.90	70.08	-2.43	-2.72
5	8, 10	4 , 6	45.00	180.00	9.42	46.72	9.72	48.24	-3.18	-3.25
6	8, 10	4 , 6	67.50	270.00	10.69	58.53	11.06	60.29	-3.46	-3.01
7	8, 10	4 , 6	90.00	360.00	11.99	67.37	12.37	69.28	-3.17	-2.84
8	8, 10	4 , 6	112.50	450.00	13.18	76.42	13.55	78.25	-2.81	-2.39
9	4, 6	8 , 10	45.00	180.00	6.71	25.59	6.56	24.97	2.24	2.42
10	4 , 6	8 , 10	67.50	270.00	7.95	36.61	7.81	35.95	1.76	1.82
11	4, 6	8 , 10	90.00	360.00	9.51	47.75	9.36	47.14	1.58	1.27
12	4 , 6	8 , 10	112.50	450.00	10.96	55.54	10.80	54.75	1.46	1.43
13	4 , 6	4 , 6	45.00	180.00	6.39	33.09	6.30	32.55	1.41	1.63
14	4, 6	4 , 6	67.50	270.00	7.65	44.43	7.54	43.90	1.44	1.19
15	4 , 6	4 , 6	90.00	360.00	9.10	53.75	8.99	53.24	1.21	0.95
16	4, 6	4 , 6	112.50	450.00	10.70	63.93	10.58	63.21	1.12	1.13

4 Conclusions

A new virtual engineering tool based on integration of NN and GA has been investigatedwith the objective of reducing physical tests on prototypes, helping designers to select the best design alternative within a short time.
In this paper the method is applied to adjust design parameters of an exhaust system responsible of a certain non linera static response.
Benefits of the proposed methodology depend on each specific case, but generally they can be measured in terms of:

- Calculation time: Very short (seconds) compared to the development of new FEA analyses for each study (hours).
- Reliability: Results from the NN program are generally within a 5% error range compared to FEA analyses.
- Availability: Program can be used by anyone in the company. No specific knowledge or training on FEA is required.
- Knowledge: The program can be used as a knowledge repository where one can look for the best design.
- Cost: The number of prototypes and tests can be reduced by analyzing in advance the NN ouputs and selecting the right design.

In this approach GA works with NN having same number of hidden layers. As a continuation of this work, it is intended to consider a NN family with different number of hidden layers. It would also be interesting to examine how the methodology performs with non linear dynamic structural analysis.

References

1. Parra, C., Asensio, J., Olazagoitia, J.L., Biera, J.: Development of intelligent tools to eliminate squeal noise in brake systems. In: SAE Internacional, Phoenix, USA (2010)

2. Parra, C., Olazagoitia, J.L., Biera, J.: Practical Tool for the Design of Brake Pads to Avoid Squeal Noise in Automotive Brake Systems. In: 6th European Conference on Braking – JEF 2010, Lille, France (2010)
3. Haykin, S.: Neural Networks. A Comprehensive Foundation. Pretince Hall, New York (1999)
4. Haykin, S.: Neural Networks and Learning Machines. Pretince Hall, New York (2009)
5. Lawrence, S., Giles, C., Tsoi, A.: What size neural network gives optimal generalization? convergence properties of backpropagation. University of Maryland Technical Report, UMIACS-TR-96-22 (1996)
6. Panchal, G., Ganata, A., Kosta, Y.P., Panchal, D.: Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers. International Jounal of Computer Theory and Engineering 3(2) (2011)
7. Fiszelew, A., Britos, P., Perichisky, G., García-Martínez, R.: Automatic Generation of Neural Networks Based on Neural Networks, PhD Thesis (2002)
8. Ileană, I., Rotar, C., Incze, A.: The Optimization of Feed forward Neural Networks structure Using Genetic Algorithms. In: International Conference on Theory and Applications of Mathematics and Informatics, Thessaloniki (2004)
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company Inc. (1989)
10. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press (1996)
11. Jenkins, W.M.: Towards structural optimization via genetic algorithm. Computers and Structures 40(5), 1321–1327 (1991)
12. Hajela, P.: Genetic search - approach to the non-convex optimization problem. Journal of AIAA 28(7), 1205–1210 (1990)
13. Rao, S.S.: Optimal placement of actuation in actively controlled structures using genetic algorithms. Journal of AIAA 29(6), 942–943 (1991)
14. Deb, K.: Optimal design of a welded beam via genetic algorithms. Journal of AIAA 29(11), 2013–2015 (1991)
15. Hajela, P., Lin, Y.: Genetic search strategies in multi-criterion optimal design. AIAA-1040-CP, 354–363 (1991)
16. Stone, M.: Cross validatory choise and assessment of statistical preditions. J. R. Stat. Soc. Ser. B (Methodological) 36, 111–147 (1974)
17. Levenberg, K.: A Method for the Solution of Certain Non-Linear Problmes in Least Square. The Quarterly of Applied Mathematics 2, 164–168 (1944)

An Incremental Hypersphere Learning Framework for Protein Membership Prediction

Noel Lopes[1,4], Daniel Correia[1,3], Carlos Pereira[1,3], Bernardete Ribeiro[1,2], and António Dourado[1,2]

[1] CISUC - Center for Informatics and Systems of University of Coimbra, Portugal
[2] Department of Informatics Engineering, University of Coimbra, Portugal
[3] ISEC - Coimbra Institute of Engineering, Portugal
[4] UDI/IPG - Research Unit, Polytechnic Institute of Guarda, Portugal
noel@ipg.pt, cpereira@isec.pt, {djcorreia,bribeiro,dourado}@dei.uc.pt

Abstract. With the recent raise of fast-growing biological databases, it is essential to develop efficient incremental learning algorithms able to extract information efficiently, in particular for constructing protein prediction models. Traditional inference inductive learning models such as SVM perform well when all the data is available. However, they are not suited to cope with the dynamic change of the databases. Recently, a new Incremental Hypersphere Classifier (IHC) Algorithm which performs instance selection has been proved to have impact in online learning settings. In this paper we propose a two-step approach which firstly uses IHC for selecting a reduced data set (and also for immediate prediction), and secondly applies Support Vector Machines (SVM) for protein detection. By retaining the samples that play the most significant role in the construction of the decision surface while removing those that have less or no impact in the model, IHC can be used to efficiently select a reduced data set. Under some conditions, our proposed IHC-SVM approach is able to improve performance accuracy over the baseline SVM for the problem of peptidase detection.

Keywords: Incremental Learning, Support Vector Machines, Incremental Hypersphere Classifier, Protein Membership Classification.

1 Introduction

The study of proteins plays a prominent role in understanding many biological systems. In particular, predicting protein membership based on unidentified sequences is an actual and relevant task for which huge amounts of data already exist. Batch learning algorithms are unable cope with large scale datasets that are becoming pervasive in many domains. Specifically, the growing rate of biological databases demands for incremental learning algorithms that can quickly update their models to incorporate new information.

The classification of protein sequences into functional and structural groups based on sequence similarity is a contemporary and relevant issue in the bioinformatics domain [7]. However, despite all the energy spent into deciphering the

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 429–439, 2012.

proteomes the available knowledge is still limited. Thus, our focus consists of extracting relevant features from the proteins primary structure, and to design a protein prediction membership model able to face the everyday challenges.

In this context, we propose to use a new incremental algorithm Incremental Hypersphere Classifier (IHC) as the first step for learning new incremental data, and then to use Support Vector Machines (SVM) for final protein classification. The easiness of the proposed approach IHC-SVM will be demonstrated throughout the paper. The IHC is an instance based algorithm that retains the samples that play the most significant role in the construction of the decision surface (given the available memory) while removing those that have less or no impact in the model. A major advantage of this algorithm relies on the possibility of building models incrementally on a sample-by-sample basis while accommodating restrictions in terms of memory and computational power. Moreover the interpretability of the resulting models makes IHC particularly suited for handling the problem of protein classification.

The remainder of this paper is organized as follows. The next section describes the related work. Section 3 describes the Incremental Hypersphere Classifier (IHC) algorithm, and presents the proposed learning framework approach IHC-SVM. Section 4 presents the datasets, explains the preprocessing steps, describes the evaluation metrics, and discusses the results. Finally, section 5 gives the conclusions and points out future lines of work.

2 Related Work

Peptidases are proteolytic enzymes that catalyze chemical reactions, allowing the decomposition of protein substances into smaller molecules. They are involved in several processes crucial for the correct functioning of organisms. Their importance is proved by the fact approximately 2% of all genes encode peptidases and their homologues in all kinds of organisms [11].

The Support Vector Machine (SVM) is an approaches from the machine learning techniques based on the statistical learning theory [13]. This kernel formulation has been highly successful in solving both classification and regression problems, building discriminative models that combine high accuracy with good generalization qualities. Because of this, SVM have been widely applied in (static) classification of biological data including sub-sequence cellular prediction or protein sequence classification [4,10,12].

Due to the ever increasing biological databases a fast response for protein classification prompts the need for algorithms able to handle incremental learning. There are two main approaches for incremental learning [6]. One approach consists of using a single model that is dynamically updated as new data becomes available. The other is a hybrid batch-incremental approach, that relies on several models built using batch learning techniques. New models are continuously being created by using only a small subset of the data. As more reliable models become available they replace the older and obsolete ones. Some of the hybrid approaches rely on a single model while others use an ensemble of models [6]. In

the next section we describe a new incremental learning algoritm (IHC) which offers advantages to the problem of peptidases classification, in particular, when combined with SVM.

3 Incremental Learning Framework

3.1 IHC algorithm

The Incremental Hypersphere Classifier (IHC) is a new incremental instance-based learning algorithm which exhibits advantages in terms of multi-class support, complexity, scalability and interpretability while providing good classification results [5]. Basically, it assigns a region of influence to each sample, by which classification is achieved. Let us consider that a training sample i consists of an input vector $\mathbf{x}_i \in \mathbb{R}^d$ with an associated class label $y_i \in \{1, \ldots, C\}$, where d is the space dimension and C is the number of classes. The region of influence of sample i is then defined by an hypersphere of radius r_i, given by (1):

$$r_i = \frac{\min(||\mathbf{x}_i - \mathbf{x}_j||)}{2}, \text{ for all } j \text{ where } y_j \neq y_i \; . \tag{1}$$

Each hypersphere will occupy as much space as possible provided that hypersphere's belonging to different classes do not overlap. New data points are classified according to the class of the nearest region of influence (not the nearest sample). Let $\mathbf{x}_k$ represent an input vector whose class y_k is unknown. Then, sample k belongs to class y_i ($y_k = y_i$) provided that:

$$||\mathbf{x}_i - \mathbf{x}_k|| - g a_i r_i \leq ||\mathbf{x}_j - \mathbf{x}_k|| - g a_j r_j, \text{ for all } j \neq i \; . \tag{2}$$

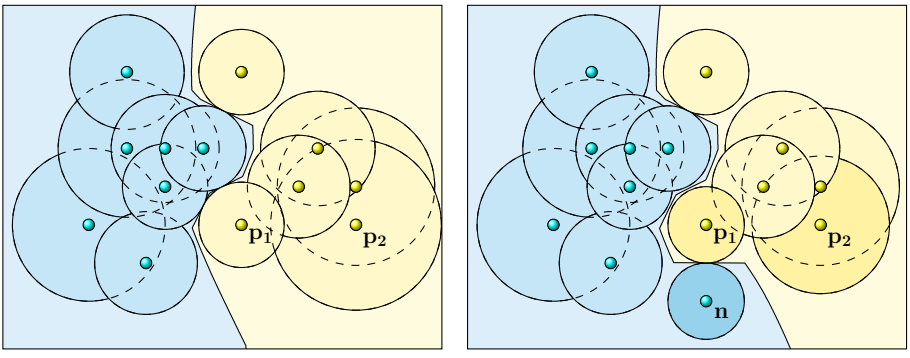
where g (gravity) controls the extension of the zones of influence, increasing or shrinking them and a_i is the accuracy of sample i which is given by (3):

$$a_i = \frac{tp_i}{tp_i + fp_i} \; . \tag{3}$$

In the aforementioned equation, tp_i and fp_i represent respectively the number of true and false positives classified by sample i. Each sample, i, classifies exclusively itself and the forgotten training samples for which i was the nearest sample in the memory. A forgotten sample is one that has either been removed from memory or did not qualify to enter the memory. Hence, the accuracy is only updated when the memory is full. In such scenario, at each iteration, the accuracy of a single (nearest) sample is updated, while the accuracy of all the others samples remains unmodified.

The accuracy is the first mechanism of defense against outliers. As it decreases so does the influence of the hypersphere associated. This effectively reduces the damage caused by outliers and by samples with zones of influence excessively large.

Figures 1(a), 1(b) show the regions of influence and the corresponding decision surfaces generated by IHC for a chosen toy problem before and after the addition



(a) Before adding a new sample. (b) After adding a new sample.

Fig. 1. Regions of influence and decision surfaces generated by IHC for a toy problem

of a new sample, n (considering $g = 1$). Notice that adding a new sample might affect the radius of the samples already in the model (in this particular case p_1 and p_2).

The farthest from the decision border an hypersphere is, the larger its radius will be. This provides a simple method for determining the relevance that a sample has in the classification task: samples with smaller radius (r_i) are more important to the classification task than those with bigger radius. When the memory is full, the radius of a new sample is compared with the radius of the nearest sample of the same class and the one with the smallest radius is kept in the memory while the other is discarded. By doing so, we are keeping the samples that play the most significant role in the construction of the decision surface (given the available memory) while removing those that have less or no impact in the model.

Unfortunately, outliers will most likely have small radius and end-up occupying our limited memory resources. Thus, although their impact is diminished by the use of the accuracy in (2), it is still important to identify and remove them from memory. To address this problem we mimic the process used by the IB3 (Instance-Based learning) algorithm, which consists of removing all samples that are believed to be noisy by employing a significance test [14,1]. Confidence intervals are determined both for the instance accuracy (not including its own classification, unlike in eq. (2)) and for the relative frequency of the classes [1]. The instances whose maximum (interval) accuracy is less than the minimum class frequency (for the desired confidence level – typically 70%) are considered outliers and consequently dropped off.

To cope with unbalanced datasets and avoid storing a disproportionate number of samples for each class, the algorithm assumes that the memory is divided into C groups.

3.2 Proposed IHC-SVM Learning Framework

We propose a two-step learning approach which firstly uses IHC for selecting a reduced data set and secondly applies Support Vector Machines (SVM) for protein detection. By retaining the samples that play the most significant role in the construction of the decision surface while removing those that have less or no impact in the model IHC can be used to efficiently select a reduced dataset.

Our IHC-SVM learning framework works as follows: as new data becomes available IHC is used to create a new model or to update an existing one. As mentioned before, this can be accomplished incrementally on a sample by sample basis. Since IHC is an instance-based algorithm it could also be used as an instance selection algorithm. Thus, we can periodically check if the IHC model has changed and use the samples that constitute the IHC model to create more robust models. To this end, we can use state of the art batch algorithms, such as the Support Vector Machines (SVM) algorithm, that otherwise could not be applied to all the data, due to processing power and memory constraints. Figure 2 depicts the resulting learning framework.

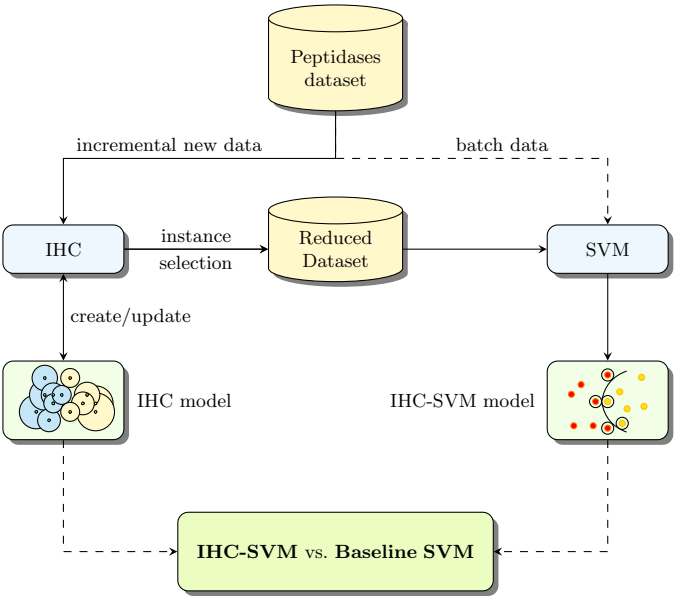


Fig. 2. Proposed Learning Framework

By combining incremental and batch algorithms in the same framework, we expect to obtain the benefits of both approaches while minimizing their disadvantages. Namely, we expect the proposed framework to be able to cope with extremely large, fast changing, datasets while retaining state of the art classification performance.

4 Experimental Setup

In this section we start by describing the problem of protein membership prediction (see Section 4.1). The data pre-processing is then covered in Section 4.2. Section 4.3 presents the evaluation metrics. Section 4.4 analysis and discusses the results.

4.1 Experimental Datasets

Peptidases are a class of enzymes that catalyze chemical reactions, allowing the decomposition of protein substances into smaller molecules. They are involved in several processes crucial for the correct functioning of organisms. Its detection is central to a better understand of their role in a biological system. For the purpose of peptidase detection a benchmark dataset was constructed using the MEROPS [11] and the SCOP [8] databases. From those databases, 20778 proteins sequences were collected, 18068 positive samples from MEROPS 9.4 and 2710 sequences non peptidase from SCOP 1.75, both being randomly selected. This benchmark dataset was previously used and described in Pereira et al. [9].

The dataset has been divided in two groups, 17164 sequences for training (15358 positive and 1806 negative examples) and 3614 sequences for testing purposes (2710 positive examples and 904 negative). The sequences were randomly selected from the complete dataset. We specifically choose a relatively small dataset, so that we could optimize the baseline SVM algorithm parameters in order to guarantee the validity of the comparisons between the proposed approach and the baseline.

4.2 Pre-processing

The features of the protein primary structure were extracted from the dataset using text mining techniques [3]. The idea behind is to split the continuous flow of amino acids in substrings (n-grams) of length (n) – The n-grams are formed by n consecutive characters and each one corresponds to a feature in a particular sequence. As an example, considering the partial sequence 'PKIYGY', the trigrams would be 'PKI', 'KIY', 'IYG' and 'YGY'.

The unigrams, bigrams, trigrams and the combinations of those n-grams have been considered in this study for feature selection. In order to implement the extraction of those features the WVTool library [15] has been used. This Java library provides a simple and flexible implementation to create a vector representation. That representation is formed by a two dimensional vector where each line corresponds to a given protein sequence and each column represents an n-gram. In our case, we used three steps of the WVTool vectorization process, (i) the processing step in order to load and process the local files with the protein sequences, (ii) the tokenizer for n-gram extraction and the step for create the word vector and (iii) compute the relevance of each n-gram.

There are quite a lot of methods for compute the relevance of each feature, such as binary occurrence, term occurrences, term frequency and more. In this work, the chosen metric was the term frequency (tf) defined as (4):

$$tf_{ij} = \frac{f_{i,j}}{fd_j} \tag{4}$$

where $f_{i,j}$ is the number of occurrences of feature i in the sequence j and fd_j is the sum of the number of occurrences of all the features in sequence j.

4.3 Evaluation Metrics

For evaluating the performance of the classifiers and asserting its quality, we use the F-measure score (expressed as percentage). The F-measure is based on the precision and recall, which in turn are based on a confusion matrix. A confusion matrix contains the number of correctly and incorrectly classified examples for each class.

The precision and recall, are given respectively by (5) and (6):

$$precision = \frac{tp}{tp + fp} \,. \tag{5}$$

$$recall = \frac{tp}{tp + fn} \,. \tag{6}$$

A classifier presenting a high precision is rarely wrong when it predicts that a sample belongs to the class of interest (positive). While a classifier presenting a high recall rarely misclassifies a sample that belongs to the class of interest. Usually there is a trade-off between precision and accuracy and although there are cases where it is important to favor one in detriment of the other, for most problems it is important to balance and maximize both. This can be accomplished by maximizing the F-measure score, given by (7):

$$F\text{-}measure = 2 \times \frac{precision \times recall}{precision + recall} \,. \tag{7}$$

4.4 Results and Discussion

To demonstrate the validity of the proposed approach we present the results which were obtained by performing the experiments using a Core 2 Quad Q 9300 2.5 GHz CPU. Moreover, in our study the well-known LIBSVM [2] version 3.1 has been used, since this library implements various SVM formulations for classification, regression and distribution estimation.

In Lopes and Ribeiro [5] we have shown that the complexity of IHC is $O(2dN)$. Thus, we can control the computational power that IHC requires, simply by adjusting the amount of memory (N) that the algorithm is allowed to use. Figure 3 shows the time required to update the IHC model, after the number of samples in the memory is stabilized. Prior to that, the time required is much smaller. Note that the number of samples actually stored is inferior to N because the training dataset is strongly unbalanced. Since IHC divides the available memory by C classes (see Section 3.1), the number of samples stored in memory for the

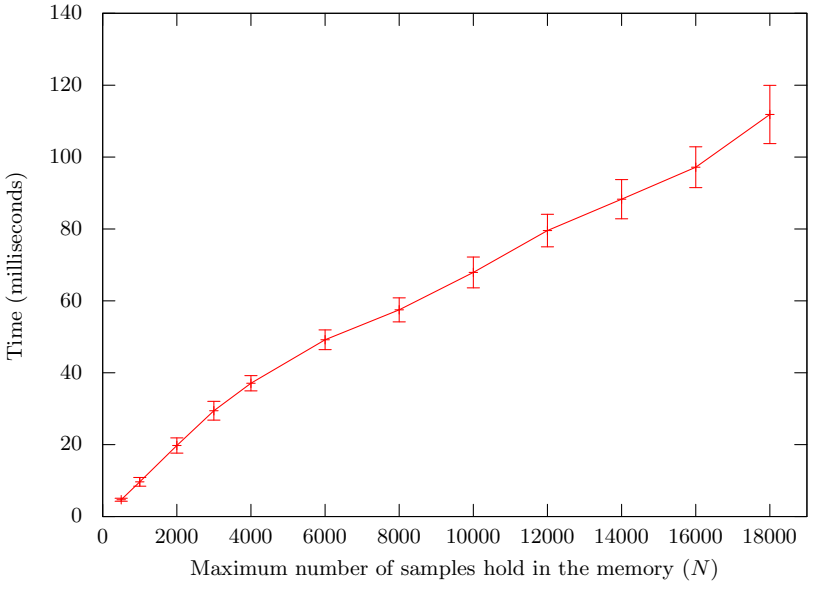


Fig. 3. Average time required to update the IHC model (with a new sample)

non peptidase class will be at most 1806. Thus, for $N = 20000$ the actual number of samples stored will never exceed 11806.

To define a baseline of comparison, we started by computing the performance of the SVM algorithm which is a state of the art batch algorithm for classification. For this purpose, several kernels and parameters (using grid search) were tried using 5-fold cross validation on the training dataset in order to determine the best possible configuration. Specifically we found the best configuration to use an RBF (Gaussian Radial Basis Function) kernel with parameters $\gamma = 0.4$ and $C = 100$. Adopting the specified configuration we obtained macro-average F-measure for the test dataset of 95.91%. The same configurations were used to train the SVMs in the proposed (IHC-SVM) approach.

We tested both the IHC and the IHC-SVM approach for the concerned problem, using parameters $g = 1$ and $g = 2$ which have demonstrated to yield good results in Lopes and Ribeiro [5]. Based on the information collected, we set $g = 2$. Figure 4 show the macro-average F-measure for both the IHC algorithm and for IHC-SVM approach, using the specified parameter.

Incremental algorithms, such as the IHC have no access to previously presented data and no control on the order in which the data is presented. With this constraints, we cannot expect their performance to match those of batch algorithms. Despite that, the IHC is able to achieve an F-measure of 93.73%. A working version of the IHC algorithm for peptidase detection can be found in `http://193.137.78.18/ihc/`.

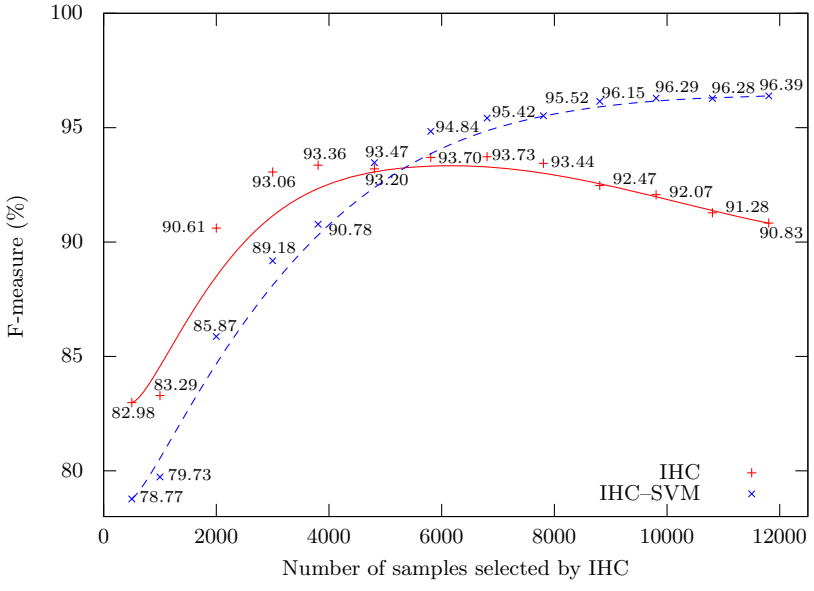


Fig. 4. IHC and IHC-SVM F-measure (macro-average) performance

Notice that when the number of samples stored (tied directly to N) is too small, the resulting models will be unable to capture the underling distribution of the data (under-fitting), since we will only be able to store a fraction of the samples that define the decision frontier (see Figure 4). In such situation the IHC algorithm performs better than the IHC-SVM approach. This might seem strange at first, however the explanation is quite simple: while the SVM algorithm has only access to the instances selected by IHC, the latter had access to the whole dataset (although in a sample by sample basis and not in the desired (optimal) order) and thus it able to use the additional information to construct better models. As N grows, the number of stored samples gets larger and as a result, the number of forgotten samples declines. Since these are essential to reduce the damage caused by outliers and by samples with zones of influence excessively large we will end-up with over-fitting models, concerning the IHC algorithm. However, the IHC-SVM approach is not affected in the same manner, since the SVM algorithm is able to create better models with the additional data supplied by IHC. In fact, in this situation the proposed approach (IHC-SVM) even works better than the baseline batch SVM. This provides evidence that the process used by IHC to determine the relevance of each sample and decide which ones to retain and which ones to discard is efficient.

Table 1 shows the gains of the proposed incremental-batch approach (IHC-SVM) over the baseline batch approach (SVM). Note that, the IHC-SVM approach is able to excel the baseline (SVM) approach using only a subset of the data. With roughly 50% of the original data (8806 samples out of 17164) it is

Table 1. IHC-SVM compression ratio and classification improvement over the baseline (SVM).

Number of samples selected by IHC	Compression ratio (%)	F-Measure improvement (%)
500	97.09	↓ 17.87
1000	94.17	↓ 16.87
2000	88.35	↓ 10.47
3000	82.52	↓ 7.02
3806	77.83	↓ 5.35
4806	72.00	↓ 2.54
5806	66.17	↓ 1.12
6806	60.35	↓ 0.51
7806	54.52	↓ 0.41
8806	48.69	↑ **0.25**
9806	42.87	↑ **0.40**
10806	37.04	↑ **0.39**
11806	31.22	↑ **0.50**

possible to create improved models. Moreover, it is possible to compact the data even further and still obtain models that match closely the performance of the baseline model.

5 Conclusions and Future Work

We presented a learning framework approach (IHC-SVM) for predicting protein membership which is able to deal with the dynamic everyday changes of biological databases. It has been demonstrated that under certain conditions the IHC-SVM presents better performance than SVM baseline using sequences of proteins built from well-known peptidase repositories. There is evidence that by using IHC as the first step to determine the relevance of each sample and decide which ones to retain and which ones to discard is an efficient procedural for the incremental learning framework. The SVM training on the reduced data set builds a model that yields high accuracy which is desirable to handle the dynamic (and pervasive) biological databases.

A major advantage of this algorithm relies on the possibility of building models incrementally on a sample-by-sample basis while accommodating restrictions in terms of memory and computational power.

Future work will extend this study to further experiments involving larger dynamic datasets as well as comparing the proposed approach with other incremental and batch classifiers.

Acknowledgment. This work was supported by FCT (Foundation for Science and Technology) and FEDER through Program COMPETE (QREN) under the project FCOMP-01-0124-FEDER-010160 (PTDC/EIA/71770/2006), designated BIOINK - Incremental Kernel Learning for Biological Data Analysis.

References

1. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)
2. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 1–27 (2001)
3. Cheng, B.Y., Carbonell, J.G., Klein-Seetharaman, J.: Protein classification based on text document classification techniques. Proteins: Structure, Function, and Bioinformatics 58(4), 955–970 (2005)
4. Hua, S., Sun, Z.: Support vector machine approach for protein subcellular localization prediction. Bioinformatics 17(1), 721–728 (2001)
5. Lopes, N., Ribeiro, B.: An Incremental Class Boundary Preserving Hypersphere Classifier. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part II. LNCS, vol. 7063, pp. 690–699. Springer, Heidelberg (2011)
6. Masud, M.M., Chen, Q., Khan, L., Aggarwal, C., Gao, J., Han, J., Thuraisingham, B.: Addressing concept-evolution in concept-drifting data streams. In: Proceedings of the 2010 IEEE International Conference on Data Mining, pp. 929–934. IEEE Computer Society Press, Washington, DC (2010)
7. Morgado, L., Pereira, C., Veríssimo, P., Dourado, A.: A support vector machine based framework for protein membership prediction. In: Computational Intelligence for Engineering Systems, Intelligent Systems, Control and Automation: Science and Engineering, vol. 46, pp. 90–103. Springer, Netherlands (2011)
8. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of molecular biology 247(4), 536–540 (1995)
9. Pereira, C., Morgado, L., Correia, D., Verissimo, P., Dourado, A.: Kernel machines for proteomics data analysis: Algorithms and tools. Presented at the European Network for Business and Industrial Statistics, Coimbra, Portugal (2011)
10. Ratsch, G., Sonnenburg, S., Schafer, C.: Learning interpretable svms for biological sequence classification. BMC Bioinformatics 7, S1–S9 (2006)
11. Rawlings, N.D., Barrett, A.J., Bateman, A.: MEROPS: the peptidase database. Nucleic Acids Research 38(Database-Issue), 227–233 (2010)
12. She, R., Chen, F., Wang, K., Ester, M., Gardy, J.L., Brinkman, F.S.L.: Frequent-subsequence-based prediction of outer membrane proteins. In: Proceedings of Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA (2003)
13. Vapnik, V.N.: The nature of statistical learning theory. Springer, Heidelberg (1995)
14. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)
15. Wurst, M.: The word vector tool user guide operator reference developer tutorial (2007)

An Evolutionary Approach to Generate Solutions for Conflict Scenarios

Davide Carneiro, Cesar Analide, Paulo Novais, and José Neves

Departamento de Informática, Universidade do Minho, Campus de Gualtar, Braga, Portugal
{dcarneiro,analide,pjon,jneves}@di.uminho.pt

Abstract. Conflict resolution is nowadays an important topic.Online Dispute Resolution in particular is nowadays a major research topic, focusing onthe development of technology-based tools to assist parties involved in conflict resolution processes. In this paper we present such a tool aimed at the generation of solutions. It is based on Genetic Algorithms that evolve a population of solutions through successive iterations, generating more specialized ones. The result is a tree of solutions that the conflict resolution platform can use to guide the conflict resolution process. This approach is especially suited for parties which have no ability or are unwilling to generate realistic proposals for the resolution of the conflict.

Keywords: Online Dispute Resolution, Genetic Algorithms, Distributive Negotiation.

1 Introduction

Given the current state of courts, new approaches on conflict resolution are needed. Specifically, courts are nowadays unable to deal with the amount and characteristics of new disputes. In fact, while in the past conflicts emerged between persons generally in the geographical vicinity of each other, nowadays a conflict may emerge between two persons, regardless their location, and it may even involve software agents. This new modality of contracting, the so-called electronic contracting, is in fact one of the biggest challenges for current legal systems, relying in paper-based courts still shaped after the industrial revolution [1]. In that sense, alternatives to litigation are needed.

The first ones involved parties trying to solve their differences without recourse to litigation, generally with the assistance of a third, neutral party. These processes include negotiation, mediation, arbitration or conciliation, just to name a few [2-4], and are part of the so-called Alternative Dispute Resolution [3]. However, under these traditional approaches stakeholders still have to meet in person. In that sense, Online Dispute Resolution emerged as the use of traditional conflict resolution mechanisms under virtual environments [5]. In fact, the use of technology for conflict resolution may not only be used to bring parties into contact but also to develop high value tools that can be used for the definition of strategies, for the generation of solutions or even for compiling useful information for the parties [6,7].

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 440–451, 2012.

In the last years, our research has focused on developing such tools based on Intelligent System techniques, giving birth to the UMCourt conflict resolution platform [8,9]. Specifically, we have been researching how the coming together of different fields of research can solve concrete problems in an efficient manner [13, 14], giving birth to the so-called Hybrid Intelligent Systems, applied to the domain of The Law in the specific case of this work. In this work we propose a module for generating solutions for a conflict resolution scenario based on Genetic Algorithms (GA) [10, 15]. Our line of attack targets a very specific issue in conflict resolution: the inability or unwillingness of parties to generate solutions. In fact, frequently parties find it difficult to generate realistic proposals, because they are not fully aware of what their chances are or what rules apply. In other cases, parties are simply uncooperative and do not want to bother creating solutions [11]. With this work we complement our conflict resolution platform with the ability to propose fair and realistic solutions for concrete conflict resolution scenarios.

The rest of the paper is organized as follows. In section 2 we provide a definition of the general conflict resolution scenario and how GAs can be modeled to be used as a problem solving methodology. Section 3 details the initialization of the GA while section 4 is devoted to depicting the selection process. The reproduction operators are detailed in section 5 and the termination of the algorithm is described in section 6. Finally, in section 7 we present the concluding remarks of this work.

2 Defininga Conflict Resolution Scenario with Genetic Algorithms

Even from a technological point of view, the problem of generating solutions may be a challenging task, although several different techniques can be used. One of the possible lines of attack is the use of case-based approaches, in an attempt to shape the cognitive models of human experts which rely on experience. However, this approach has some potential limitations. Specifically, it is likely to fail in scenarios in which case-bases with insufficient cases are used. Moreover, bigger case-bases ensure more completeness but generally also result in slower processes. In this paper we propose an approach that is independent of these constraints: the use of Genetic Algorithmsto create solutions for conflict resolution scenarios.

Under GA approaches, a solution for a problem is represented by a chromosome. In that sense, given the domain of application of this work, each chromosome represents a solution for a specific conflict resolution problem, generally a distribution of the items being disputed among several parties. The population of chromosomes evolves from generation to generation through the application of genetic operators that act on the distribution, thus changing its fitness. This approach has also a specificity considering fitness. Usually, in a GA problem, fitness is seen as an absolute value. In this context, a solution has several values of fitness, one for each party, i.e., a solution that is good for a given party is most likely not that good for any other given that they tend to have conflicting objectives. As the generations of solutions succeed, there are lines of evolution of chromosomes that tend to be more fit

to a given party. These lines emerge naturally and are called species. A species is thus defined as a group of chromosomes from the same line of evolution whose fitness is better for a specific party. In that sense, each party has a species of solutions. A chromosome may also belong to more than one species if it has satisfactory values of fitness for more than one party. These chromosomes are evidently more attractive since they correspond to solutions that will be more easily accepted by the parties.

A population P of size s is defined by a set of chromosomes Ch (Figure 1), in which each chromosome $Ch_i, i \in \{1,2,\dots,s\}$represents a possible solution for the problem, i.e., who gets how much of what. Considering a dispute involving n parties and m issues, a chromosome Ch can be represented as an m-by-n matrix (equation 1).

$$Ch = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} \tag{1}$$

$$P = \left| Ch_1 = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} \right| Ch_2 = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} \Big| \dots \Big| Ch_s = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} \Big|$$

Fig. 1. Under this model a population of size s is represented as a set of chromosomes with a cardinality of s

Under this representation, the value $V_{m,n}$of the chromosome Chrepresents the amount of issue m that the party n receives in this specific solution. Evidently, the actual content of the chromosome depends on the domain of the dispute. Likewise, domain-dependent rules must be defined that enforce the correctness of the solutions generated. Let us take as example the general model of distributive negotiation. Under this model, there is a set of items that must be distributed by a number of parties. Traditional scenarios include divorces or winding up of companies. Under this model each entry in the matrix contains a value between 0 and 1 (equation 2), and the sum of the values of each line must at all times be 1 (equation 3). The total amount of resources received by party n, R_n, is defined as the sum of the values of column n (equation 4).

$$V_{m,n} \in A,\ A = \{x \in \mathbb{R} | \ 0 \leq x \leq 1\} \tag{2}$$

$$\sum_{i=1}^{n} V_{m,n} = 1\,, \forall\, m \in \{1,2,\dots,m\} \tag{3}$$

$$R_n = \sum_{i=1}^{m} V_{m,n} \tag{4}$$

In the development of this model we also take into consideration the possible existence of indivisible items, i.e., items that due to its characteristics or due to a decision of the parties cannot be divided (e.g. many parties do not agree on selling the item to split the value). For each indivisible item m, equation 5 applies.

$$V_{m,i} = 1 \Rightarrow V_{m,x} = 0, \forall\, x \in \{1, 2, \ldots, n\}, x \neq i \quad (5)$$

More specific domains may require the definition of additional rules. This allows this model to be applied to virtually every legal domain. Let us take as example the labor law domain. Under this domain, the items in dispute may be of very different nature, ranging from monetary compensations to the possibility or not of being fired. Considering the Portuguese context, a worker being fired without a just cause is entitled to a compensation that ranges from 15 to 45 days of wage for each year of antiquity. This is generally one of the items being distributed (e.g. in a scenario in which the employee receives 30 days of wage for each year of antiquity, the other 15 days are received by the employer). Other issues generally include night or extra hours, the existence or not of complaints against the employee, among others. All these issues may be modeled in this generic model.

In a general way, the GA lifecycle follows the model depicted in Figure2: it starts with an initialization of a population, usually in a random way. Afterwards, a cycle repeats until an ending condition is met: the fitness of the population is evaluated and then the population evolves through the application of genetic operators that create new populations with different characteristics. In each iteration, the fittest of the population are selected, thus evolving the population. This is the general model of the algorithm presented in this paper, detailed in the following sections.

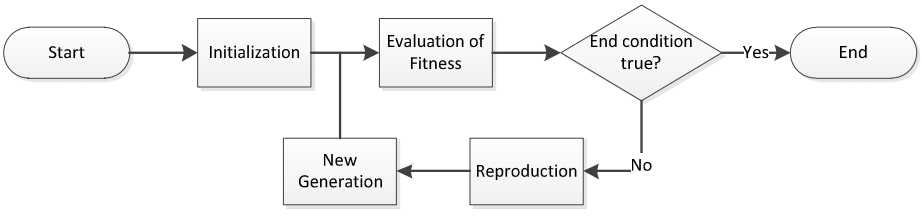


Fig. 2. Generic model of a Genetic Algorithm

3 Initialization

The initialization is the first step in the use of a Genetic Algorithm, in which the key characteristics of the population and of the individuals are defined. Figure 3 depicts the interface that allows the initialization of the algorithm. In terms of the GA, it is necessary to provide a termination condition, in terms of a maximum number of rounds, and the size of the population (i.e. the number of chromosomes in each generation). The interface also allows to define the number of the best individuals that are selected from each species to create the next generation of offspring through the application of the genetic operators. Concerning these operators, it is possible to specify the weight of each operator on the generation of new solutions. This has, evidently, a significant impact on the evolution of the population. Finally, the interface also allows configuring the weight of the components of the fitness function. Specifically, it is possible to assign the weight of the monetary value and of the

personal value. In fact, the measure of the fitness of a solution is given in terms of the monetary value of the items in dispute but also in terms of the personal value, i.e., it is also taken into consideration how much each party wants a given issue.

Concerning the specific information about the negotiation process itself, it is possible to state which are the items under negotiation (including their name, value or type) and which parties are involved. Moreover, each party must assign its preferences regarding the items in dispute. They do so by distributing a total of 100 points for all the items. This information allows the system to know how much each party wants each item and enables an estimation of the personal evaluation of the solutions.

All this information is used to initialize the algorithm. At this point, a population of the specified size is created with random solutions, i.e., each chromosome has a random distribution of items.

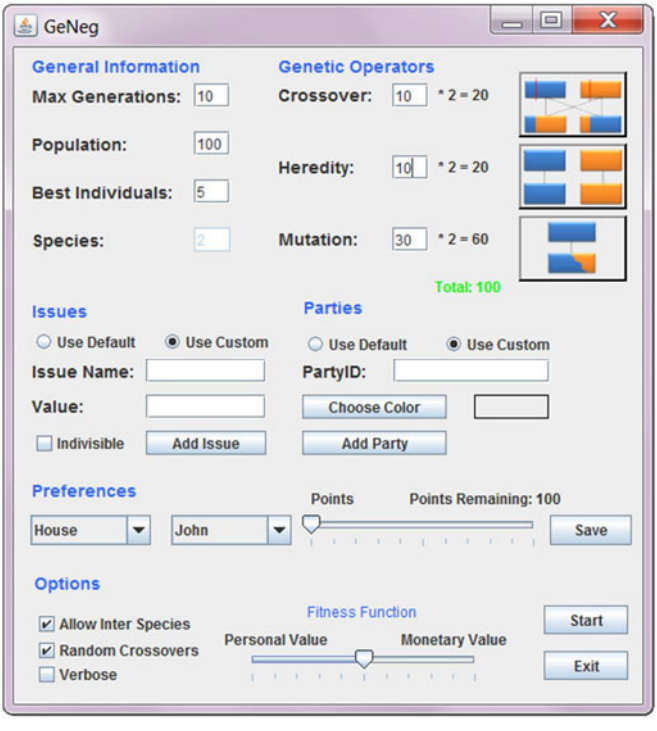


Fig. 3. The system interface used to configure the genetic algorithm, including information about the parties, the issues and the weight of each genetic operator

4 Selection

In each iteration of the algorithm a part of the population is selected to generate the following generation. This step relies on a fitness function that evaluates each individual and allows finding the best solutions. Given that solutions have different

fitness values for different parties, the fitness of each individual must be computed for each party. Thus being, for a conflict resolution involving *n* parties and for a population of size *s*, *n* * *s* values of fitness will be computed in each iteration of the algorithm.

The fitness function returns a value that is based on the portion of the items that each party receives, its monetary value and the assigned personal value. Moreover, the value of fitness also depends of the weights of the monetary and personal components, defined in the initialization. Two fitness functions were considered in this experiment (equations 6 and 7), where

- *tmv*denotes the case economic value, i.e., the total amount of money that the issues in dispute are worth with, being defined as $tmv = \sum_{i=1}^{I} mv_i$;
- *I* defines the number of issues;
- mv_i stands for the monetary value of issue *i*;
- $fit_{j,p}$ represents the fitness of chromosome *j* for party *p*;
- $W_m$denotes the weight of the monetary component while $W_p$stands fot the weight of the individual component.

When equation 6 is used as the fitness function, the solutions selected tend to be the ones in which the parties receive approximately the items that they want. That is, equation 6minimizes the difference between the personal value of the items and the points attributed to them by the parties.

$$fit_{j,p} = W_m * \frac{\sum_{i=1}^{I} Ch_{j,p} * mv_i}{tmv} + W_p * \left(1 - \sum_{i=1}^{I} \frac{|Ch_{j,p} - prefs_i|}{I}\right) \quad (6)$$

On the other hand, equation 7 tends to result in solutions in which both the monetary and the personal values are maximized. In that sense, solutions selected by the fitness function depicted in equation 6 may more likely be described as fair (as there is no blind maximization of the individual gains) while the ones by equation 7 will be more competitive and hard to be accepted by the opposing parties.

$$fit_{j,p} = W_m * \frac{\sum_{i=1}^{I} Ch_{j,p} * mv_i}{tmv} + W_p * \sum_{i=1}^{I} \frac{|Ch_{j,p} - prefs_i|}{I} \quad (7)$$

Given this, we are currently making use of equation 6 as it results in solutions that are more balanced and thus more likely to be accepted by all the parties. Given this, in each iteration of the algorithm the fittest solutions of each species are selected to give birth to anoffspring by means of genetic operators, as depicted in section 5.

5 Reproduction

The reproduction is the step in a GA in which the search heuristic moves forwards, through the engendering of new populations, towards the maximization of the fitness function. In this work, three genetic operators are being used: crossover, mutation and heredity. All of the three act on the distribution of the items, thus changing its fitness. They are applied to the selected chromosomes according to what was specified during the initialization. The operators used are defined in the following three sub-sections.

5.1 Mutation

A mutation is formally defined in genetics as a spontaneous and random change in a genomic sequence. Transposing this definition for the domain of our work, we can define mutation as a random change in the distribution of the items. The extent of the mutation is given by the mutation threshold, here designated as μ. The mutation is a unary operator that works by randomly selecting one issue and two parties from the chromosome. The distribution is then changed for the item and the parties selected according to μ. If the item is divisible, the amount of the selected item is decreased for one party and accordingly increased for the other, according to μ.On the other hand, if the item is indivisible, there is a probability given in function of μthat the owner of the item is changed between the two parties.

Whenever a new chromosome is created, its validity is checked to determine if all the invariants hold, according to rules of the type of the ones defined in section 2. Let us now consider an example scenario in which three parties are disputing four issues. Let us also assume that issue 2 is divisible and it was randomly selected to be exchanged between party 1 and party 2. The parent chromosome (*Ch*) and the offspring (*Ch'*) are depicted in equation 8.

$$Ch = \begin{bmatrix} V_{1,1} & V_{1,2} & V_{1,3} \\ V_{2,1} & V_{2,2} & V_{2,3} \\ V_{3,1} & V_{3,2} & V_{3,3} \\ V_{4,1} & V_{3,2} & V_{4,3} \end{bmatrix} \; Ch' = \begin{bmatrix} V_{1,1} & V_{1,2} & V_{1,3} \\ V_{2,1} + \mu & V_{2,2} - \mu & V_{2,3} \\ V_{3,1} & V_{3,2} & V_{3,3} \\ V_{4,1} & V_{3,2} & V_{4,3} \end{bmatrix} \tag{8}$$

After theapplication of the mutation operator the fitness of the solution for each party changes. That is, the new solution will most likely be more favorable to party1 and less favorable to party2.

Description of the Mutation algorithm.

```
Algorithm Mutation is
Input: List of parties, L
       List of issues, I
       Parent chromosome, C
Output: A new chromosome, C'
Do
i  := select random issue from I
p1 := select random party from L
p2 := select random party from L such that p1 != p2
C' := C
if (i is divisible)
      C'i,p1:= C'i,p1 + μ * C'i,p1
      C'i,p2:= C'i,p2 - μ * C'i,p2
else if (randomNumber>μ)
temp := C'i,p1
           C'i,p1:= C'i,p2
           C'i,p2:= temp
While (C' is invalid solution)
Return C'
```

5.2 Crossover

In genetics, crossover is a process by means of which a new chromosome is created using the genetic information of more than one parent solutions. In this work, crossover is a binary operator. More specifically, a two-point crossover technique is used. In this specific technique, two points are selected in the two chromosomes and all the information between those two points is swapped. In this precise context, the two points are always the beginning and the end of an issue in the matrix of distribution. Thus being, crossover consists in swapping two distributions of the same issue, generating two new solutions.

Two different approaches can be selected in the initialization form that influence the way that the crossover operator is implemented: *inter species* and *random parents*. The *inter species* option allows the system to cross chromosomes of different species. This will increase the variety of the following generation, but will most likely also delay a convergence. On the other hand, if the *inter species* option is not used, only chromosomes from the same species will be crossed. The *random parents* option tells the system about which parents to cross. If the option is used, parents are selected randomly. On the other hand, if the option is not used, the best parents from each generation are crossed. While the use of this option may increase the variety and widen the search space, it may also delay the convergence towards satisfactory solutions. In equation 9 we depict an example of the use of the crossover operator in two parent chromosomes *Ch1* and *Ch2,* to generate two offspring *Ch1'* and *Ch2'*. In this example the distribution of issue 2 was randomly selected to be swapped. Given that this technique changes the distribution of each solution, it will have effect on the fitness function.

$$Ch1 = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \\ J & K & L \end{bmatrix} Ch2 = \begin{bmatrix} M & N & O \\ P & Q & R \\ S & T & U \\ V & W & X \end{bmatrix} Ch1' = \begin{bmatrix} A & B & C \\ P & Q & R \\ G & H & I \\ J & K & L \end{bmatrix} Ch2' = \begin{bmatrix} M & N & O \\ D & E & F \\ S & T & U \\ V & W & X \end{bmatrix} \quad (9)$$

The description of the generic algorithm that implements the crossover technique being used here.

```
Algorithm Crossover is
Input: List of parties, L
       List of issues, I
Output: New chromosomes, C1', C2'
i  := select random issue from I
if (interspecies)
   s1 = select random species
   s2 = select random species such that s1 != s2
if (randomparents)
C1 := select random ch from s1
C2 := select random ch from s2
else
C1 := select best ch from s1
C2 := select best ch from s2
```

```
else
   s1 = select random species
if (randomparents)
C1 := select random ch from s1
C2 := select random ch from s1 such that C1 != C2
else
C1 := select best ch from s1
C2 := select second best ch from s1
swap issues and generate C1', C2'
return C1', C2'
```

5.3 Heredity

Heredity is generally defined as the passing of specific traits from parents to offspring. In this process, the offspring inherits characteristics that may be described as similar to the ones of the parent. During the evolution, the species usually tend to accumulate the best characteristics of their ancestors. In this work, heredity is a very simple unary operator which creates a new chromosome with the same characteristics of the parent, i.e., the same distribution. The objective is to apply this operator to the best individuals only, thus passing the best characteristics of one generation to the next, avoiding losing the best of each generation. However, this operator must be used with caution as an excessive use will result in a population that evolves slowly or that does not evolve at all.

In fact, the weight of each of the above described operators must be chosen appropriately. The *Crossover*operator can be applied thoroughly to the population. However, the *Heredity* and the *Mutation* operators must be applied in smaller amounts. In fact, a big incidence of the *Mutation* operator will significantly increase the variety of the solutions, making it harder for a convergence to emerge. On the other hand, a big incidence of the *Heredity* operator would have the exact opposite problem, i.e., the evolution would stop and new favorable solutions would hardly appear. In that sense, these two operators can be useful as long as they are used in small proportions.

6 Termination

The process of selection of the fittest and reproduction is repeated until a termination condition is reached: a non-evolving fitness of the population or the number of iterations established in the initialization. At this point, the system has a significant number of solutions. However, some of them will be very similar to each other while others have simply no interest because their value of fitness is low. In that sense, it is not feasible or productive to present the parties or mediators with all this information.

Thus being, only the best solutions of each generation are available to be used by mediators or parties. This helps to simplify the information generated, allowing the users to be focused on what really matters. Figure 4 depicts the simplified view of the

information generated, in terms of the solutions attained, and their lines of evolution. Each solution is represented with one or more colors. A solution with a given color means that it belongs to the species of that color. This will allow one to see the natural emergence of species, i.e., the lines of evolution that tend towards the maximization of the fitness value for a given party. Colorless individuals denote solutions that are not among the most fit for a particular population but generate offspring that are among the best of future ones and, for that reason, were included in the group of relevant solutions. It is also possible to look at a chromosome's content, as well to its fitness, and the mean fitness, by clicking on it.

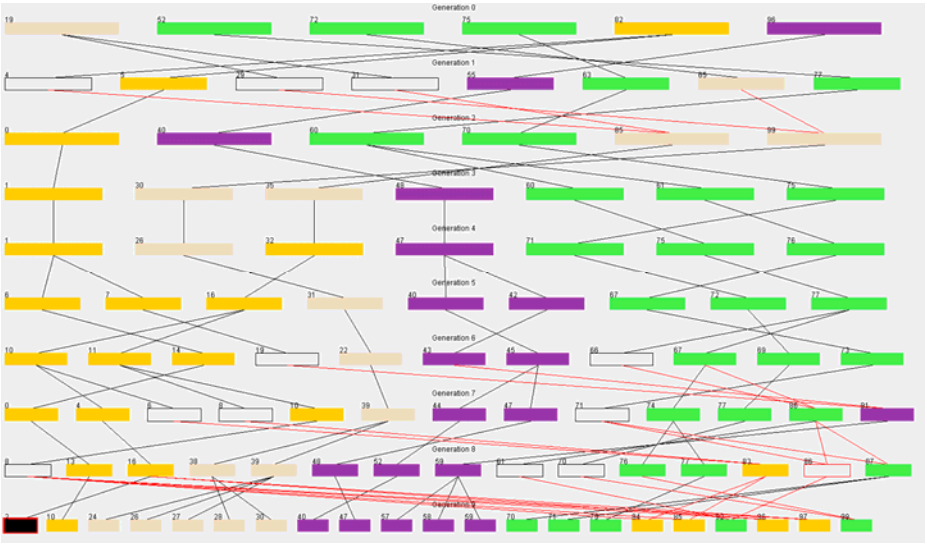


Fig. 4. The lines of evolution of the genetic course and their outcome.Only the individuals that lead to the best leafs are shown.

The lines between individuals represent the parent-offspring relationships. A unary genetic operator generated an individual that has a single line connecting to the previous population, while an individual that has two lines was generated by crossover.

These solutions can then be proposed to the parties by a mediator or by the conflict resolution system. We are currently working on the development of an intelligent conflict resolution environment that is able to collect important information from the context of interaction of the parties [12]. This information includes the levels of stress, the emotional state or the levels of escalation.Based on this, the conflict resolution system or the mediator will select in each time, the most indicated solution for the parties. This will result in a dynamic conflict resolution environment that allows strategies to be adapted in real-time, according to relevant changes in the context-of-interaction.

7 Conclusions

One of the most serious limitations in a conflict resolution process is the inability or unwillingness of parties to design solutions for the resolution of the conflicts. The work described in this paper was developed with the objective of empowering parties and mediators in a conflict resolution process with a tool that is able to provide solutions for concrete problems. Moreover, the solutions generated may be described as fair since they take into consideration not only the monetary value of the items assigned to each party but also the personal value that each party allocates to each item. In that sense, the solutions proposed are more likely to be accepted by the parties.

Compared with our previous case-based approach, this line of attack has as main advantage the independence of a case-base, i.e., the amount and quality of the solutions retrieved does not depend on the quality, quantity or legal domain of the cases in a case-base. In that sense, it provides a more complete answer to the problem. Moreover, despite the computational inefficiency that is generally associated to evolutionary approaches, the performance is good enough for the domain of conflict resolution. In fact, the solutions may be generated as soon as the parties finish providing the data for their case and even before the actual conflict resolution process starts (which is not immediately). In that sense, we can use relatively large parameters on the GA algorithm (e.g. population size, number of generations) ensuring that a big enough number of solutions are generated from which to choose from.

We are now merging this tool into our conflict resolution platform as a solution generation module, to propose solutions during a negotiation process.

Acknowledgments. The work described in this paper is included in TIARAC - *Telematics and Artificial Intelligence in Alternative Conflict Resolution Project* (PTDC/JUR/71354/2006), which is a research project supported by FCT (Science & Technology Foundation), Portugal. The work of Davide Carneiro is also supported by a doctoral grant by FCT (SFRH/BD/64890/2009).

References

1. Katsh, E., Rifkin, J., Gaitenby, A.: E-Commerce, E-Disputes, and E-Dispute Resolution: In the Shadow of eBay Law. Ohio State Journal on Dispute Resolution 15, 705 (1999)
2. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press (2002)
3. Brown, H., Marriott, A.: ADR Principles and Practice. Sweet and Maxwell (1999)
4. Bennett, S.C.: Arbitration: essential concepts. ALM Publishing (2002)
5. Katsch, E., Rifkin, J.: Online dispute resolution – resolving conflicts in cyberspace. Jossey-Bass Wiley Company, San Francisco (2001)
6. Lodder, A., Thiessen, E.: The role of artificial intelligence in online dispute resolution. In: Workshop on Online Dispute Resolution at the International Conference on Artificial Intelligence and Law, Edinburgh, UK (2003)
7. Peruginelli, G.: Artificial Intelligence in Alternative Dispute Resolution. In: Sartor, G. (ed.) Proceedings of the workshop on the Law of Electronic Agents, LEA 2002 (2002)

8. Carneiro, D., Novais, P., Andrade, F., Neves, J.: Retrieving Information in Online Dispute Resolution Platforms: A Hybrid Method. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Law. University of Pittsburgh School of Law. ACM (2011) ISBN: 978-1-4503-0755-0
9. Carneiro, D., Novais, P., Neves, J.: An Agent-Based Architecture for Multifaceted Online Dispute ResolutionTools. In: Mehrotra, K.G., Mohan, C., Oh, J.C., Varshney, P.K., Ali, M. (eds.) Developing Concepts in Applied Intelligence. SCI, vol. 363, pp. 89–94. Springer, Heidelberg (2011)
10. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning, 1st edn. Addison-Wesley Professional (1989)
11. Thomas, K., Kilmann, R.: Conflict and Conflict Management (1974), `http://www.kilmann.com/conflict.html` (accessed in May 2010)
12. Carneiro, D., Gomes, M., Novais, P., Neves, J.: Developing Dynamic Conflict Resolution Models Based on the Interpretation of Personal Conflict Styles. In: Antunes, L. (ed.) EPIA 2011. LNCS(LNAI), vol. 7026, pp. 44–58. Springer, Heidelberg (2011)
13. Corchado, E., Abraham, A., Carvalho, A.C.P.L.F.D.: Hybrid intelligent algorithms and applications. Inf. Sci., 2633–2634 (2010)
14. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
15. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)

Initialization Procedures for Multiobjective Evolutionary Approaches to the Segmentation Issue

José L. Guerrero, Antonio Berlanga, and José Manuel Molina

University Carlos III of Madrid, Computer Science Department
Group of Applied Artificial Intelligence
Avda. Universidad Carlos III, 22, Colmenarejo, Spain
jguerrer@inf.uc3m.es aberlan@ia.uc3m.es molina@ia.uc3m.es

Abstract. Evolutionary algorithms have been applied to a wide variety of domains with successful results, supported by the increase of computational resources. One of such domains is segmentation, the representation of a given curve by means of a series of linear models minimizing the representation error. This work analyzes the impact of the initialization method on the performance of a multiobjective evolutionary algorithm for this segmentation domain, comparing a random initialization with two different approaches introducing domain knowledge: a hybrid approach based on the application of a local search method and a novel method based on the analysis of the Pareto Front structure.

Keywords: Initialization, Segmentation, Evolution Strategies, Multiobjective Optimization, Memetic algorithms.

1 Introduction

Evolutionary algorithms are a versatile tool to deal with a huge variety of problem domains [1], being convergence speed one of their known issues. Hybridization of intelligent systems [6] to overcome their handicaps is one of the most important trends in artificial intelligence, with applications in fields as diverse as robotics, dimensionality reduction, reasoning methods or multiobjective optimization [5]. Hybrid applications of evolutionary algorithms combine their exploratory capabilities with the information exploitation provided by local search procedures, being also known as memetic algorithms [12].

The initialization of the population in evolutionary algorithms has been considered as a key operator for the final quality of the result and the computational cost required to obtain that result [2]. The most popular initialization process consists in a random initialization of the values for the chosen representation. This procedure aims to maximize the coverage of the search space and, thus, the exploration capabilities of the algorithm. Different alternatives have been considered to improve this behavior, such as novel general alternative initialization strategies [14], reuse of previous solutions [15] or the introduction of specific

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 452–463, 2012.

domain information [3]. In general, default initialization processes are designed to provide a reasonable performance over a wide series of problems, but the injection of additional information may help to improve that performance over specific problem instances (following the no-free-lunch theorem [18]).

Segmentation problems are based on the division of a given curve in a set of n segments (being each of these segments represented by a linear model, which points to another common naming convention for this process: piecewise linear representation, PLR) minimizing the representation error. This issue has been faced from several perspectives, such as time series segmentation [11] or polygonal approximation [16], leading to different algorithms (some of which are closely related). Regarding evolutionary algorithms, they have been applied to this issue using different specific operators and focuses [19,9]. Multi-objective evolutionary algorithms optimize several objectives that may be in conflict with each other at the same time [4], and have been also applied to segmentation issues [7], dealing with the number of segments and representation error of given solutions to provide a Pareto Front of non-dominated solutions.

This paper analyzes the effect of different initialization methods on a multi-objective approach for the segmentation issue. The default random initialization method is compared with different alternatives, based on local search information exploitation (according to similar hybridization principles to the ones used by memetic algorithms, but applied to population initialization) or on the analysis of the Pareto front and its relationship to the input variables.

The work is organized as follows: the second section introduces the formalization of the segmentation issue as a multiobjective problem, along with an evolutionary approach to solve it. The third section covers three different initialization procedures, highlighting their characteristics and differences, while the fourth presents the used dataset, along with the results obtained for the techniques covered in the previous section. Finally, the conclusions obtained from these results are presented, along with possible future lines.

2 Segmentation Formalization

A segmentation process divides a series of data into a certain number of individual segments according to a model (or set of models) minimizing the representation error. This work is focused on PLR segmentation (or polygonal approximation) which uses linear models for this approach. This process can also be seen as the search of the individual points which minimize the overall approximation error. These points are usually called *dominant points*. A possible formalization for this process is presented in equation 1.

$$S(T) = \{B_m\} \rightarrow B_m = \{\boldsymbol{x}_k\} j \in [k_{min}...k_{max}] m \in [1...seg_{num}] \\ \underset{max}{min} f_{quality}(\{B_m\}) \tag{1}$$

where T is the original data, $S(T)$ is the segmentation process, B_m is a given resultant segment from that process and $f_{quality}$ is the used quality function.

Depending on the definition of this quality function, the objective may be to minimize or maximize this function. The quality of a segmentation process is traditionally determined by the following criteria [11]:

1. Minimizing the overall representation error ($total\_error$)
2. Minimizing the number of segments such that the representation error is less than a certain value ($max\_segment\_error$)
3. Minimizing the number of segments so that the total representation error does not exceed $total\_error$

The previous considerations introduce several interesting facts. First of all, the quality of a segmentation process depends on several objectives in conflict: minimizing the number of required segments while minimizing the representation error. Secondly, the configuration of such processes can be problem dependent due to the required parameters ($total\_error$ and $max\_segment\_error$) and, thus, a general technique may be hard to apply to a set of problem instances with consistent results. The consideration for the measurement of several objectives in conflict to test the quality of a segmentation process was faced in [8] with the use of multiobjective quality indicators [20]. Given the multiobjective nature of this process, in [7] the segmentation issue was formalized according to equation 2, which explicitly presents this nature.

$$S(T) = \{B_m\} \rightarrow B_m = \{\boldsymbol{x}_k\} j \in [k_{min}...k_{max}] m \in [1...seg_{num}]$$
$$\begin{cases} d(S(T), T) \leq total_{error} \\ \forall m, d(f_{ap}(B_m), B_m) \leq max\_segment\_error \end{cases} \quad (2)$$

where *d(x, y)* is a distance error function between segments x and y and $f_{ap}(x)$, which is the approximation function result over segment x. According to the traditional criteria, $total\_error$ and $max\_segment\_error$ represent certain constrains which are required by certain segmentation algorithms in order to determine whether they must be stopped [11]. It must be noted that these characteristics may change abruptly among different problem instances, even though they may be used to constrain the search process of the evolutionary approach. Since these parameters are not strictly required and their choice is neither trivial nor problem independent, they have been excluded from the presented approach to provide a more focused discussion.

In [7] the proposed codification is based on integer values representing the *dominant points* of the segmentation process (the edges of the segments which the data is divided into). This representation was introduced in order to preserve the importance of the different dominant points obtained during the evolutionary process, but also implied an increase over the size of the search space when compared to a more commonly used genetic codification [19]. This codification formalizes a representation where each chromosome has a size equal to the length of the data being analyzed, and each gene represents whether that particular position is considered a *dominant point*. This codification is the one followed in the current approach, due to its reduced search space.

Several error functions may be used as well. Two different fitness functions used in evolutionary approaches to segmentation are the maximum error function (equation 3) and the integral squared error function (equation 4). This last option is the one followed in this work.

$$E_{\infty}(\alpha) = \underset{1 \leq i \leq n}{max}\, e_i(\alpha) \tag{3}$$

$$E_2(\alpha) = \sum_{i=1}^{n} [e_i(\alpha)]^2 \tag{4}$$

The remaining operators are chosen according to standard values: bit-flip mutation, 1-point crossover and binary tournament selection. The crossover probability used is 0.9 and the mutation probability is $1/length$. These operators are not problem specific (nor their associated probabilities), differing from some of the single objective approaches available. The general multiobjective algorithm chosen is SPEA2 [21], which introduces an archive to keep track of the best solutions found during the evolutionary cycles of the different generations. In this problem it is crucial to preserve non-dominated solutions found at different points of the evolutionary cycle, leading to the choice of this algorithm along with an archive size equal to the length of the problem instance being solved, in order to be able to, ideally, store one non-dominated solution for each of the different possible representations regarding their number of dominant points, while the chosen population size will be 100 individuals.

3 Population Initialization

Convergence speed is a constant issue in evolutionary computation, and it has been approached with modifications in the different involved processes: crossover, mutation, selection, etc. Initialization procedures have received a reduced amount of interest from the research community, generally assuming that the overall impact over the performance of the algorithm is lower. Most genetic algorithms use a default bitstring uniform initialization procedure, assigning values of 0 or 1 to every bit for each individual in the population, obtaining a uniform population regarding the binary space, which also exhibits the maximal bit-wise diversity [10]. However, early research showed that this may not be the optimal initialization procedure for specific domains, such as inverse problems in Structural Mechanics [17], where the solutions were known to contain far more 0's than 1's.

General approaches have to provide a trade-off between the improved initial population obtained and the cost of the process. Such a discussion is carried out in [14], where opposition-based and quasi-random [13] initialization methods are compared, highlighting the computational issues and dimensionality effectiveness. In [15] the reuse of previous solutions in terms of population initialization is considered for the application of evolutionary algorithms to dynamic environments, but the established principles can be used for static environments where

an approximation to the solution is known (or can be calculated, as in the local search based method compared in this work). Finally, domain specific approaches introduce characteristics from the faced problem in order to include a seeding in the initial population which can improve the overall results. In [3] such an approach is studied for the timetabling problem, where heuristic individuals go through some randomization process in order to generate the initial population, presenting a discussion of the diversity effect of such a process over the final outcome of the algorithm.

Three different initialization procedures will be compared for the presented problem: default (bitstring uniform), uniform (in terms of Pareto front) and local search. Default initialization assigns a 50% chance of becoming a *dominant point* to each point in the original data. According to that probability, this method generates an initial population which, in the number of segments objective function, is centered around 1/2 of the number of original elements in the data. Being this objective also closely related to the representation error, this generates a poor diversity on the number of segments (or, similarly, the number dominant points), which also implies a poor diversity on the covered range of approximation errors.

Even though default initialization produces the maximal bit-wise diversity, a poor one is obtained in the resultant Pareto front. Since multiobjective optimization seeks the Optimal Pareto Set in the variable space and its associated Optimal Pareto Front in the objective space (the set of solutions where one solution objective function value cannot be improved unless another objection function value is degraded [4]), this may not be the optimal strategy. Uniform initialization tries to ensure the diversity of the front obtained. To achieve this task, each individual is generated according to a number of random dominant points, which are then included into the chromosome at random gene positions. This generates a population which is spread along the dominant points objective, obtaining as well a good diversity over the representation error objective function. Related to the initialization approaches presented at the beginning of this section, this approach is general (in terms of exploiting the Pareto front diversity in the initial population) but uses a domain specific procedure to produce the front with a very low computational cost.

Local search initialization is a heuristic seeding approach using bottom-up segmentation [11] to introduce good individuals into the initial population, a technique which is claimed to obtain comparatively better results than other offline alternatives. This algorithm creates the finest possible approximation of the time series, dividing it into n-1 (where n is the number of points in the time series) segments of length value 2. Afterwards, the cost of merging each pair of adjacent segments is calculated and, if the merge with the lowest cost has an error bellow the user defined value, the segments are merged. The process continues until no pair of adjacent segments can be merged with an acceptable error value. It is important to notice that in every step of the algorithm the costs of the adjacent segments to the merged one in the previous step must be updated.

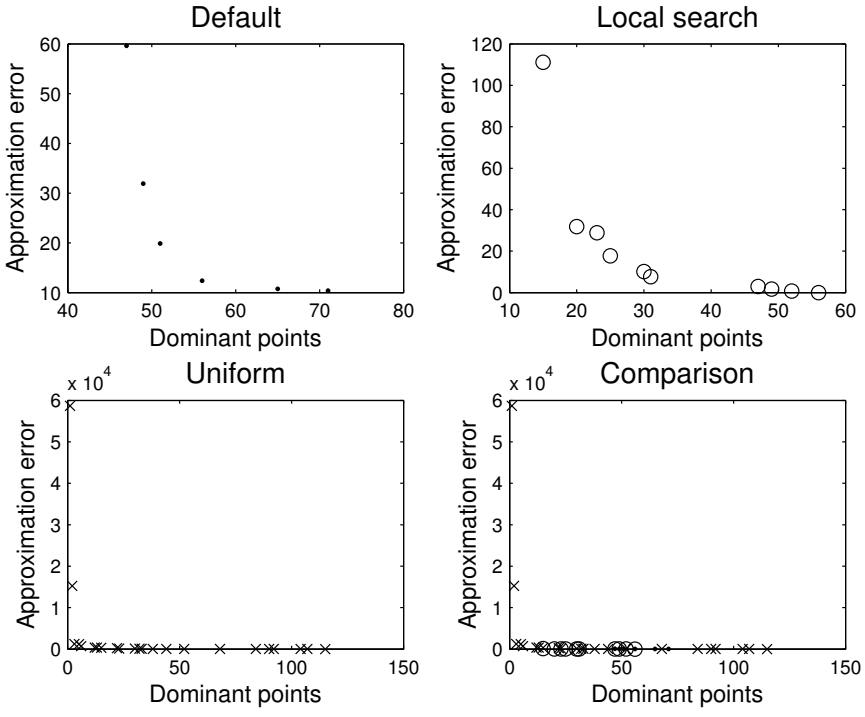


Fig. 1. Initial Pareto front comparison for the three presented methods (leaf curve)

One of the difficulties arising in the application of these single objective procedures is that, in order to obtain a certain number of different individuals to be introduced into the initial population, there is a lack of direct control over the objective functions values. This may require several executions to obtain a single individual which can be introduced into the population, thus increasing the overall computational cost. On the other hand, unlike other presented alternatives in the literature ([3]) different individuals are obtained with the different configuration parameters directly from the heuristic technique, eliminating the requirement for additional randomization processes.

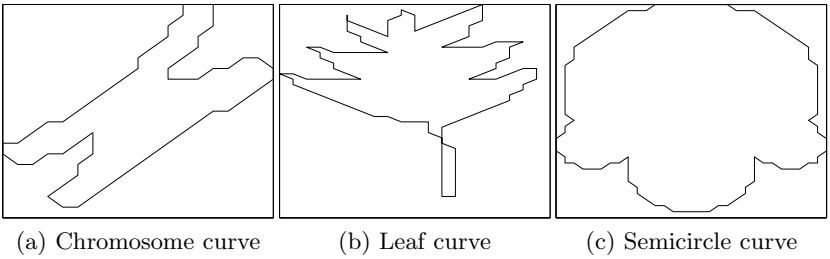
Figure 1 presents the non-dominated solutions obtained in an initial population of 100 individuals generated with the default method and the proposed approach based on the diversity in the objective space, along with a Pareto front composed from ten solutions obtained with different runs of the detailed single-objective algorithm. As expected, the range in the objective functions covered by the default initialization is very limited compared to the one which focuses on objective function diversity. The number of non-dominated individuals generated is clearly inferior to those in the uniform approach as well, obtaining an initial population which, even though it is composed of the same number of individuals, provides the algorithm with less valuable information (Pareto front individuals).

Local search initialization provides individuals which are clearly superior to the ones randomly initialized (by either of the alternative procedures), but their range is limited compared to the ones performed by uniform initialization.

4 Experimental Validation

Along with the performance of the presented initialization methods, current experimental validation will try to determine whether the inclusion of local search individuals in the population generated by either of the alternative methods improves its results. Three commonly used curves from the polygonal approximation domain are introduced into the data set: chromosome, semicircle and leaf, presented in figure 2. For the validation of the performance of the different initialization methods, 30 runs of every configuration have been performed, the unary hypervolume [20] of the resultant Pareto Front calculated for each of the alternatives (both for the initial and final populations), and the difference between the different pairings calculated. Afterwards, a t-test is carried out to determine the statistical significance of the obtained results. The reference front used for the hypervolume computation is obtained with a uniform initialization procedure and a population size of 1000 individuals left to run for 2000 generations.

The representation of the initial population Pareto fronts for the three curves in the dataset are presented in figures 1 (leaf), 3 (chromosome) and 4 (semicircle). Graphically these figures show several interesting facts regarding the proposed initialization: assuming that the heuristic seeding provided by the local search technique provides good solutions in terms of objective functions values and diversity, the comparison with the default process shows that bitstring uniform populations may provide good (figure 3) or very bad solutions (figure 1), being this quality problem dependent (determined by whether the solutions around 50% dominnant points are meaningful or not for the final Pareto front), disencouraging the use of this technique for an unknown problem instance. On the other hand, the initial populations provided by the uniform method exhibit for all the different dataset instances Pareto fronts with a very good diversity over the two objectives, being thus applicable to new unknown instances with a certain guarantee over the quality of the initial population's Pareto front.



(a) Chromosome curve (b) Leaf curve (c) Semicircle curve

Fig. 2. Curves included in the data set

Table 1. Initial populations comparison

Chromosome curve								
Default		L.S.	Uniform		Unif. + l.s.		Def. + l.s.	
Mean	Std	Mean	Mean	Std	Mean	Std	Mean	Std
4.47E-01	4.39E-02	8.59E-01	9.54E-01	7.52E-03	9.60E-01	6.82E-03	8.59E-01	9.94E-07
Leaf curve								
1,66E-01	3,22E-02	7,45E-01	9,62E-01	1,99E-02	9,63E-01	1,99E-02	7,45E-01	3,39E-16
Semicircle curve								
2,80E-01	5,21E-02	8,08E-01	9,50E-01	2,42E-02	9,51E-01	2,42E-02	8,08E-01	4,52E-16

Table 2. Final populations comparison

Chromosome curve							
Default		Uniform		Unif. + l.s.		Def. + l.s.	
Mean	Std	Mean	Std	Mean	Std	Mean	Std
9,41E-01	2,97E-02	9,67E-01	1,07E-04	9,66E-01	4,70E-03	9,67E-01	1,07E-04
Leaf curve							
7,77E-01	4,76E-02	9,79E-01	3,55E-03	9,76E-01	1,20E-02	9,77E-01	7,58E-03
Semicircle curve							
8,46E-01	4,06E-02	9,75E-01	5,27E-03	9,76E-01	3,08E-03	9,77E-01	3,39E-04

The hypervolume results obtained for the three different curves are presented in tables 1 and 2, while the statistical significance results over those values are presented in table 3. The initial populations comparison does not provide a standard deviation value for the local search initialization, since each of the runs starts with the exact same initial population. In final populations, no results for local search are provided, since, as will be detailed, the populations obtained by local search dominate those created by a default initialization process, providing the same final results (disregarding the stochastic nature of evolutionary approaches) in local search and local search plus default initialization configurations (being these results included under this last heading).

The test results presented in table 3 are obtained from the final populations, since all the differences in the initial ones were statistically significant. The results show that uniform initialization yields better performance of the algorithm compared to any of the remaining alternatives, and also that the addition of local search individuals to its initial population does not improve its results (in the final outcome of the algorithm). However, local search use does improve (for the two harder problem instances, lead and semicircle) the default initialization performance.

The initial populations provided by the different runs of a default initialization procedure become, in general, fully dominated by the individuals introduced by the local search (results in table 1 for local search and local search plus default individuals are the same). The impact of the local search procedures is related

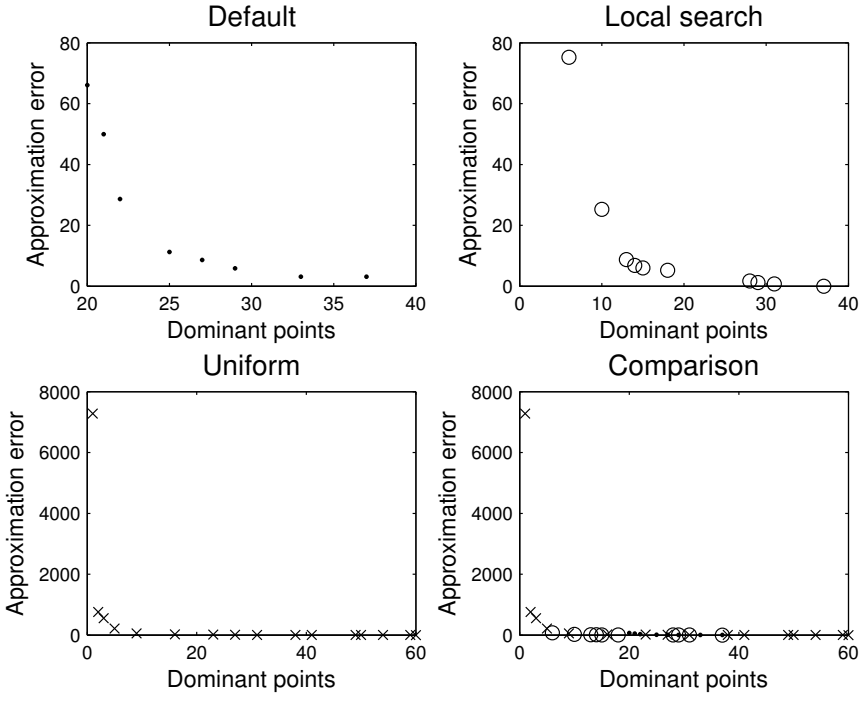


Fig. 3. Initial Pareto front comparison for the chromosome curve

to the quality of its results compared to the optimal Pareto front and the cost of their computation. As presented in table 3, the heuristic seeding does improve the results of the bitstring random initialization process (in two of the three curves in the dataset), but also requires a computational cost to obtain those individuals. As previously explained, obtaining n individuals for this initial population by means of the local search procedure may require more than n executions of this algorithm, and this cost may be even higher if certain diversity is required in those heuristic individuals.

Uniform initialization provides a higher range of objective function values to its individuals (which are graphically represented by the initial and final *"tails"* of the Pareto front), which provides additional non-dominated individuals to the algorithm and allowing it to obtain better final solutions, as seen in

Table 3. Statistical significance test

Curve	Def/l.s.	Def./Unif.	Unif./l.s.	Unif./Unif. + l.s.	Def./Def. + l.s.
Chromosome	No	Yes	Yes	No	No
Leaf	Yes	Yes	Yes	No	Yes
Semicircle	Yes	Yes	Yes	No	Yes

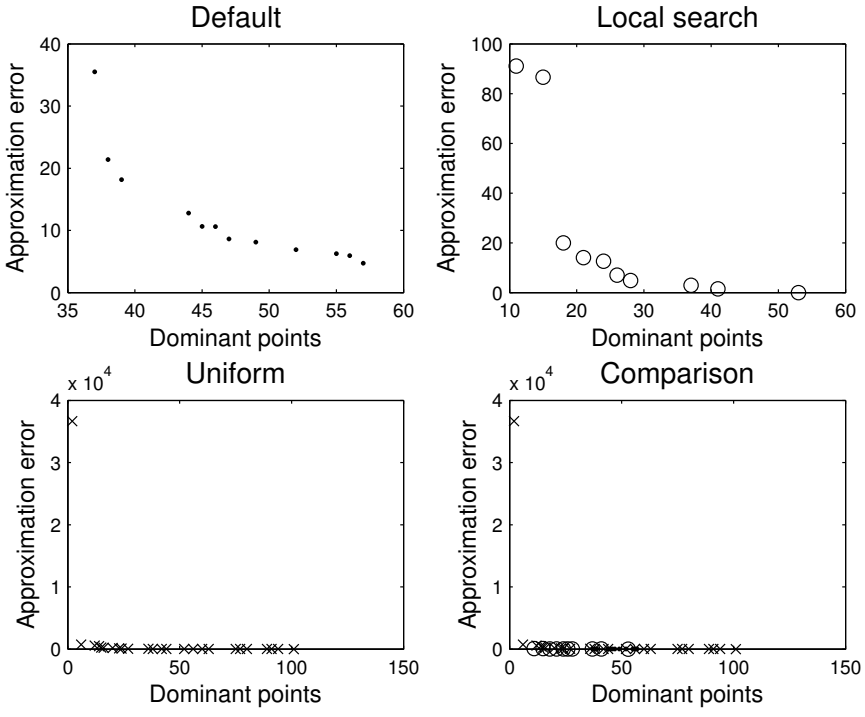


Fig. 4. Initial Pareto front comparison for the semicircle curve

table 3. This shows the importance of the diversity in terms of objective space, which cannot be obtained with the default bitstring random initialization. Even though there is no general technique to be able to obtain this diversity in the objective space for a general problem, the presented technique allows to do so in the segmentation domain with a very low computational cost (similar to that of the default initialization process) being clearly superior to the considered alternatives.

5 Conclusions

This work has presented the importance of initialization for evolutionary approaches, particularly for the segmentation issue. A common approach to include domain information into an evolutionary approach is to perform a hybridization including some local search step, which involves a considerable computational cost but aims to accelerate the exploitation step of the search, with the possible degradation of exploration capabilities. An overview of different initialization approaches is presented, and a comparison among three different possibilities is carried out: random initialization, a hybrid initialization including individuals obtained by means of a local search based procedure and a uniform approach

based on the analysis of the Pareto Front shape in order to obtain an initial population focused on the diversity of individuals in the objective space. This uniform approach yields a performance not only better than the one provided by default initialization, but also superior to the one provided by a local search based initial population. The addition of local search individuals to an initial population generated by the uniform approach showed no statistically significant improvements in the outcome of the algorithm.

The measured improved performance comes from the amount of valuable information contained in the Pareto Front obtained: the increased covered ranges of objective function values by the individuals in the initial population provide a higher number of non-dominated individuals, which allows a better final performance of the algorithm, highlighting the importance of diversity in the objective space rather than the decision variables space. Future lines of this work include the inclusion of local search procedures at different steps of the evolutionary algorithm, additional research on initial population creation methods and the study of the applicability of these techniques to different multiobjective problems.

Acknowledgments. This work was supported in part by Projects CICYT TIN2011-28620-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

1. Affenzeller, M., Winkler, S.: Genetic algorithms and genetic programming: modern concepts and practical applications. Chapman & Hall/CRC (2009)
2. Bramlette, M.: Initialization, mutation and selection methods in genetic algorithms for function optimization. In: Proceedings of the Fourth International Conference on Genetic Algorithms, vol. 100, p. 107. Morgan Kaufmann, San Mateo (1991)
3. Burke, E., Newall, J., Weare, R.: Initialization strategies and diversity in evolutionary timetabling. Evolutionary Computation 6(1), 81–103 (1998)
4. Coello, C., Lamont, G., Van Veldhuizen, D.: Evolutionary algorithms for solving multi-objective problems. Springer-Verlag New York Inc. (2007)
5. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
6. Corchado, E., Graña, M., Wozniak, M.: Editorial: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
7. Guerrero, J., Berlanga, A., García, J., Molina, J.: Piecewise linear representation segmentation as a multiobjective optimization problem. Distributed Computing and Artificial Intelligence, 267–274 (2010)
8. Guerrero, J., García, J., Molina, J.: Piecewise linear representation segmentation in noisy domains with a large number of measurements: the air traffic control domain. International Journal on Artificial Intelligence Tools 20(2), 367–399 (2011)
9. Ho, S., Chen, Y.: An efficient evolutionary algorithm for accurate polygonal approximation. Pattern Recognition 34(12), 2305–2317 (2001)
10. Kallel, L., Schoenauer, M.: Alternative random initialization in genetic algorithms. In: Proceedings of the 7th International Conference on Genetic Algorithms, pp. 268–275 (1997)

11. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. Data mining in time series databases, pp. 1–21 (2003)
12. Krasnogor, N., Smith, J.: A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation 9(5), 474–488 (2005)
13. Maaranen, H., Miettinen, K., Mäkelä, M.: Quasi-random initial population for genetic algorithms*. Computers & Mathematics with Applications 47(12), 1885–1895 (2004)
14. Rahnamayan, S., Tizhoosh, H., Salama, M.: A novel population initialization method for accelerating evolutionary algorithms. Computers & Mathematics with Applications 53(10), 1605–1614 (2007)
15. Ramsey, C., Grefenstette, J.: Case-based initialization of genetic algorithms. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 84–91 (1993)
16. Sarfraz, M.: Linear Capture of Digital Curves. In: Interactive Curve Modeling, pp. 241–265. Springer, London (2008)
17. Schoenauer, M.: Shape representations and evolution schemes. In: Proceedings of the Fifth Annual Conference on Evolutionary Programming, pp. 121–129 (1996)
18. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)
19. Yin, P.Y.: A new method for polygonal approximation using genetic algorithms. Pattern Recognition Letters 19(11), 1017–1026 (1998)
20. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., Da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
21. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, EUROGEN 2001, Athens, Greece, pp. 95–100 (2001)

Optimization of Neuro-Coefficient Smooth Transition Autoregressive Models Using Differential Evolution

Christoph Bergmeir[1], Isaac Triguero[1], Francisco Velasco[2], and José Manuel Benítez[1]

[1] Department of Computer Science and Artificial Intelligence, CITIC-UGR, University of Granada, Spain
{c.bergmeir,triguero,j.m.benitez}@decsai.ugr.es
[2] Department of Computer Languages and Systems, CITIC-UGR, University of Granada, Spain
fvelasco@ugr.es
http://dicits.ugr.es

Abstract. This paper presents a procedure for parameter estimation of the neuro-coefficient smooth transition autoregressive model, substituting the combination of grid search and local search of the original proposal of Medeiros and Veiga (2005, IEEE Trans. NN, 16(1):97-113) with a differential evolution scheme. The purpose of this novel fitting procedure is to obtain more accurate models under preservation of the most important model characteristics. These are, firstly, that the models are built using an iterative approach based on statistical tests, and therewith have a mathematically sound construction procedure. And secondly, that the models are interpretable in terms of fuzzy rules. The proposed procedure has been tested empirically by applying it to different real-world time series. The results indicate that, in terms of accuracy, significantly improved models can be achieved, so that accuracy of the resulting models is comparable to other standard time series forecasting methods.

Keywords: time series, statistical models, threshold autoregressive models, NCSTAR, differential evolution.

1 Introduction

Prediction and modeling of time series is an interdisciplinary field with foundations in Statistics, while their application spans many fields in Science and Engineering. However, modeling techniques developed in the context of Artificial Intelligence have also found a fertile area for adaptation and exploitation within this field. Classical statistical schemes for modeling time series have their fundamental reference in the work of Box and Jenkins [6] in the 1970's. The reference model is the autoregressive (AR) moving average model, which is characterized by linear relationships between lagged values of the series, thus performing a linear regression. But with the presence of computer systems in nearly all aspects of everyday life, nowadays often huge amounts of data are available for analysis,

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 464–473, 2012.

facilitating the search for more complex, not necessarily linear patterns. So, the need to model nonlinear behavior arises.

One of the best known nonlinear models is that proposed by Tong [18]. It uses two or more linear models to fit disjoint regions of the series' domain, using a variable threshold for transition between the models. In the literature there are numerous variants of this idea, known generically as threshold AR models (TAR). For example, in the so-called self-excited TAR, the threshold variable is a delayed value of the series. Or, considering the generally continuous nature of the series, TAR models that replace the abrupt change of a linear model to another for a smooth transition function are also common. These are denoted smooth transition AR models. Based on this, Medeiros and Veiga [12] proposed the neuro-coefficient smooth transition AR model (NCSTAR). The NCSTAR is a linear model whose coefficients change over time and are determined using a multi-layer perceptron [1]. In addition to the model, those authors presented a construction method that works in an iterative manner to determine the number of hidden units in the neural network, based on techniques of statistical hypothesis testing.

Another important advantage of TAR models is that each of them can be expressed in terms of a fuzzy rule-based system (FRBS) [1–4]. This offers a theoretical framework for the interpretability of such models, and so enables forecast practitioners not only to use the model as a black box system, but also to, e.g., gain valuable information about the underlying processes.

The NCSTAR model fitting procedure is a combination of a grid search and a local search method. This can be decisive during the incremental building of an NCSTAR, especially w.r.t. the accuracy of the model. In contrast, evolutionary algorithms [9] are techniques that have proven very efficient in order to address optimization problems in various fields [19]. Specifically, the differential evolution algorithm (DE) [17] stands as one of the most promising methods in continuous optimization problems, and as such seems worth to be evaluated as a method for the NCSTAR model fitting problem.

So, the aim of this paper is to provide a process for training and adjustment of NCSTAR models based on the evolutionary algorithm DE, with the intent to obtain models that preserve the natural interpretability of the models, being at the same time more accurate.

The rest of the paper is organized as follows: Section 2 develops the theory of threshold autoregressive models. Section 3 describes the most important features of the DE algorithm. Section 4 presents the proposed algorithm using differential evolution to construct NCSTAR models. Section 5 shows the experiments, and Section 6 concludes the paper.

2 Threshold Autoregressive Models

In a linear autoregression as popularized by Box and Jenkins [6], the value at the current time t of a time series x is estimated by a linear combination of past values of the series in the following way:

$$x(t) = a_0 + \sum_{i=1}^{d} a_i x(t-i) + e(t) \tag{1}$$

Here, d denotes the number of past values that are taken into account, $a_0, \ldots, a_d$ are the model coefficients, and e is a series of errors. The values of e are independent and identically distributed (i.i.d.).

With $\mathbf{z}(t) = (1,\ x(t-1),\ x(t-2), \ldots,\ x(t-d))^T$ and $\mathbf{a} = (a_0, \ldots, a_d)$, Equation (1) can be written in vector notation:

$$x(t) = \mathbf{a}\mathbf{z}(t) + e(t). \tag{2}$$

A piece-wise linear model, i.e., a model that combines various linear models, can be generally defined by:

$$x(t) = \sum_{j=1}^{k} f_j(th_j(t))\mathbf{a}_j\mathbf{z}(t) + e(t). \tag{3}$$

Here, the functions f_j are nonlinear functions that combine the linear models, and th_j are the threshold functions that define the criterion on which the switching is based (exogenous variables, delayed values of the series, or both).

In the TAR model, indicator functions are used, i.e., $f_j := I_j$, which perform sharp switches between the linear models, depending on the current value of the threshold variable, $th_j(t)$, in the following form:

$$I_j(th_j(t)) = \begin{cases} 1, & \text{if } th_j(t) \in (c_{j-1}, c_j]; \\ 0, & \text{otherwise.} \end{cases}$$

Here, $c_0 \ldots, c_k$ are threshold constants with $-\infty = c_0 < c_1 < \ldots < c_k = \infty$. In the self-exciting TAR model (SETAR), $th_j(t)$ is then defined, using a delay parameter d, as a concrete delayed value $x(t-d)$.

In order to avoid the sharp switching between models, which might not reflect well the behavior of the series, smooth switching between regimes can be achieved by using, e.g., the logistic function:

$$f_j(th_j(t)) = \frac{1}{1 + \exp(-\gamma_j(th_j(t) - c_j))}. \tag{4}$$

The NCSTAR model uses this smooth kind of regime switching with the logistic function, and furthermore the following threshold function:

$$th_j(t) = \omega_j \mathbf{w}(t). \tag{5}$$

Here, $\mathbf{w}(t)$ is a vector composed of the relevant delayed and exogenous variables, and ω_j is the weights vector which has length one, i.e., $\|\omega_j\| = 1$, and the same dimension as $\mathbf{w}(t)$. In the following, we assume that no exogenous variables are taken into account and all variables of the autoregression are also relevant for the threshold, i.e., $\mathbf{w}(t) = (x(t-1),\ x(t-2), \ldots,\ x(t-d))^T$.

The most important advantages of the NCSTAR models are their efficient iterative building process, and their interpretability in terms of fuzzy rule-based systems. Along with the model definition, Medeiros and Veiga [12] developed a method to construct the model using an iterative statistical testing procedure. Furthermore, Aznarte and Benítez [1] showed that "NCSTAR models are functionally equivalent to additive TSK FRBS with logistic membership function," which facilitates greatly their interpretability, as every regime can be directly translated into a fuzzy rule.

In total, the NCSTAR model has the nonlinear parameters γ_j, c_j, and ω_j with $j = 1, \ldots, k$ as in Equations (4) and (5), that are involved in the regime switching, and the linear parameters of the single regimes. When the nonlinear parameters are fixed, the linear parameters can be calculated in a closed form solution. Since γ_j, in theory, is a parameter that has to be scaled with the time series, the series are normalized before application of the NCSTAR model building procedure. In the version of NCSTAR as proposed by Medeiros and Veiga [12], the following procedure, which is a combination of grid search and local search, is used in order to perform optimization of the nonlinear parameters, whenever a new regime j is added:

- ω_j is drawn randomly from a uniform distribution and then normalized to ensure it has length one. If $\omega_{1j} < 0$, the sign of all elements of ω_j is inverted. This is performed M times, in order to obtain m vectors ω_j^m.
- For every ω_j^m, c_j is defined as the median of $\omega_j^m \mathbf{x}$, with $\mathbf{x}$ being the embedded time series, i.e., a matrix composed column-wise by the lagged versions of the series.
- A grid of N values is defined in order to choose γ_j^n (with $n = 1, \ldots, N$) as the values $1, \ldots, N$.

So, the grid consists of $N \times M$ initial solutions for the new regime. Starting from the solution of the grid which, in combination with the regimes already present, allows for the largest reduction of the error, a local search algorithm is started to further improve on all parameters. In the original version of the algorithm, the Levenberg-Marquardt algorithm [13] is used for the local search.

3 Differential Evolution

Differential evolution follows the general procedure of an evolutionary algorithm [8]. DE starts with a population of NP solutions, so-called individuals. The generations are denoted by $G = 0, 1, \ldots, G_{max}$. It is usual to denote each individual as a D-dimensional vector $X_{i,G} = \{x^1_{i,G}, \ldots, x^D_{i,G}\}$, called "target vector."

After initialization, DE applies the mutation operator to generate a mutant vector $V_{i,G}$, with respect to each individual $X_{i,G}$, in the current population. For each target $X_{i,G}$, at the generation G, its associated mutant vector $V_{i,G} = \{V^1_{i,G}, \ldots, V^D_{i,G}\}$. The method of creating this mutant vector is that which differentiates one DE scheme from another. In this contribution, we focus on the *DE/Rand/1* which generates the mutant vector as follows:

$$V_{i,G} = X_{r_1^i,G} + F \cdot (X_{r_2^i,G} - X_{r_3^i,G}) \tag{6}$$

The indices r_1^i, r_2^i are mutually exclusive integers randomly generated within the range $[1, NP]$, which are also different from the base index i. The scaling factor F is a positive control parameter for scaling the different vectors.

After the mutation phase, the crossover operation is applied to each pair of the target vector $X_{i,G}$ and its corresponding mutant vector $V_{i,G}$ to generate a new trial vector that we denote $U_{i,G}$. We will focus on the binomial crossover scheme, which is performed on each component whenever a randomly picked number between 0 and 1 is less than or equal to the crossover rate (CR), which controls the fraction of parameter values copied from the mutant vector. Then, we must decide which individual should survive in the next generation $G + 1$. If the new trial vector yields an equal or better solution than the target vector, it replaces the corresponding target vector in the next generation; otherwise the target is retained in the population.

4 Using DE for the Optimization of the NCSTAR Model

In order to use the DE algorithm as a replacement for the combination of grid search and local search presented in Section 2, we propose the following adaptations:

We define vectors $X = \{\gamma_1, \ldots, \gamma_k, c_1, \ldots, c_k, \omega_{11}, \ldots, \omega_{d1}, \ldots, \omega_{1k}, \ldots, \omega_{dk}\}$, which contain all nonlinear parameters γ_j, c_j, and ω_j (with $j = 1, \ldots, k$) necessary to define an NCSTAR model. The population of the DE algorithm consists of NP such vectors.

Regarding the parameter domains, some problems have to be taken into account: Numerical problems during estimation of the linear parameters of a regime may occur, if its assigned threshold c_j makes it relevant for very few data points only. And, if the value of γ is high, the smoothness of the transition function gets lost, which may yield worse generalization properties. So it is important to choose the parameter domains for the γ_j and c_j properly. In this work, we use the following:

- Domain of the γ_j: $\gamma_j \in [0, (max(x) - min(x)) \cdot \gamma_0]$, with x being the time series and γ_0 being a parameter of the method.
- Domain of the c_j: The c_j are constrained to lie within the $[min(x), max(x)]$-interval.
- Domain and codification of the ω_j: In order to manage the restriction $\|\omega_j\| = 1$, ω_j is encoded using n-dimensional polar coordinates [5], so that $\omega_{ij} \in [0, \pi]$ for $i = 1, \ldots, (d-1)$, and $\omega_{dj} \in [0, 2\pi]$.

The NCSTAR algorithm works then as follows. At first, a linear model is fit to the time series, and the NCSTAR model is initialized with this first regime. Then, the method determines if the time series is linear, using a statistical test. If the result of the test suggests that the series is linear, the method exits, returning the initial linear model. Otherwise, the iterative building procedure is started. In iteration k, a $(k+1)$-regime NCSTAR model is built in the following way:

1. In the initial population of DE, an individual composed of the regimes of the current model with a new, randomly initialized regime, is included. The rest of the initial population is randomly generated according to a uniform distribution within the respective domains of the parameters.
2. The nonlinear parameters for all k transitions are computed by the optimization algorithm (the linear parameters have to be fitted in each evaluation of the error function).
3. The statistical test of linearity of the residuals is applied.
4. If the test indicates the need to add a new regime, return to step (1.). Otherwise, the method finishes.

5 Empirical Study and Analysis of the Results

An experimental study was carried out in order to assess the effectiveness of the method we propose. In this section, we discuss the time series data that has been used throughout the experiments, the algorithms that have been compared, their configuration parameters, and the results obtained.

5.1 Used Time Series

In this study, we used data from the competition NNGC1[1]. We used the series with the highest numbers of observations, i.e., the series that represent hourly, daily, and weekly observation recordings. The hourly recorded series represent airport arrival times of airplanes, and train arrival times at metro stations. The series recorded daily represent the amount of cars that passed certain tunnels at the respective days. The weekly recorded data are economic indicators taken from the oil industry, such as prices or import volumes.

The NCSTAR is a model for stationary time series (as linear AR and TAR models in general [18]). Furthermore, as seen in Section 4, if the time series is assumed to be linear due to the initial linearity test of the method, the iterative building procedure for the NCSTAR model is not used, but a linear model is built instead. So, we apply the Dickey-Fuller test [16] and the linearity test of the NCSTAR model to the time series and use only the series that show stationary and nonlinear behavior. In total, 15 series are used.

The series are partitioned for the experiments into a training and a test set in the following way: 20% of the data from the end of each series are withheld during model fitting, and used for testing afterwards. All series considered are available in the KEEL-dataset repository[2].

5.2 Algorithms and Parameters

The R programming language [15] was used to implement the methods and conduct the experiments. The implementation of the NCSTAR model is based

[1] http://www.neural-forecasting-competition.com/datasets
[2] http://sci2s.ugr.es/keel/timeseries.php

on code from the package `tsDyn` [10]. The BFGS algorithm, which is available in R via the function `optim` from the base package, is used instead of the Levenberg-Marquardt algorithm. In the following we call this implementation of the original algorithm `NCSTAR`.

Another version of the original algorithm, called `NCSTAR-BOX`, was also implemented. It uses the L-BFGS-B algorithm, which is a box-constrained version of the BFGS algorithm, and available through the same function `optim` (see Venables and Ripley [20]). The motivation for the use of this two algorithms is, that the results allow for studying possible effects of the parameter domains specified in Section 4.

The proposed algorithm, i.e., the version that employs the DE algorithm for optimization, is called `NCSTAR-DE`. We use an implementation of DE that is available in R from the package `RcppDE` (which is a reimplementation of the function `DEoptim` [14]).

The `NCSTAR` and `NCSTAR-BOX` algorithms were used with $N = 140$, and M =2,000. The $N = 140$ values of γ are chosen equidistant within the domain of γ as specified in Section 4, with $\gamma_0 = 20$ in all our experiments. The local search algorithm is stopped after a maximum of 20,000 iterations. Therefore, a total of approximately 300,000 evaluations of the fitness function are performed for every new regime that is added.

To make a fair comparison, the DE algorithms are used with the same number of evaluations of the error function. The population size is set to 30 individuals across all experiments, and 10,000 generations are calculated, which also produces a total of approx. 300,000 evaluations of the error function for each new regime. Furthermore, the DE is used with the parameters $F = 0.5$ and $CR = 0.5$.

In addition to comparing different NCSTAR models, a comparative study of the proposed method with other commonly used methods for time series prediction was carried out. Specifically, we considered ϵ support vector regression (SVR), a multilayer perceptron (MLP) trained with the BFGS algorithm, and a linear AR model. SVR is available in R via the package `e1071`, which implements a wrapper for the LIBSVM [7]. The MLP trained with the BFGS is available through the package `nnet` [20].

The parameters of the methods have been established with 30 artificial time series generated by an NCSTAR process. First, a parameter grid was defined empirically. Then, the set of parameters which yielded the best performance in all these series, with respect to the root mean squared error (RMSE) in the test set, was chosen. The parameters finally used for the experiments were cost = 10, gamma = 0.2, epsilon = 0.1 for the `SVR`, and size = 9, decay = 0.1, and maxit = 1,000 for `MLP`. The model `AR` has no free parameters.

5.3 Results Obtained and Statistical Analysis

Models were trained for the 15 time series, predictions were made on the respective test sets, and the RMSE was computed for each series. Table 1 shows the results obtained. The best result for each series is remarked in bold. As the series are normalized, comparing the RMSE of the different series is feasible.

Table 1. Experimental results: RMSE on the test set for the 15 time series

Time series	NCSTAR-DE	NCSTAR	NCSTAR-BOX	AR	MLP	SVR
TS-01	0.193	0.200	0.196	0.184	**0.181**	0.200
TS-02	0.538	0.535	0.538	**0.417**	0.535	0.661
TS-03	0.903	0.912	0.906	**0.813**	1.012	0.938
TS-04	**0.545**	0.617	0.692	0.855	0.551	0.586
TS-05	0.504	**0.493**	0.559	0.851	0.526	0.530
TS-06	0.455	0.486	0.461	0.426	0.463	**0.419**
TS-07	1.042	2.435	1.323	**0.967**	1.075	0.968
TS-08	**0.098**	0.103	0.120	0.472	0.145	0.159
TS-09	**0.176**	0.184	0.188	0.558	0.190	0.216
TS-10	0.120	**0.099**	0.142	0.460	0.155	0.171
TS-11	**0.142**	0.156	0.153	0.519	0.187	0.173
TS-12	**0.221**	0.333	0.236	0.639	0.284	0.320
TS-13	0.144	0.157	**0.132**	0.570	0.203	0.235
TS-14	**0.194**	0.335	0.261	0.579	0.274	0.255
TS-15	**0.153**	0.177	0.214	0.655	0.225	0.215
Mean	**0.362**	0.482	0.408	0.598	0.400	0.403

Table 2. Results of the nonparametric statistical tests

Method	Ranks	$p - value$ of Holm
NCSTAR-DE	1.87	–
NCSTAR-BOX	3.40	0.0383
NCSTAR	3.47	0.0383
MLP	3.80	0.0140
SVR	4.00	0.0072
AR	4.47	0.0007

Results highlight that `NCSTAR-DE` obtains the lowest average error. Nevertheless, considering only average results could lead us to erroneous conclusions because this measure may be affected by the influence of *outliers*. Due to this fact, we will accomplish statistical comparisons over multiple data sets based on non-parametric tests [11][3].

Table 2 presents the statistical analysis conducted. Specifically, we will focus on the use of the Friedman test, which detects significant differences at a significance level of $\alpha = 0.005$. This test computes the set of rankings that represent the effectiveness associated with each algorithm (second column). This table is ordered from the best to the worst ranking. In addition, the third column shows the adjusted p-value with the Holm's test. Note that `NCSTAR-DE` is established as the control algorithm because it has obtained the best (lowest) Friedman ranking. By using a level of significance $\alpha = 0.05$, `NCSTAR-DE` is significantly better than the rest of the methods.

[3] More information about these tests and other statistical procedures specifically designed for use in the field of Machine Learning can be found at the SCI2S thematic public website on *Statistical Inference in Computational Intelligence and Data Mining* `http://sci2s.ugr.es/sicidm/`

6 Conclusions

The NCSTAR models have an iterative building procedure and can be interpreted as fuzzy rule-based systems. Within the building procedure, a nonlinear optimization has to be performed in each iteration to adjust the model parameters. We propose the use of DE for this task: In each iteration, a statistical test for linearity is used to determine whether the procedure should terminate or add a new regime and readjust the whole system using an optimization method based on DE.

Within the NCSTAR algorithm, characterized as an interpretable model, the use of DE as a method to adjust its parameters demonstrated to result in significantly more accurate models than the combination of grid search and local search in the original algorithm scheme.

With the use of an evolutionary optimization algorithm for parameter adjustment, the NCSTAR model achieves model accuracy comparable to other standard time series prediction procedures. Therefore, with NCSTAR-DE, we obtain an algorithm capable of creating models for time series which are accurate and interpretable.

Because of the close relationship of the TAR models in general, and NCSTAR in particular, with fuzzy rule-based systems, it is easy to deduce that the proposed adjustment method presented in this paper could be applied successfully in the direct adjustment of fuzzy rule-based systems for time series modeling.

Acknowledgements. This work was supported in part by the Spanish Ministry of Science and Innovation (MICINN) under Project TIN-2009-14575. C. Bergmeir holds a scholarship from the Spanish Ministry of Education (MEC) of the "Programa de Formación del Profesorado Universitario (FPU)". I. Triguero holds an FPU scholarship from the University of Granada.

References

1. Aznarte, J.L., Benítez, J.M.: Equivalences between neural-autoregressive time series models and fuzzy systems. IEEE Transactions on Neural Networks 21(9), 1434–1444 (2010)
2. Aznarte, J.L., Manuel Benítez, J., Luis Castro, J.: Smooth transition autoregressive models and fuzzy rule-based systems: Functional equivalence and consequences. Fuzzy Sets and Systems 158(24), 2734–2745 (2007)
3. Aznarte, J.L., Medeiros, M.C., Benítez, J.M.: Linearity testing for fuzzy rule-based models. Fuzzy Sets and Systems 161(13), 1836–1851 (2010)
4. Aznarte, J.L., Medeiros, M.C., Benítez, J.M.: Testing for remaining autocorrelation of the residuals in the framework of fuzzy rule-based time series modelling. International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems 18(4), 371–387 (2010)
5. Blumenson, L.E.: A derivation of n-dimensional spherical coordinates. The American Mathematical Monthly 67(1), 63–66 (1960)
6. Box, G., Jenkins, G.: Time series analysis: Forecasting and control. Holden-Day (1970)

7. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
8. Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation 15(1), 4–31 (2011)
9. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Berlin (2003)
10. Di Narzo, A.F., Aznarte, J.L., Stigler, M.: tsDyn: Time series analysis based on dynamical systems theory, R package version 0.7 (2009), http://CRAN.R-Project.org/package=tsDyn
11. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
12. Medeiros, M.C., Veiga, A.: A flexible coefficient smooth transition time series model. IEEE Transactions on Neural Networks 16(1), 97–113 (2005)
13. Moré, J.J.: The Levenberg-Marquardt algorithm: Implementation and theory. In: Numerical Analysis: Proceedings of the Biennial Conference, pp. 104–116. Springer, Heidelberg (1978)
14. Mullen, K.M., Ardia, D., Gil, D.L., Windover, D., Cline, J.: Deoptim: An r package for global optimization by differential evolution. Journal of Statistical Software 40(6), 1–26 (2011)
15. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2009) ISBN 3-900051-07-0
16. Said, S.E., Dickey, D.A.: Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika 71, 599–607 (1984)
17. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
18. Tong, H.: Non-linear Time Series: a dynamical system approach. Clarendon Press, Oxford (1990)
19. Triguero, I., García, S., Herrera, F.: Ipade: Iterative prototype adjustment for nearest neighbor classification. IEEE Transactions on Neural Networks 21(12), 1984–1990 (2010)
20. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. Springer, Heidelberg (2002)

ReactGA – The Search Space Transformation for the Local Optimum Escaping

Radosław Ziembiński

Institute of Computing Science, Poznan University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland,
radoslaw.ziembinski@cs.put.poznan.pl

Abstract. This paper describes enhancement to genetic algorithms that allows them to escape from the local optima during the optimization. The proposed method relies on the search space transformation that helps in resumption of the search process. It allows to continue the optimization for the same population's size without the random probing of the local optimum's neighborhood. It also reduces a necessity for the genetic algorithm's restart with a differently distributed initial population. In the result, it can converge to the global optimum after a lower number of iterations. The proposed method is applicable to optimization problems described by any number of real–valued variables. The paper describes this method and presents results from the experimental evaluation that highlights properties of the proposed method.

Keywords: Evolutionary Optimization, Local Optimum Escaping.

1 Introduction

Stochastic methods of optimization have a long story of successful applications with many application in the engineering, chemistry, economy and entertainment. The main feature promoting them to these applications is ability of handling computationally complex problems often with non–linear relationships of parameters and optimized variables. These problems are often very difficult to solve with available algebraic methods without a significant simplification of their definition.

The inspiration for the proposed method comes from limitation of the plain genetic algorithm. More advanced stochastic methods like genetic algorithms manage the search process. They avoid unnecessary probings of the search space and have mechanism ensuring convergence. In the result, the search space is then narrowed to a tiny fraction of the original space. This approach would profit if this region has a global optimum. But such management introduces additional risk to the processing. In some cases the search space may become wrongly narrowed around a local optimum. This would lead to the premature end of the optimization process far from the best possible solution. The introduced new technique simplifies the search space in a way allowing to resume optimization from its dead–end at the local optimum.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 474–484, 2012.

If the optimization process stops at the local optimum then genetic algorithm's operators are potentially capable to resume the search process. But the successful resumption would need finding a solution that is better than the solutions from the population. Unfortunately, the existing population weights on solutions found in the local optimum's surround and the population converges quickly to the local optimum. The proposed method uses technique recognizable as complex fitness scaling. It transforms the search space and removes features of the search space that would form an obstacle to the resumed search process. Then, it explores so created transformed search space until better solutions than the best one was found. If the search process successfully has escaped from the local optimum then the algorithm resumes exploration for better solutions in the original search space.

This paper begins from a short description of related works. It introduces to the problem of the local optima escaping. Then, it describes the proposed concept and the enhanced genetic algorithm. The paper completes a presentation of some results from experimental study of the proposed method and conclusions.

2 Related Works

Genetic algorithms are mature stochastic methods inspired by the biological process of the evolution adapted as the general purpose optimization method e.g., in [4] and [6]. There are already some methods proposed to avoid or escape from the local optimum e.g., in [5] where many separate sub–populations probe the search space at different locations simultaneously. Another method proposed in [5] is fitness scaling. It relies on functional transformation of the search space what can help in effective avoidance of some local optima. Yet another method involves a spatial separation of solutions from the population e.g., a method from [8]. This approach may allow insertion of solutions from the neighborhood of the local optimum. However, it suffers from costly mutual comparisons of solutions from the population to keep up the dispersion leading to wasted iterations. Moreover, the choice of distance measure for the particular optimization problem is not trivial. The paper [7] proposes alternative Gray encoding for the solution phenotype. This method transforms the search space to another one with a lower number of local optima. This observation was an inspiration for a modification of the CHC genetic algorithm from [3] in [1]. The new algorithm repeatedly switches the representation to the Gray encoded if it encountered local optimum (e.g., when the population remains unchanged after some number of iterations). Some properties of the Gray encoding were critically reviewed e.g., [2] and following studies proposed more efficient methods of encoding e.g., [10] and [9].

From this perspective, the introduced method have following features:

– The proposed method dynamically transforms the search space on the base of the best solution. The transformed search space is rebuilt each time after the search process encounters the local optimum. It allows to use unmodified GA operators in the transformed search space.

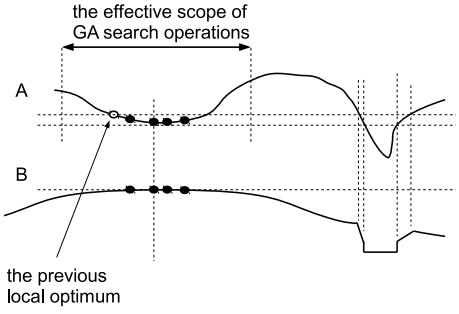


Fig. 1. The original search space and the transformed one

- This method creates the transformed search space with exposed regions where the search process can find better solutions from the original search space. The proposed method squeezes a larger part of the search space to a close neighborhood in the transformed space what increases the scope of the search process. The tuning of the transformation relies on information from the current population.
- The search in transformed search space stops if a better solution was found than the optimal one from the original space. Afterward the search resumes in the original search space with this new better solution.

3 The Method Description

Single dimensional case in Fig. 1 illustrates the proposed method of the local optimum escaping. Dots on the utility surface determined by the original goal function (part A) represent solutions. Clearly, the region on the left contains better solutions however it cannot be explored by the search method. Gradient of the search space and narrowed scope of genetic operators cause that the search process could not be guided to it. This results in the optimization suspension that could last for unlimited number of iterations.

The increasing of the population size or the increasing of the mutation scope may lead to the finding of a better solution located left from locally the best one. However, if the population has converged then this process would be difficult because similar solutions dominate the population. Therefore, the proposed method creates the transformed search space what illustrates Fig. 1 (part B). The transformed search space has removed all "obstacles" caused by regions with solutions worse than the best one. Instead, the introduced gradient is guiding the search process in the neighborhood. Additionally, the gradient has phased edges. Scope of the phasing depends on the previous local optimum evaluation. Hence, faster convergence causes more aggressive phasing in the transformed search phase. Finally, the transformed search space is exponentially squeezed in direction of the gradient's center. This allows to browse a wider neighborhood during the search. In the consequence, the locally optimal solution is the worst

solution in the transformed search space. Thus, the transformation inverts the search process to escape it from the local optimum.

The transformation occurs every time if no improvement to the population has happened after a specified number of iterations. Therefore, subsequently transformed search spaces are different from previous ones. New transformed search spaces contain a lower number of "holes" with better solutions inside them as the optimization progresses. Additionally, the offset position of the transformed search space follows the coordinates of the best solution from the population in the original search space. The search process returns to the original search space after it found in the transformed search space the first solution better from the locally optimal one. The transformation function for the minimization problem is following:

$$st(x, t_{\text{hard}}, t_{\text{soft}}, d) = \begin{cases} 0 \quad \text{if } x < t_{\text{hard}} \\ f_{\text{crv}}(d) \cdot \left\{1 - \frac{(x - t_{\text{soft}})}{(t_{\text{hard}} - t_{\text{soft}})}\right\} \\ \qquad\qquad \text{if } t_{\text{soft}} > x \geq t_{\text{hard}} \\ f_{\text{crv}}(d) \quad \text{if } x \geq t_{\text{soft}} \end{cases} \tag{1}$$

The x parameter of the function is the evaluation value from the original search space, d is the distance from the transformation center, t_{hard} and t_{soft} are thresholds dependent on the current and the previous local optima values. The function $f_{\text{crv}}(d) = 0.0001 + 0.4998 \cdot (cos(2 * PI * d) + 1)$ determines the curvature of the transformed space. Moreover, the exponential squeezing determine the distance transformation function:

$$sq(s, d) = \frac{1}{s^{-d}} - 1 \quad \text{for } d \geq 0 \wedge s > 1 \tag{2}$$

The following algorithm calculates the proposed transformation of the original search space to the transformed one:

```
function artificalGoalFunction
Input:        aSol: the evaluated solution
              rSol: the reference solution from
                                 the original search space
              params: the algorithm's parameters
              ctx: the algorithm's search context

  variables, distance, distance0 = convert(aSol, rSol, params)
  eval = evaluate(variables, params)
  factor = sq(params.scalingFactor, PI / (4 * params.phase))

  if distance > 0 and distance < factor then
      return st(eval, rSol.evaluation, ctx.softLimit,
                      ctx.phase * distance0)
  else return 1
```

The cost of the transformation grows linearly to the number of dimensions. It mostly includes the calculation of the distance from the solution to the center of the transformation. The following pseudo–code transforms artificial space coordinates to the original space ones:

```
Function convert
Input:        aSol: the evaluated solution
              rSol: the reference solution from the original search space
              params: the algorithm's parameters
  factor = 0
  // the distance to the transformation center
  distanceLnz = euclideanDistanceToTransformationCenter(aSol)

  if distanceLnz > 0 do
      factor = sq(params.scalingFactor, distanceLnz)
      scale = factor / distanceLnz
      for i in 1 .. n do // iterate for all dimensions
          variables[i] = scale * aSol.variables[i] + rSol.variables[i]
  else
      variables = rSol.variables // copy coordinates for the center

  return variables, factor, distanceLnz;
```

The *scalingFactor* context's parameter controls the exponential squeezing of the search space. Then, the *phase* parameter determines a border of the search neighborhood in the transformed search space. This parameter is cyclically updated if the search process could not find the solution to expand the neighborhood. The third parameter *softLimit* phases the edges of holes and evaluation of the previous locally optimal solution determines its value. The variables in the transformed search space differ from the variables in the original search space because of the squeezing. Thus, the population replaces the previous one after each switch between algorithms' modes if the squeezing is enabled.

4 ReactGA Algorithm

The ReactGA algorithm is basic genetic algorithm with a single population enhanced with the proposed local optima escaping method. The following pseudo–code describes the algorithm:

```
Function iterate
  if testIfPopulationDidNotChanged() then
      resetDeadlockCounters()
      switchAlgorithmsMode()
      if isArtificialMode() then
          ctx.phase = iteratePhase(ctx.phase)
          convertPopulationToArtificialSpace()
  else
      ctx.softLimit = getBestSolutionEvaluation()

  childSolution = null
  if not isArtificialMode() then
      childSolution = produceOffspringOriginal()
  else
      childSolution = produceOffspringArtificial()
```

```
    if testIfBetterSolutionFound(child) then
        switchAlgorithmsMode()
        convertPopulationToOriginalSpace()
  updateImprovementIterators()

  return childSolution
```

The algorithm works in two modes switched in the *testIfPopulationDidNotChanged* function. This procedure tests if the population has not changed after recent iterations before the mode would be switched. The *testIfBetterSolutionFound* function restores the original search space if a better solution was found. Both functions use counters managed by *updateImprovementIterators* function. Functions *produceOffspringOriginal* and *produceOffspringArtificial* produces offspring using a randomly selected operator. These functions maintain also the population size. Domains of the original search space and the transformed one differ because of the exponential squeezing. Thus, *convertPopulationToArtificialSpace* and *convertPopulationToOriginalSpace* procedures convert the populations between both "worlds". The *iteratePhase* function manages the scope of the search process in the transformed search space.

5 The Experimental Evaluation

The experimental evaluation was performed on a single objective optimization problem with a randomly parametrized goal function called Cave. This function defines a diversified multidimensional space illustrated in Fig. 2 resembling interconnected caves. It has following formula:

$$\begin{aligned} cave(V_n, \delta, P_{n,m}, E_{n,m}, S_m) = \\ (1 + 1.5 \cdot (e^{-|1.5 \cdot \delta - 0.75|} - e^{-|0.5 \cdot \delta - 0.25|})) \\ \cdot \sum_{i=1 \ldots m} S_i \cdot \frac{e^{-1.5 \cdot f_{\mathrm{dt}}(V_n, P_{n,m}, E_{n,m})}}{\sqrt{30 \cdot f_{\mathrm{dt}}(V_n, P_{n,m}, E_{n,m})}} \\ -1.6 \cdot (\cos(60 \cdot \Pi \cdot f_{\mathrm{dt}}(V_n, P_{n,m}, E_{n,m})) + 2) \end{aligned} \tag{3}$$

$$f_{\mathrm{dt}}(V_n, P_{n,m}, E_{n,m}) = \sum_{i=1 \ldots m} \sum_{j=1 \ldots n} (V_j - P_{j,i})^2 \cdot E_{j,i}$$

Matrices P, E and S parametrize the cave function, m determines the function complexity and n is the problem dimensionality. The distance from the search space center specifies δ. The following function generated random parameters of the Cave function for experiments:

```
Function generateRandomFunction
Input:          dimensionality: the problem's dimensionality
                attractors: attracting points number
                detractors: detracting points number
                forcesRadius: the influence of forces
  m = attractors + detractors
  for i in 1 .. m do
      C = randomInUnitHyperball()
```

```
    for j in 1 .. n do
        if i < detractors then
            P[j, i] = C[j] * 0.20 + 0.50
        else
            P[j, i] = C[j] * 0.40 + 0.50
        E[j, i] = random(1, 5)

    normalize(E)
    distance = euclideanDistance(P, spaceCenterCoords)
    S[i] = forcesRadius *
        (random(0, 1) + exp(-distance * 10) * (i mod 8))
    if i >= detractors then
        S[i] = -S[i]

return P, E, S
```

The parameters *attractors* and *detractors* define complexity of the goal function. If they are increasing then the search space is getting more rippled. The *forcesRadius* defines a scale of interferences.

The basic configuration of the Cave function in experiments had 80 attractors and 20 detractors and the *forcesRadius* parameter's value was 0.1. Variables' values of the feasible solution were in range from 0 to 1. The genetic algorithm's population size was 200 solutions. The initial populations for all algorithms runs had uniform distribution within hyper–box at coordinates 0.15–0.35. Reliable performance measurements required 32 repetitions of algorithms run for the same problem configuration but with different randomly generated goal functions and initial random set of solutions. Presented figures contain averages from these measurements.

Efficiency measurements were obtained in the experimental comparison of RactGA to PlainGA, PlainGA SRNS (Small Radius Neighborhood Search), PlainGA LRNS (Large Radius Neighborhood Search). The PlainGA is a simple GA with no protection against the local optima. But PlainGA SRNS and PlainGA LRNS algorithms randomly probe the neighborhood at the local optimum. The probing procedure replaces 75% of the population with random solutions from the neighborhood. This measure may resume the optimization in the case of the dead–end at the local optimum. The probing scope radius is 0.2 for PlainGA SRNS and 0.4 for PlainGA LRNS from the local optimum location. The condition that switches the probing procedure is the same as in ReactGA for switching on the transformation. It occurs if the worst solution from the population remains unchanged after 50 iterations.

The ReactGA algorithm processing illustrates Fig.2. It is an example of a two–dimensional search problem. Capital letters in top–left figure (representing the original search space) label subsequently encountered local optima. This figure shows convergence of the search process to the global optimum after escaping the local optima with the help of subsequent transformations. Additionally, the transformation results illustrate the search process finding its way to the global optimum.

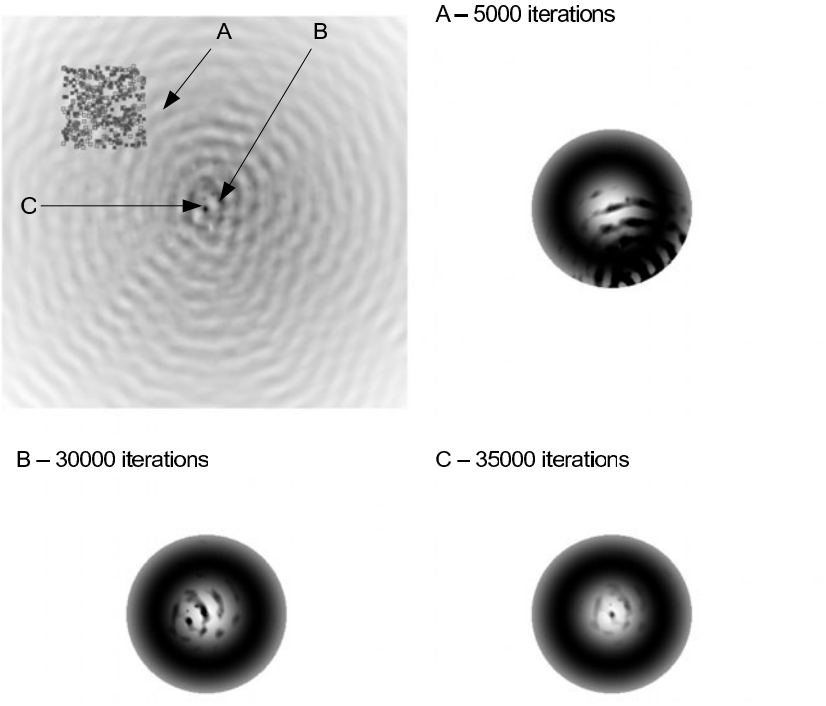


Fig. 2. The original 2–dimensional search space and transformed ones after 5000, 30000 and 35000 iterations (darker is better)

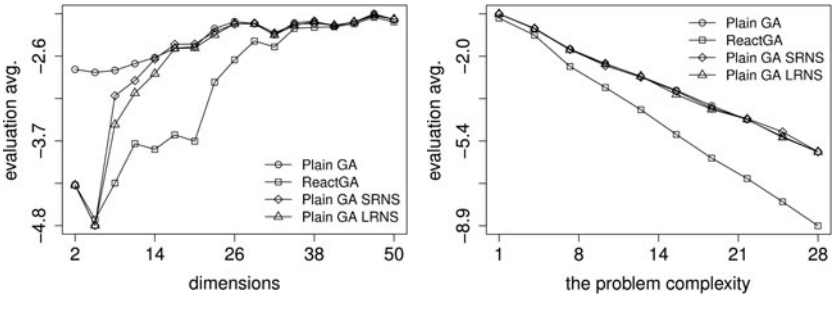


Fig. 3. The results after 200000 iterations for different dimensionality and complexity of the optimization problem (less is better)

The first experiment has been conducted to measure algorithms' performance if dimensionality of the problem is increasing. The left part of Fig. 3 contains results from this experiment. These results show that the ReactGA is more efficient than other algorithms for the problems with dimensionality ranging from 4 to 30. If the dimensionality is below 4 then the genetic algorithms with random probing have similar performance. The low dimensionality helps the random probing and it appears to be sufficiently dense to find some better solutions in the neighborhood. However, if the dimensionality increases then the limited population size cannot ensure sufficiently dense probing and these algorithms' performance decreases. On the other hand the highly multidimensional cases with 30 and more dimensions have relatively smooth and vast search spaces. Complexity of the search function is the same in this experiment thus it has the smeared distinctive features of the search space over more dimensions. In the consequence, all algorithms find solutions with almost equal performance.

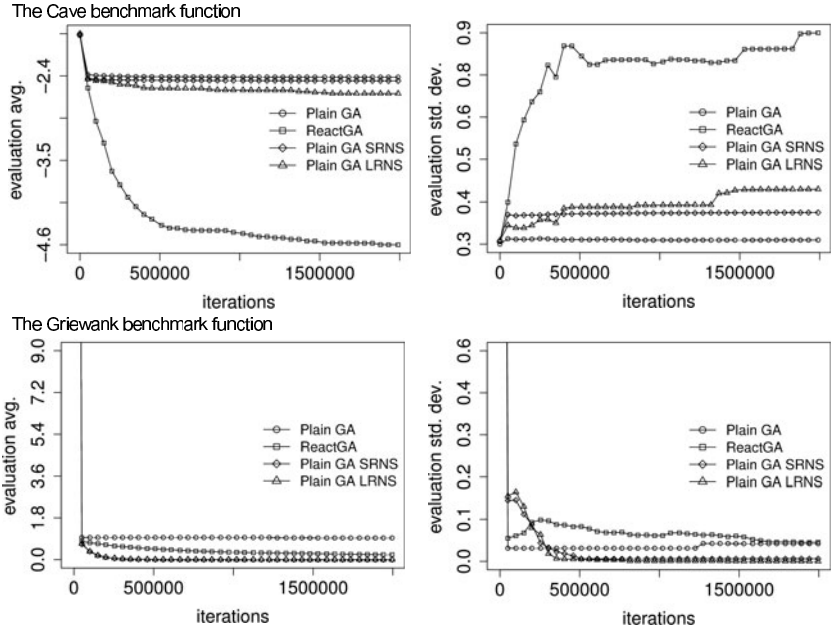


Fig. 4. The performance of algorithms through 2000000 iterations for 20–dimensional Cave and Griewank problems

The second experiment has been conducted to measure performance of the genetic algorithms if complexity of the search function is growing. Numbers of attractors and detractors were multiplied by an increasing factor presented on the horizontal axis. However, the proportion of attractors to detractors was always 8/2 and the Cave function was 20–dimensional. Right part of Fig. 3 contains the results from the experiment. It shows that the efficiency of ReactGA is

growing in relation to the results obtained from other algorithms if complexity of the problem is growing. This may suggest that the algorithm is relatively more efficient for diversified search spaces.

The final experiment was conducted to evaluate convergence of studied algorithms. Tested algorithms' performed 2 millions of iterations. A range of the Griewank function's parameters considered in the experiment was from -600 to 600. Fig. 4 illustrates results from this experiment. According to results, ReactGA outperforms other algorithms after a relatively small number of iterations. But for the Griewank function all algorithms almost equally converge to the middle niche. The mechanism for escaping from the local optima in ReactGA causes slower convergence in the fine structure of the Griewank function. Thus, it inefficiently wastes iterations on the neighborhood search. Instead, a precise local search would be required for that. Moreover, the greatest standard deviation characterizes results obtained from ReactGA. The scale effect of the lower goal function values and higher dispersion of the global optima in 32 computed cases partially caused these high values of standard deviations.

6 Conclusions

The proposed method enhances genetic algorithms allowing them to escape the search process from the local optima. The search space transformation technique is relatively costless in the implementation. Its application costs do not depend on the goal function complexity but on the problem dimensionality. This method can substitute some complex genetic operators developed for this purpose. Additionally, it uses operators already designed for the original search space without modifications. This feature could be particularly useful for some more advanced operators organizing the information in solutions.

Following enumeration should conclude some features of the proposed approach:

- The extended genetic algorithm relies on a single population with constant size. Even excessive convergence of the population would not prevent this method from the search process continuation (if it is feasible).
- The method removes all features from the search space that would be obstacles in the search process resumption. The transformation gradually reduces complexity of the search space. Thus, it can identify subsequently better local optima.
- Application of the transformation has a low complexity. It depends linearly on the dimensionality of the search problem.
- The presented method of the local optima escaping is applicable to the problem described by real–valued variables and the metric search space.

The proposed method can be used with different stochastic optimization methods. Particularly, the transformation does not change the optimization algorithm and it only modifies the goal function. However, the parametrization of the transformation has to be done carefully because the gradient introduced by the transformation may misguide more dynamic optimization methods like the particle

swarm optimization. This may result in a low efficiency of the transformed space exploration.

The experimental evaluation has shown that ReactGA is more efficient than compared algorithms with random probing around the local optimum. Convergence of the search process to the global optimum in ReactGA was faster than in compared competitors. Finally, the obtained results suggest that relative efficiency of ReactGA increases if the roughness of the search space increases. But very fine structures of the fitness landscape may require additional techniques to improve their detection.

Concluding, the experiments shown that the proposed method can be successfully applied to escape from the local optimum without increasing a size of the population.

References

1. Barbulescu, L., Watson, J.-P., Whitley, L.D.: Dynamic Representations and Escaping Local Optima: Improving Genetic Algorithms and Local Search. In: AAAI/IAAI, pp. 879–884 (2000)
2. Chakraborty, U.K., Janikow, C.Z.: An analysis of Gray versus binary encoding in genetic search. Inf. Sci., 253–269 (2003)
3. Eshelman, L.J.: The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In: FOGA, pp. 265–283. Elsevier Science Inc. (1990)
4. Fraser, A.S.: Simulation of genetic systems by automatic digital computers. Australian Journal of Biological Sciences, 484–491 (1957)
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, pp. 513–518. Addison-Wesley Longman Publishing Co., Inc. (1989)
6. Holland, J.: Adaptation in natural and artificial systems, pp. 484–491. University of Michigan Press (1975)
7. Mathias, K.E., Whitley, L.D.: Transforming the search space with Gray coding. In: Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 513–518. IEEE Service Center (1994)
8. Pál, K.F.: Selection Schemes with Spatial Isolation for Genetic Optimization. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 170–179. Springer, Heidelberg (1994)
9. Rothlauf, F.: Representations for Genetic and Evolutionary Algorithms, pp. 484–491. Springer-Verlag New York, Inc. (2006)
10. Weicker, K.: A binary encoding supporting both mutation and recombination. In: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part I, pp. 134–143. Springer, Heidelberg (2010)

PATMAP: Polyadenylation Site Identification from Next-Generation Sequencing Data

Xiaohui Wu[1], Meishuang Tang[2], Junfeng Yao[3], Shuiyuan Lin[2], Zhe Xiang[1], and Guoli Ji[1,*]

[1] Department of Automation, Xiamen University, Xiamen 361005, China
{xhuister,glji}@xmu.edu.cn, vipxiangzhe@163.com
[2] Modern Educational Technical and Practical Training Center, Xiamen University, Xiamen 361005, China
{tangms,sylin}@xmu.edu.cn
[3]Software School, Xiamen University, Xiamen 361005, China
yao0010@xmu.edu.cn

Abstract. Polyadenylation is an essential post-transcriptional processing step in the maturation of eukaryotic mRNA. The coming flood of next-generation sequencing (NGS) data creates new opportunities for intensive study of polyadenylation. We present an automated flow called PATMAP to identify polyadenylation sites (poly(A) sites) by integrating NGS data cleaning, processing, mapping, normalizing and clustering. The ambiguous region was introduced to parse the genome annotation by first. Then a series of Perl scripts were seamlessly integrated to iteratively map the single-end or paired-end sequences to the reference genome. After mapping, the poly(A) tags (PATs) at the same coordinate were grouped into one cleavage site, and the internal priming artifacts were removed. Finally, these cleavage sites from different samples were normalized by a MA-based method and clustered into poly(A) clusters (PACs) by empirical Bayesian method. The effectiveness of PATMAP was demonstrated by identifying thousands of reliable PACs from millions of NGS sequences in *Arabidopsis* and yeast.

Keywords: next-generation sequencing, empirical Bayesian, poly(A) site.

1 Introduction

Polyadenylation is a critical post-transcriptional processing step in the maturation of eukaryotic messenger RNA (mRNA) [1]. The location where the pre-mRNA is cleaved (also known as the poly(A) site) marks the end of mRNA transcript. Many eukaryotic genes possess two or more poly(A) sites [2-4], and thus are involved in alternative polyadenylation (APA). APA is a powerful pathway that entails the selection of alternate poly(A) sites in a pre-mRNA and leads to the production of multiple mature mRNA isoforms from the same gene [1].

* Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 485–496, 2012.

Given the clear importance of APA, it is required that we determine its scope and prevalence to further explore and understand the genome. Previous EST-based analyses showed that over 50% of genes in humans and ~30% of genes in mice [3] contained APA sites. In plants, EST analyses found that APA existed in ~50% of rice genes [2] and ~33% of *Chlamydomonas* genes [5]. The cost and efficiency of EST-based data for APA studies, however, significantly degrades the polyadenylation information derivable from the sequences and limits the number of poly(A) sites that can be found [6]. Quickly replacing conventional Sanger sequencing, high throughput next-generation sequencing (NGS) technologies have provided us the sequences in greater depth and coverage. The latest study using NGS data has shown that over 70% of *Arabidopsis* genes use APA sites [4], which is significantly higher than previous extent of 25% revealed by MPSS data [7]. The coming flood of NGS data creates new opportunities for the more comprehensive study of the polyadenylated transcriptome.

Till now, our understanding of the polyadenylation-related biological processes such as non-templated nucleotide addition [8] and micro-heterogeneity of cleavage sites [3] is limited, thus the accurate determination of poly(A) sites using NGS data is still challenging. Non-templated nucleotide addition before poly(A) tails is found in ~50% of mRNAs in *Arabidopsis* [8] and at both 3'- and 5'-ends of mRNAs in *Chlamydomonas* [9]. This issue, may affect the accurate determination of poly(A) sites, yet has not been explicitly addressed in most poly(A) studies to date. Micro-heterogeneity is another complicated biological process that a few nucleotides at the 3'-end of many transcripts map to the same loci in cleaving pre-mRNA before polyadenylation [4-6; 9; 10]. There are some attempts to minimize the likely impact of micro-heterogeneity, Tian *et al*. clustered the cleavage sites within 24 nt to get distinctive poly(A) sites, based on the statistical distribution of distances between adjacent cleavage sites [10]. Shen *et al*. used 30 nt based on the distance between poly(A) signal and poly(A) site [2]. In these studies, different cleavage sites from the same genes were clustered into distinctive groups. However, many cleavage sites from different transcripts of the same gene may be distributed closely on the boundaries of different genomic regions, thus the clustering approach solely based on 24- or 30-nt intervals is too coarse to allow scrutiny of the potential subtle differences. Therefore, a conservative approach in determining poly(A) sites should take into consideration the expression of individual cleavage sites and their gene structural categorization (i.e., locations in intron vs. exon).

We propose an automated flow (called PATMAP) to identify poly(A) sites from NGS data, by integrating NGS data cleaning, processing, mapping, normalizing and clustering into one pipeline. PATMAP is designed to be generic for different formats (e.g., Fasta, Fastq), or different types (e.g., single-end, paired-end) of NGS data. The high-quality and high-magnitude poly(A) data from PATMAP will empower individual researchers to study alternative transcript processing in greater depth.

2 Materials and Methods

2.1 Schema of PATMAP

The schema of PATMAP is shown in Fig. 1. The PATMAP pipeline was implemented by several Perl scripts, integrating some in-house tools as add-ons. ParseGFF.pl is a Perl script for parsing the annotation file in GFF format into the

required format. RunMapping.pl is a Perl script to seamlessly integrate a series of Perl scripts to map the single-end or paired-end sequences to the reference genome. Perl scripts including Polyseq2PA.pl, DivideANtag.pl, JoinTAN.pl, parseBwt2PNP.pl were implemented for the iterative mapping procedure. After the mapping, PAT2PA.pl was used to group the poly(A) tags (PATs) at the same coordinate into one cleavage sites, then RemoveIP.pl was used to remove the internal priming cleavage sites. We then used ClusterPA.pl to cluster the cleavage sites into poly(A) clusters (PACs). Finally, two Perl scripts AlterPA.pl and AlterPAC.pl were used to parse the files storing the cleavage sites and the files recording the PACs into the PAC database. Based on the PAC database, we could make further analysis of polyadenylation or APA, such as detecting differentially expressed PACs and categorizing PACs based on their locations.

2.2 Dataset

In this study, the NGS data were downloaded from NCBI SRA, including two wild type leaf datasets from paired-end sequencing in *Arabidopsis* [4] and 5 datasets from single-end sequencing and 13 paired-end datasets both in yeast [11]. The format (Fasta or Fastaq) or types (single-end or paired-end) of the sequences depend on the sequencing technology. Since single-end sequencing only generates the sequences from one end, the single-end sequence is easier to be processed than the paired-end sequences. Here we take a pair of sequences from the paired-end sequencing [4] for example (see Fig. 2). Because the sequence reads have fixed length (75 nt here), there are variable length of 5' and/or 3' sequencing adapters in the front or at the end of each sequence read. Each sequencing run generates two files in Fastaq format, for example, PAT1.fastq and PAT2.fastq. PAT1 is the sequence with a poly(T) stretch at its 5' end and PAT2 is the sequence with or without a poly(A) tail at its 3' end.

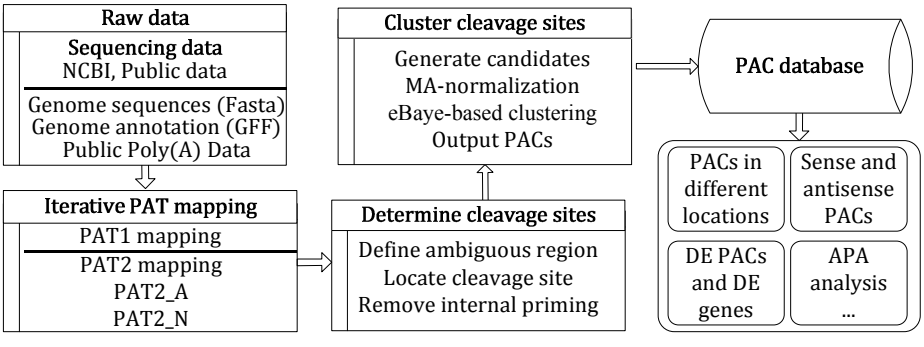


Fig. 1. Schema of PATMAP

2.3 Iterative PAT Mapping

The flow chart of the iterative PAT mapping is shown in Fig. 3. For sequences from single-end sequencing, the flow for PAT1 is sufficient for mapping, while if the sequences are from paired-end sequencing, then both the PAT1 mapping and PAT2 mapping should be carried out to get candidate loci. In PAT1 mapping flow, PAT1

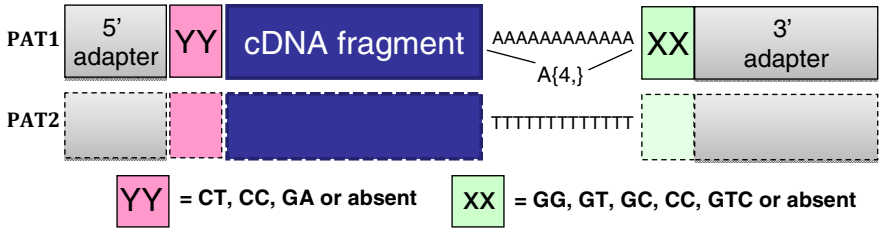


Fig. 2. PAT1 and PAT2 from paired-end sequencing. YY and XX denote the barcodes for separating the sequences of different samples from the same lane.

with the terminal bar code and a run of 8 or more consecutive Ts are identified and the sub-sequence after the poly(T) is also trimmed. The sequence with length shorter than 25 nt after trimming is discarded. Otherwise, it is remained for further mapping. Since a variable length of adaptor is expected in the remaining sub-sequence (Fig. 2), an iterative process is implemented which is designated to minimize the impact of non-templated nucleotide addition. Bowtie [12], an ultrafast short read aligner, is used in our pipeline for short sequence mapping. Only unique hit and at most two nt mismatches are allowed for a perfect match. If there is no perfect match for this sequence, then its 3'-most two nts are trimmed for mapping again. This process is repeated until the sequence is shorter than 25 nt or multiple hits are found.

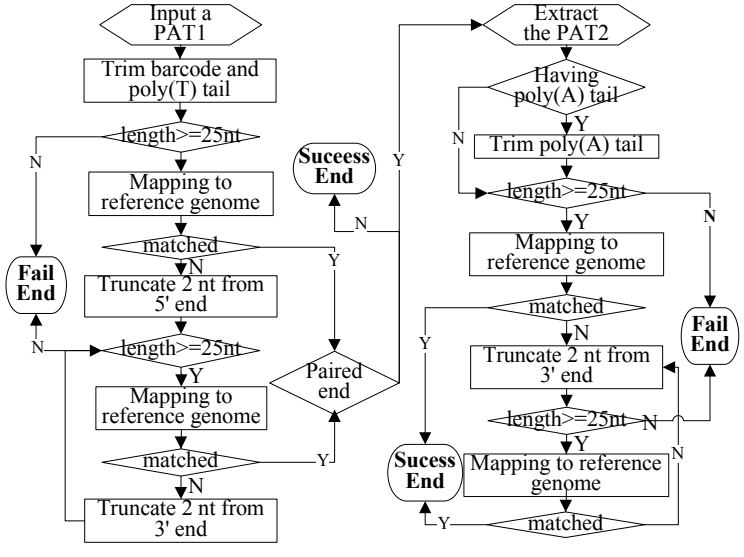


Fig. 3. PAT mapping for PAT1 and PAT2

If the raw data is from paired-end sequencing, then the paired-end partners (PAT2) to the mapped PAT1 are then filtered and divided into two groups: PAT2_A and PAT2_N. PAT2_A stores the sequences with at least 8 consecutive As (i.e., poly(A) tail). PAT2_N contains sequences without such poly(A) stretch. The sequences in

PAT2_A are trimmed off the poly(A) tail. Both sequences in PAT2_A and PAT2_N are then mapped to the genome using the same iterative process as PAT1 mapping to recover as many authentic PATs as possible.

2.4 Determination of Cleavage Sites

After PAT mapping, the coordinates of the mapped sequences are recorded and related with genome annotation to define their located genomic regions or intergenic regions. Unlike previous study where only the first transcript of the gene was considered, here the ambiguous region is introduced in the processing of the genome annotation file. Since there may be several transcripts corresponding to the same gene in the annotation file, the ambiguous region is defined as the common region shared by all transcripts of the same gene. Ambiguous region will make the genome annotation on which the further analyses are based more complete, thus enable the more specific evaluation of the APA prevalence.

Using the refined genome annotation, the mapped sequences are then associated with their genomic or intergenic regions. If the sequences are not from paired-end sequencing, then the coordinates of PAT1 are the final candidate cleavage sites. If they are from paired-end sequencing, then the candidate cleavage sites are determined using the following criteria: both PAT and PAT2 need to map to the same gene, or to the same chromosome on the same strands (either + or -) and within a distance less than 1000 nt. After mapping, the candidate cleavage sites that represented possible internal priming by reverse transcriptase are discarded using a strategy similar to that described in [3].

Fig. 4 shows some possible cases of paired-end sequences. If PAT2 is of type PAT2_A, then the rest of the PAT2 trimmed off poly(A) tail and the rest of the PAT1 trimmed off poly(T) stretch will be mapped to the same locus which is the cleavage site (Fig. 4A). If PAT2 is of type PAT2_N, then PAT2 and PAT1 will be mapped to different loci but within a certain distance (Fig. 4B). Interestingly, there is a stretch of A in the PAT2 in Fig. 4B, while this A stretch is not a real poly(A) tail of mRNA but from the reference genome. This result also indicates that our mapping procedure has the ability to differentiate the real poly(A) tail from the mere A stretch. Fig. 4C shows a case of internal priming. Though the PAT1 can be mapped to the genome, there is a stretch of T (or the complementary As) around the candidate cleavage site. This T stretch is actually from the reference genome rather than the mRNA, therefore this cleavage site is not real and is discarded.

Fig. 4. Cases of paired-end sequences. (A) PAT1 has poly(A) tail, and PAT1 and PAT2 map to the same locus; (B) PAT1 and PAT2 map to different loci; (C) Case of internal priming.

2.5 Clustering of Micro-heterogeneic Cleavage Sites

Here we develop an empirical Bayesian based method to cluster cleavage sites into PACs. First, the potential clusters of cleavage sites are obtained by iterative grouping the cleavage sites within a certain distance. Then, the likelihood that the number of PATs in each cluster is similar to background is calculated by empirical Bayesian method (eBayes) [13]. The higher the likelihood of the cluster is, the higher similarity between the cluster and the background is. We then rank all the candidate clusters in order of the likelihood and the cluster whose likelihood is higher than a cut-off is discarded.

To calculate the likelihood by eBayes, first, the prior parameters on the data are estimated using Poisson-Gamma method [13]. The discrete data from a set of sequencing or other NGS experiments is arranged in a matrix such that each column describes a sample and each row describes the clusters for which counts exists. Let $S = \{S_1, ..., S_n\}$ denote a set of *n* samples and $(t_{1\alpha}, ..., t_{n\alpha})$ denote the number of PATs for a cluster α from *S*. $F = (f_1, ..., f_n)$ is the normalization factor for *S*. Then each cluster can be represented as

$$C_\alpha = \{t_{1\alpha} f_1, ..., t_{n\alpha} f_n\}$$

Normalization is required when there are more than one sample (n>1). We propose a method based on MA-plot [14] to estimate the normalization factors. In a MA-plot, y-axis *M* is the Cy5/Cy3 (or red (R) /green (G)) intensity ratio, and x-axis *A* is the average intensity. *M* and *A* are defined by the following equations:

$$M = \log_2 R - \log_2 G;\ A = (\log_2 R + \log_2 G)/2 \tag{1}$$

For count data, we calculate *M* and *A* for the cluster α in sample S_i and S_j as

$$\begin{aligned} M_\alpha &= \log_2(t_{\alpha i} / \textstyle\sum_\alpha t_{\alpha i}) - \log 2(t_{\alpha j} / \textstyle\sum_\alpha t_{\alpha j}) \\ A_\alpha &= (\log_2(t_{\alpha i} / \textstyle\sum_\alpha t_{\alpha i}) + \log 2(t_{\alpha j} / \textstyle\sum_\alpha t_{\alpha j}))/2 \end{aligned} \tag{2}$$

To normalize the sample, first the clusters with no expression in S_i or S_j are removed to avoid zero item in log calculation. Then the clusters with extremely high or low number of PATs are also filtered out to ensure that these clusters have no impact on the whole library. Set the lower limit of *A* and *M* are A_0 and M_0, respectively, then the number of clusters with low number of PATs is calculated as:

$$N_{A0} = \underset{\alpha}{count}(A_\alpha < A_0);\ \ N_{M0} = \underset{\alpha}{count}(M_\alpha < M_0) \tag{3}$$

Here we set the number of clusters with high number of PATs equal to those with low number of PATs. The *M* and *A* values are sorted and these clusters are removed from the library. Finally the observed *M* values are summarized as the normalization factor.

To calculate likelihood, given a model *O*, its posterior probability given C_α is

$$P(O \mid C_\alpha) = P(C_\alpha \mid O)P(O) / P(C_\alpha) \tag{4}$$

There are three items required in Eqn. 4. $P(C_\alpha \mid O)$ can be calculated by

$$P(C_\alpha \mid O) = \int P(C_\alpha \mid K, O)P(K \mid O)dK \tag{5}$$

Studies have shown that Poisson seems appropriate for technical replicates [15], thus here we assume that the data is Poisson distributed and the parameters are Gamma distributed for estimating the parameters distribution *K* and calculating $P(C_\alpha \mid O)$. The second item $P(O)$ is the prior probability of the model, which may not be easily provided in many cases. We adopt a method for estimating proportions of differentially expressed genes [16] to estimate these priors. In belief, an initial value *p* for the prior probability is set by first. Then the posterior probability for each cluster can be estimated. We then update the prior probability by this posterior probability. This step is iterated until convergence and the estimate of the prior probability p^* can be acquired. The third item $P(C_\alpha)$ is a scaling factor, which is estimated by summing over all possible models *O*, given the estimated priors $P^*(O)$.

3 Results

3.1 Poly(A) Sites from *Arabidopsis*

There were more than 14 million sequence reads of type PAT1 (or PAT2) in the raw datasets of *Arabidopsis*. We compared the PACs obtained from PATMAP with those from previous study [4]. There were total 48,821 PACs from PATMAP, while the PAC number was higher in previous study (55,505). To find where these PACs were enriched, we cataloged PACs based on their localization. As shown in Fig. 5A, there were less PACs in 3' UTR by PATMAP, which was probably due to the use of different clustering strategies. In previous study, the cleavage sites within 24 nt were grouped as the same PAC in a simple manner, which might result in inaccurate clustering of close cleavage sites and lead to higher number of clusters. In contrast, we adopted eBayes to cluster cleavage sites, so that each cluster would be clearly distinguished from its background level. To further examine the impact of different clustering methods on the final clustering result, we compared the distributions of the range (or length) of PACs between PATMAP and previous study. As shown in Fig. 5B, there is a spike in the distribution of PAC length in previous study, which is due to the 24 nt parameter. It is clear that PATMAP can identify more reasonable PACs.

In addition to the advantage of the clustering strategy, PATMAP also distinguishes the PACs in ambiguous regions (MISC column in Fig. 5A), which can better characterize PACs with the same loci but located in different regions of different transcripts from the same gene. These PACs in ambiguous regions were probably classified as

the 3' UTR PACs in previous study, which might be another reason for the higher number of 3' UTR PAC in previous study.

To characterize better the PACs in ambiguous region, we identified 345 PACs with number of PAT higher than 10, covering 290 genes. Fig. 6 shows one of such PACs for gene AT1G49300. There are two transcripts of gene AT1G49300 and two ambiguous regions are detected. From the left box of Fig. 6, it is apparent that there is only one PAC whose dominant cleavage site has more than 150 PATs. However, if the clustering strategy from previous study was used, then these cleavage sites would be arbitrarily divided into two PACs, which would probably lead to misevaluation of APA prevalence.

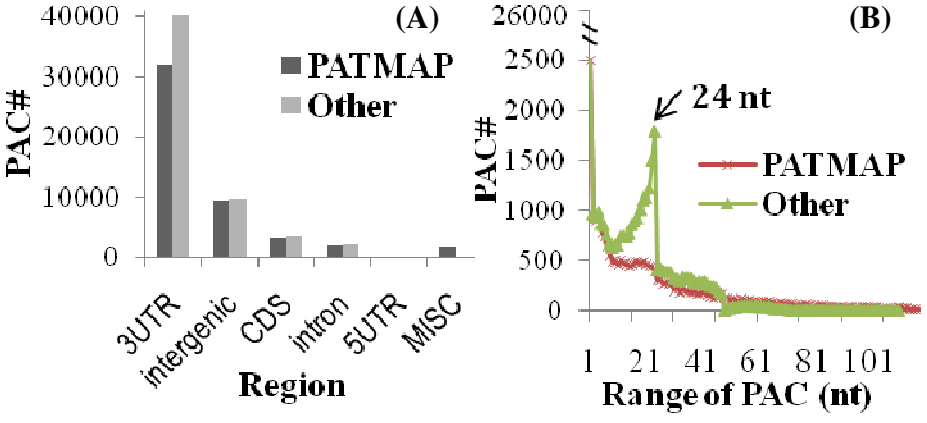


Fig. 5. Comparison between PATMAP and previous study. (A) Distributions of PACs. (B) Distributions of the range of PACs. MISC denotes the ambiguous region. Legend "Other" represents the result from previous study.

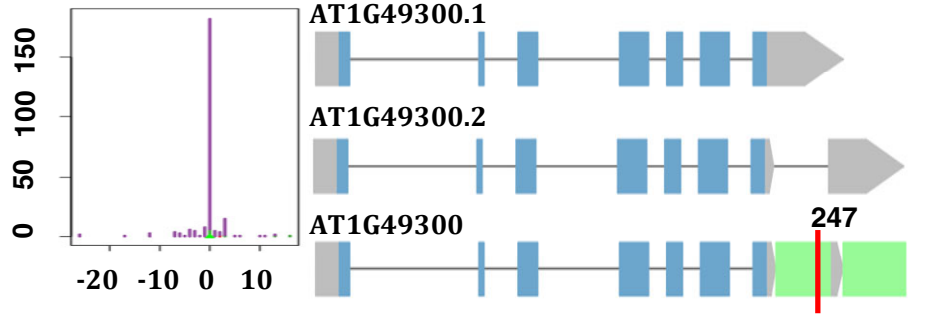


Fig. 6. An example of PAC in ambiguous region for gene AT1G49300. The blue box is CDS, grey box represents UTR, and the ambiguous region is colored in green. The PAC is marked by the red vertical line, and the number of PATs is shown on the top. The PAT distribution is plotted in the left frame, where x-axis is the distance from other cleavage site to the dominant one (zero) and y-axis is the number of PATs of each cleavage site in the PAC.

3.2 Poly(A) Sites from Yeast

In addition to *Arabidopsis*, we also identified poly(A) sites from several NGS datasets in yeast using PATMAP. Surprisingly, although there were more than 200 million raw sequences from these datasets, less than 0.5 million PATs had poly(T) tail. It was probable that these sequencing protocols did not focus on polyadenylation analysis, thus only a very small part of the polyadenylated transcripts were sequenced. Even so, our PATMAP was able to extract a moderate number of PACs from these large amounts of NGS data. Finally, total 15,666 PACs were identified by PATMAP, which was significantly higher than previous study where only 1352 unique poly(A) sites were found from ESTs [17].

To validate whether the identified PACs were authentic, the profile of the single-nucleotide base compositions surrounding PACs was generated. Such characterizations was very useful in the validation of poly(A) sites, especially when there were not so many data from other platforms available. Jan *et al*. proposed a method by mixing the random sequences from single UTRs with the true UTRs in varying proportions to estimate the fraction of false positives [18]. Here, we also generated the single nucleotide profile of Yeast poly(A) sites. The profile of the yeast PACs from PATMAP as shown in Fig. 7 was similar to the profile from previous study [17], even though the numbers of PACs that contributed to each plot varied by more than a factor of ten (between 15,666 and 1352 PACs).

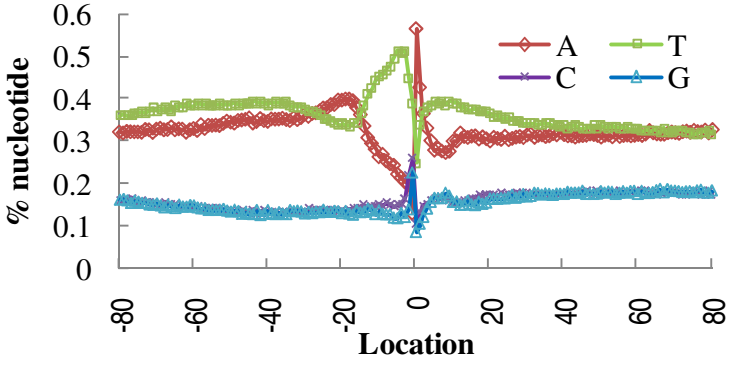


Fig. 7. Nucleotide composition of PACs in yeast identified by PATMAP

4 Conclusions

In the last few years, NGS has provided us with significantly better data for even well-studied, well-annotated model species like human and *Arabidopsis*, while avoiding many of the pitfalls of traditional Sanger sequencing methods. Currently, there are many cDNA-genome mapping programs for NGS mapping such as Bowtie [19] and GMAP [20], however, these tools typically generate the cDNA or DNA sequence mapping result rather than the poly(A)-related information. Though using NGS data,

studies have shown the extensive APA in different species like *Arabidopsis*, human, there is no generic tool or pipeline for identifying poly(A) sites from different kinds of NGS data. The coming flood of whole genome transcriptome data demands new bioinformatics tools that can exploit new information and identify poly(A) sites for more comprehensive polyadenylation study.

We present an automated flow, PATMAP, that can identify poly(A) sites from high throughput NGS data. The ambiguous region, MA-plot and empirical Bayesian method were incorporated in PATMAP for more accurate mapping of PATs and clustering of cleavage sites. Especially, the clustering method embedded in PATMAP was significantly better than the iterative clustering method used in previous study [4], in that the likelihood of the background levels was considered. Through our new bioinformatics pipeline of PATMAP, we acquired an unprecedented quantity of newly-discovered poly(A) sites in *Arabidopsis* and yeast. Though some NGS protocols are not designed for the study of polyadenylation, for example, the yeast datasets used here, PATMAP can still identify much more poly(A) sites from the NGS data than from previous ESTs. More importantly, the PACs identified by PATMAP are associated with the information about the expression (i.e., number of PATs), which enables us to analyze the poly(A) choice under different conditions (e.g., during development, in response to stress, or in different ecotypes). The poly(A) dataset generated by PATMAP will present a unique opportunity to study polyadenylation, APA, and 3' end processing regulation systematically and comprehensively.

PATMAP is designed to have flexibility in its methodology for NGS sequences mapping and cleavage sites clustering. Though PATMAP targets the NGS data, it is also applicable in the poly(A) site identification from ESTs due to the similarity between the EST and single-end sequence. The current version of PATMAP provides users with the capability to perform single-end or paired-end mapping using Bowtie and to cluster the cleavage sites based on empirical Bayesian method with different parameter settings specific to the characteristics of the sequences. With the speedy development of machine intelligence in the last few decades [21-23], many hybrid artificial intelligence systems have been designed and implemented for the application in biology field to address biological questions. In the future, we should be able to incorporate other effective hybrid artificial intelligence methods for better clustering of cleavage sites. Moreover, we are working hard on integrating other alignment programs like GMAP, so that users can compare NGS mappings using two different programs side-by-side. Efforts are also underway to incorporate the downstream analyses into PATMAP such as detection of differentially expression poly(A) sites, analyses of APA.

Acknowledgments. This project was funded by funds from the National Natural Science Foundation of China (No. 61174161), the specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20090121110022), the Fundamental Research Funds for the Central Universities of Xiamen University (Nos. 2011121047, 201112G018 and CXB2011035), the Key Research Project of Fujian Province of China (No. 2009H0044) and Xiamen University National 211 3rd Period Project of China (No. 0630-E72000).

References

1. Xing, D., Li, Q.Q.: Alternative Polyadenylation and Gene Expression Regulation in Plants. Wiley Interdisciplinary Reviews: RNA 2, 445–458 (2010)
2. Shen, Y., Ji, G., Haas, B.J., Wu, X., Zheng, J., Reese, G.J., Li, Q.Q.: Genome Level Analysis of Rice mRNA 3'-End Processing Signals and Alternative Polyadenylation. Nucleic Acids Res. 36, 3150–3161 (2008)
3. Tian, B., Hu, J., Zhang, H.B., Lutz, C.S.: A Large-Scale Analysis of mRNA Polyadenylation of Human and Mouse Genes. Nucleic Acids Res. 33, 201–212 (2005)
4. Wu, X., Liu, M., Downie, B., Liang, C., Ji, G., Li, Q.Q., Hunt, A.G.: Genome-Wide Landscape of Polyadenylation in Arabidopsis Provides Evidence for Extensive Alternative Polyadenylation. Proc. Natl. Acad. Sci. USA. 108, 12533–12538 (2011)
5. Shen, Y., Liu, Y., Liu, L., Liang, C., Li, Q.Q.: Unique Features of Nuclear mRNA Poly(a) Signals and Alternative Polvadenylation in Chlamydomonas Reinhardtii. Genetics 179, 167–176 (2008)
6. Shen, Y., Venu, R.C., Nobuta, K., Wu, X., Notibala, V., Demirci, C., Meyers, B.C., Wang, G.-L., Ji, G., Li, Q.Q.: Transcriptome Dynamics through Alternative Polyadenylation in Developmental and Environmental Responses in Plants Revealed by Deep Sequencing. Genome Res. 21, 1478–1486 (2011)
7. Meyers, B.C., Vu, T.H., Tej, S.S., Ghazal, H., Matvienko, M., Agrawal, V., Ning, J.C., Haudenschild, C.D.: Analysis of the Transcriptional Complexity of Arabidopsis Thaliana by Massively Parallel Signature Sequencing. Nat. Biotechnol. 22, 1006–1011 (2004)
8. Jin, Y., Bian, T.: Nontemplated Nucleotide Addition Prior to Polyadenylation: A Comparison of Arabidopsis cDNA and Genomic Sequences. RNA 10, 1695–1697 (2004)
9. Liang, C., Liu, Y.S., Liu, L., Davis, A.C., Shen, Y.J., Li, Q.Q.: Expressed Sequence Tags with cDNA Termini: Previously Overlooked Resources for Gene Annotation and Transcriptome Exploration in Chlamydomonas Reinhardtii. Genetics 179, 83–93 (2008)
10. Tian, B., Pan, Z.H., Lee, J.Y.: Widespread mRNA Polyadenylation Events in Introns Indicate Dynamic Interplay between Polyadenylation and Splicing. Genome Res. 17, 156–165 (2007)
11. Levin, J.Z., Yassour, M., Adiconis, X., Nusbaum, C., Thompson, D.A., Friedman, N., Gnirke, A., Regev, A.: Comprehensive Comparative Analysis of Strand-Specific Rna Sequencing Methods. Nat. Methods. 7, 709–767 (2010)
12. Langmead, B., Trapnell, C., Pop, M., Salzberg, S.L.: Ultrafast and Memory-Efficient Alignment of Short DNA Sequences to the Human Genome. Genome Biol. 10 (2009)
13. Hardcastle, T.J., Kelly, K.A.: Bayseq: Empirical Bayesian Methods for Identifying Differential Expression in Sequence Count Data. BMC Bioinformatics 11 (2010)
14. Dudoit, S., Yang, Y.H., Callow, M.J., Speed, T.P.: Statistical Methods for Identifying Differentially Expressed Genes in Replicated cDNA Microarray Experiments. Statistica Sinica 12, 111–139 (2002)
15. Bullard, J.H., Purdom, E., Hansen, K.D., Dudoit, S.: Evaluation of Statistical Methods for Normalization and Differential Expression in mRNA-Seq Experiments. BMC Bioinformatics 11 (2010)
16. Smyth, G.K.: Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. Statistical Applications in Genetics and Molecular Biology 3, article3 (2004)
17. Graber, J.H., Cantor, C.R., Mohr, S.C., Smith, T.F.: Genomic Detection of New Yeast Pre-mRNA 3 '-End-Processing Signals. Nucleic Acids Res. 27, 888–894 (1999)

18. Jan, C.H., Friedman, R.C., Ruby, J.G., Bartel, D.P.: Formation, Regulation and Evolution of Caenorhabditis Elegans 3'utrs. Nature 469, 97–101 (2011)
19. Lee, A., Hansen, K.D., Bullard, J., Dudoit, S., Sherlock, G.: Novel Low Abundance and Transient Rnas in Yeast Revealed by Tiling Microarrays and Ultra High-Throughput Sequencing Are Not Conserved across Closely Related Yeast Species. PLoS Genet. 4, e1000299 (2008)
20. Wu, T.D., Watanabe, C.K.: Gmap: A Genomic Mapping and Alignment Program for mRNA and EST Sequences. Bioinformatics 21, 1859–1875 (2005)
21. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. Neurocomputing 72, 2729–2730 (2009)
22. Garcia, S., Fernandez, A., Luengo, J., Herrera, F.: Advanced Nonparametric Tests for Multiple Comparisons in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power. Information Sciences 180, 2044–2064 (2010)
23. Corchado, E., Graña, M., Woźniak, M.: New Trends and Applications on Hybrid Artificial Intelligence Systems. Neurocomputing 75, 61–63 (2012)

How to Reduce Dimension while Improving Performance

Abdelghani Harrag[1,*], D. Saigaa[1], A. Bouchelaghem[1], M. Drif[1], S. Zeghlache[1], and N. Harrag[2]

[1] Department of Electronics, Faculty of Technology
University Mohamed Boudiaf Msila
BP 166 Ichbilia 28000 Algeria
abdelghani.harrag@gmail.com
[2] Department of Informatics, Faculty of Sciences, University Ferhat Abbas Setif,
El-Baz 19000 Setif Algeria

Abstract. This paper addresses the feature subset selection for an automatic Arabic speaker recognition system. An effective algorithm based on genetic algorithm is proposed for discovering the best feature combinations using feature reduction and recognition error rate as performance measure. Experimentation is carried out using QSDAS corpora. The results of experiments indicate that, with the optimized feature subset, the performance of the system is improved. Moreover, the speed of recognition is significantly increased, number of features is reduced over 60% which consequently decrease the complexity of our ASR system

Keywords: genetic algorithm; feature selection; speaker recognition.

1 Introduction

The speech signal is rich in information and redundancy. The redundancy is robust against background noise, distortion and damage suffered by the voice signal. This richness expresses the informations that are simultaneously conveyed by the message linguistic context, the anatomical features, the state and the socio-cultural constraints of the speaker.

Speech signals contain a huge amount of information and can be described as having a number of different levels of information. At the top level, we have lexical and syntactic features, below that are prosodic features, further below these are phonetic features, and at the most basic level we have low-level acoustic features, which generally give information on the system that creates the sound, such as the speakers' vocal tract. Information solely about how the sound is produced (from low-level acoustic features) should give enough information to identify accurately a speaker, as this is naturally speaker dependent and independent of text [1].

Low-level acoustic features also contain some redundant features, which can be eliminated using Feature Selection (FS) techniques. The objective of feature selection

* Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 497–508, 2012.

is to simplify a dataset by reducing its dimensionality and identifying relevant underlying features without sacrificing predictive accuracy. By doing that, it also reduces redundancy in the information provided by the selected features [2]. In real world problems, feature selection is a must due to the abundance of noisy, irrelevant or misleading features. Selected features should have high inter-class variance and low intra-class variability. Ideally, they should also be as independent of each other as possible in order to minimize redundancy.

Feature selection is extensive and it spreads throughout many fields, including signal processing [3], face recognition [4], text categorization [5], data mining and pattern recognition [6]. Among many methods that are proposed for feature selection, population based optimization techniques [7][8] such as genetic algorithm have attracted a lot of attention. These methods attempt to achieve better solutions by application of knowledge from previous iterations. Genetic algorithms are optimization techniques based on the mechanism of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space prompting to use them. Because of their advantages, recently, GAs have been used as a tool for feature selection in data mining [9].

In this paper, we propose a GA-based algorithm for feature selection in VQ-based Arabic Speaker Recognition (ASR) system. We apply it to feature vectors containing Mel-Frequency Cepstral Coefficients (MFCCs), their first and second derivative. Then, feature vectors are applied to a VQ model followed by K-Nearest-Neighbor (KNN) classifier used to measure the performance of selected feature vector based on recognition rate and selected feature vector size. The rest of this paper is organized as follows. Section 2 presents the taxonomy of ASR systems. Genetic algorithms are described in Section 3. Section 4 reports discussion of the results obtained. The conclusion and future works are offered in the last section.

2 Automatic Speaker Recognition System

Automatic speaker recognition refers to recognizing persons from their voice. No two individuals sound identical because their vocal tract shapes, larynx sizes, and other parts of their voice production organs are different. In addition to these physical differences, each speaker has his or her characteristic manner of speaking, including the use of a particular accent, rhythm, intonation style, pronunciation pattern, choice of vocabulary and so on. State-of-the-art speaker recognition systems use a number of these features in parallel, attempting to cover these different aspects and employing them in a complementary way to achieve more accurate recognition.

Automatic speaker recognition systems are generally divided into two categories (Figure 1), namely: automatic speaker identification systems which are designed to answer the question “who is the speaker?” or automatic speaker verification systems that aim to answer the question “is the speaker who they claim to be?”.

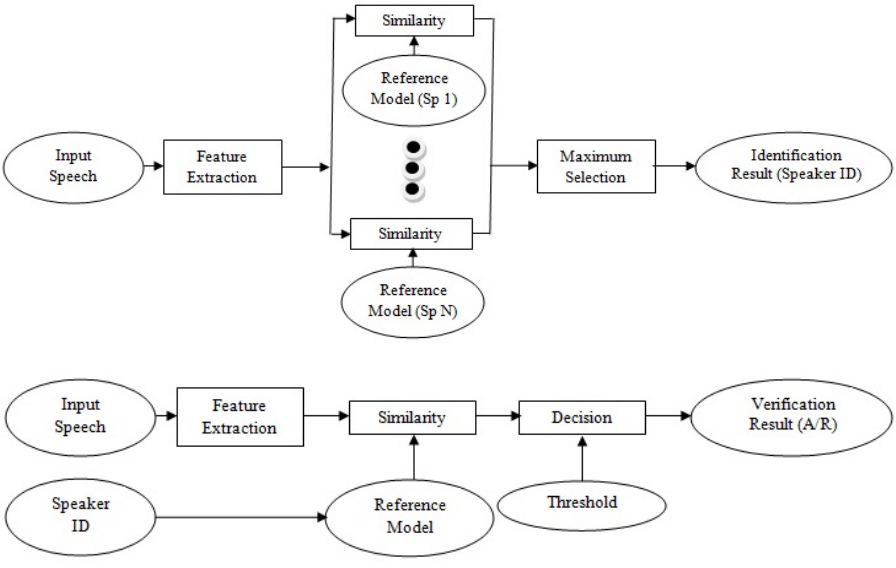


Fig. 1. Structures of ASR system: Identification (Top) and Verification (Bottom)

On other hand, speaker recognition can be classified into text-dependent and text-independent applications. When the same text is used for both training and testing, the system is called to be text-dependent while for text-independent operation, the text used to train and test of the ASR system is completely unconstrained. In contrast, Text-independent speaker recognition usually gives less performance than text-dependent speaker recognition, which requires test input to be the same sentence as training data [10].

2.1 Front-End Processing

Front-end processing is the first component in ASR, therefore the quality of the frontend processing will greatly determine the quality of the later other components. Speech signal changes continuously due to the movements of vocal system and it is intrinsically non-stationary. Nonetheless, in short segments, typically 20 to 40 ms, speech could be regarded as pseudo-stationary signal. Speech analysis is generally carried out in frequency domain with short segments and it is often called short-term spectral analysis. The pre-emphasized stream of digital data is analyzed in frames of 20 ms, at intervals of 10 ms. The Hamming window is used to reduce the distortions caused by the discontinuities at the ends of each frame.

Depending on the acoustic front-end of concatenated features, the resulting feature vectors may have from 20 to 50 components. In real-time speaker applications using low-resource devices, like service accessing through portable or embedded device with low storage and computational capabilities, 50-dimensional feature vectors do not seem suitable. For example, for choosing 20 features from 50 original features we have 4.712×10^{13} searches. Therefore, a further feature set reduction is needed.

2.2 Acoustic Feature Extraction

The speech waveform contains all information about the speaker, and each step in the extraction process can only reduce the mutual information or leave it unchanged. The objective of the feature extraction is to reduce the dimension of the extracted vectors and thereby reduce the complexity of the system. The main task for the feature extraction process is to pack as much speaker-discriminating information as possible into as few features as possible. The choice of features in any proposed ASR system is of primary concern. Most feature extraction techniques in speaker recognition were originally used in speech recognition. However, the focus in using these techniques was shifted to extract features with high variability among people.

Most commonly used features extraction techniques, such as MFCCS and Linear Prediction Cepstral Coefficients (LPCCs) have been particularly popular for ASR systems in recent years. This transforms give a highly compact representation of the spectral envelope of a sound. Delta-features, regardless of what features they are based, can be computed as a one-to-one function of the features themselves. Therefore, the delta-features do not contain more information than is already in the features, and from the theory, no gain can be achieved by using them together with the features. However, the delta-features can be used as a simplified way of exploiting inter-feature dependencies in sub-optimal schemes.

The number of features should be also relatively low. Traditional statistical models such as the Gaussian mixture model [11] cannot handle high-dimensional data. The number of required training samples for reliable density estimation grows exponentially with the number of features; this problem is known as the curse of dimensionality. The computational savings are also obvious with low-dimensional features. On other hand, dealing with hundreds of features leads to the increase of computational workload of recognition process.

MFCC Features.
State of the art systems use the Mel Frequency Cepstrum Coefficient for speech and speaker recognition, because they convey not only the frequency distribution identifying sounds, but also the glottal source and the vocal tract shape and length, which are speaker specific features. They are extensions of the cepstral which are used to better represent human auditory models**.** The MFCCs are calculated as illustrated in Figure 2.

Differential Features.
Temporal changes, in speech spectra, play an important role in perception. This information is captured in the form of velocity coefficients and acceleration coefficients referred to as differential or dynamic features. The first order derivative of MFCCs is called Delta coefficients and their second order derivative is called Delta-Delta coefficients. The delta coefficients are computed using linear regression:

$$\Delta x(m) = \frac{\sum_{i=1}^{j}(i)(x(m+1) - x(m))}{2 \times \sum_{i=1}^{j} i^2} \quad (1)$$

where, 2j+1 is the regression window size and x denotes the cepstrum. The second-order derivatives are computed using the same linear regression applied to a window of delta coefficients.

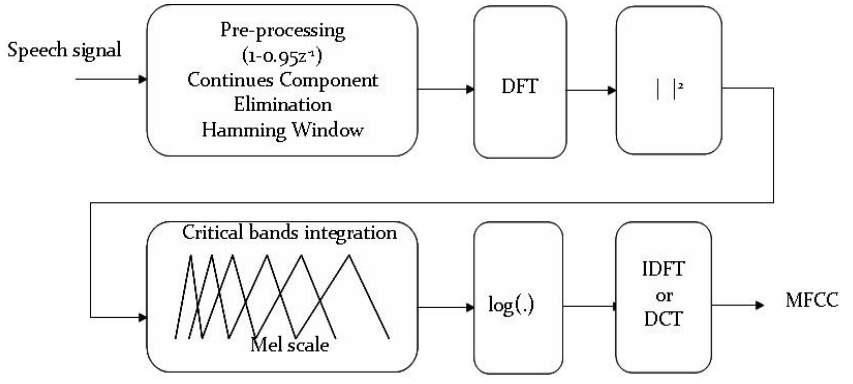


Fig. 2. MFCC features extraction

2.3 Classifier

The performance of selected feature subsets is measured by invoking an evaluation function with the corresponding reduced feature space and measuring the specified classification result. Recognition process was performed using the KNN classifier.

3 Genetic Algorithm

Genetic algorithms [12] are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a single chromosome and apply recombination operators to them so as to preserve critical information. GAs are often viewed as function optimizers, although the range of problems to which GAs have been applied is quite broad. The major reason for GAs popularity in various search and optimization problems is its global perspective, wide spread applicability and inherent parallelism. GA starts with a number of solutions known as population. These solutions are represented using a string coding of fixed length. After evaluating each chromosome using a fitness function and assigning a fitness value, three different operators selection, crossover and mutation- are applied to update the population. The selection is applied on a population and forms a mating pool. Crossover operator is applied next to the strings of mating pool. It picks two strings from the pool at random and exchanges some portion of the strings between them. Mutation operator changes a 1 to 0 and vice versa. An iteration of these three operators is known as a generation. If a stop criterion is not satisfied this process repeats. This stop criterion can be defined as reaching a predefined time limit or number of generations or population convergence. A flowchart of working principles of a simple GA is shown in Fig. 3.

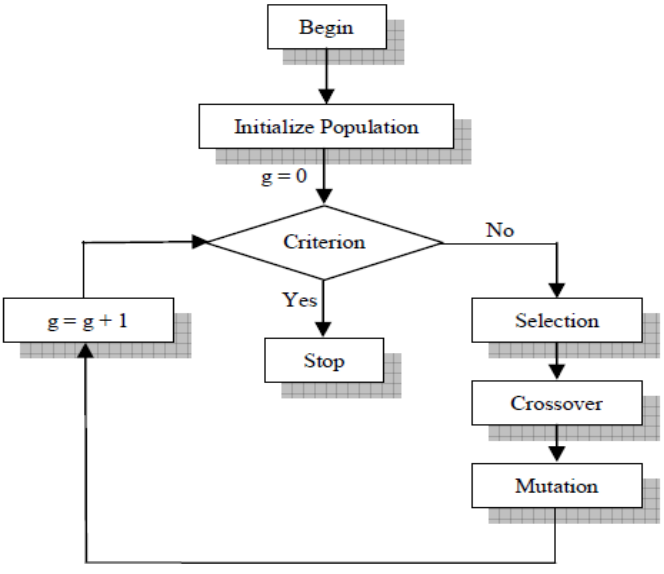


Fig. 3. Simple Genetic Algorithm

3.1 GA Optimization Process

Feature selection (or extraction) techniques can be categorized according to a number of criteria. One popular categorization consists of "filter" and "wrapper" to quantify the worth of features [13]. Filters use general characteristics of the training data to evaluate attributes and operate independently of any learning algorithm. Wrappers, on the other hand, evaluate attributes by using accuracy estimates provided by the actual target learning algorithm. Due to the fact that the wrapper model is computationally expensive [14], the filter model is usually a good choice when the number of features becomes very large. In our ASR system, we use an approach similar to one reflected in [15], after pre-processing of speech signals, the front-end is used to transform the input signals into a feature set (feature vector). After that, Feature selection is applied using GA to explore the space of all subsets of given feature set in order to reduce the dimensionality and improve the performance. The feature set optimization process is shown in Fig. 4.

3.2 MFCC Features Encoding

For GA-based feature selector, we set the length of chromosomes as the number of features. In a chromosome, each gene gi corresponds to the ith feature. If $gi = 1$, this means we select the ith feature. Otherwise, $gi = 0$, which means the ith feature is ignored. By iterations of producing chromosomes for the new generation, crossover and mutation, the algorithm tries to find a chromosome with the smallest number of 1's and the classifier accuracy is maximized.

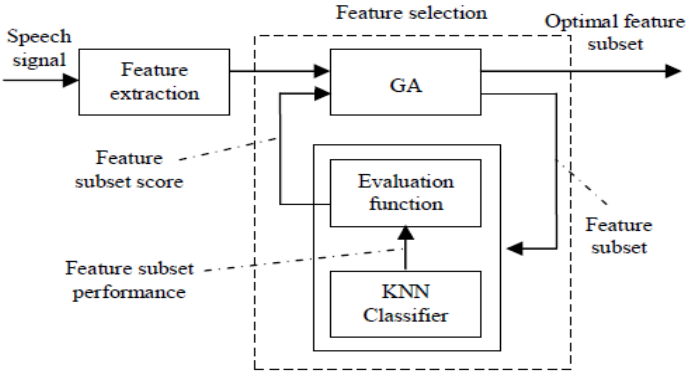


Fig. 4. GA optimization process

3.3 MFCC Features Encoding

For GA-based feature selector, we set the length of chromosomes as the number of features. In a chromosome, each gene *gi* corresponds to the *i*th feature. If $gi = 1$, this means we select the *i*th feature. Otherwise, $gi = 0$, which means the *i*th feature is ignored. By iterations of producing chromosomes for the new generation, crossover and mutation, the algorithm tries to find a chromosome with the smallest number of 1's and the classifier accuracy is maximized.

3.4 Population Initialization

GA starts by generating an initial population of chromosomes. This first population must offer a wide diversity of genetic materials. The gene pool should be as large as possible so that any solution of the search space can be engendered. Generally, the initial population is generated randomly. The chromosome size is equal to 30 (10MFCC, 10 ΔMFCC and 10 ΔΔMFCC). We choose the population size m = 50 and the maximum number of iterations k = 100.

3.5 Population Evaluation

The performance criterion is due to Error Rate (ER) and number of feature selected. The best feature subset found is then output as the recommended set of features to be used in the actual design of the classification system. In our experiments, the fitness function is defined according to Equation (2):

$$Fitness = \alpha.\varphi_s + \beta.\frac{|N|-|S|}{|N|} \quad (2)$$

where φ_s is classifier performance for the feature subset S, |N| is the total number of features, |S| is feature subset length, $\alpha \in [0;1]$ and $\beta = 1-\alpha$. In our experiment we assume that classification quality is well important as the subset length and we choose $\alpha = \beta = 0.5$.

3.6 Chromosome Selection

After evaluating all individuals of the population, we apply the elitist selection method. This method allows the genetic algorithm to retain a number of best

individuals for the next generation. These individuals may be lost if they are not selected to reproduce [16].

3.7 Crossover

Its fundamental role is to enable the recombination of information contained in the genetic heritage of the population. We applied the one point cross with the variable probability ($P_{croossover}$ = % of chromosomes having score > mean (scores)**.**

3.8 Mutation

A mutation is simply a change of a gene found in a locus randomly determined. The altered gene may cause an increase or a weakening of the solution value that represents the individual ($P_{mutation}$ = 0.02).

3.9 Replacement

The elitist replacement is the most suitable in our case; it keeps individuals with the best performance from one generation to the next. The weakest individual of the current population is replaced by the fittest individual of the immediately preceding population.

3.10 Stop Criterion

As we seek the optimum, we choose our stopping criterion the maximum number of generations even if the optimum is found before, something we can know in advance.

4 Results and Discussions

The QSDAS Base [17] is used in this paper. This corpus contains 77 speakers, each speaking 21 sentences partitioned in three sets 10, 10 and 1 respectively. The 77 speakers included in all sets were used during the trials. The effectiveness and performance of our proposed GA-based feature selection algorithm is evaluated using series of experiments. All experiments have been run on Pentium IV, Windows XP, using Matlab 7.0. The classification error rate and feature subset length are the two performance criteria considered. Tables I to IV show the feature vector size reduction and error rate reduction achieved by our genetic algorithm in case of MFCC, ΔMFCC, ΔΔMFCC and MFCC + ΔMFCC + ΔΔMFCC features, respectively.

Table 1. ER and Selected Feature Vector Size Reduction (MFCC)

	Size reduction (%)	RR improvement (%)
S_1	50,00	10,00
S_2	50,00	10,00
S_3	40,00	5,00
S_4	60,00	10,00
S_5	50,00	10,00

Table 2. ER and Selected Feature Vector Size Reduction (ΔMFCC)

	Size reduction (%)	RR improvement (%)
S_1	40,00	5,00
S_2	40,00	10,00
S_3	50,00	5,00
S_4	50,00	10,00
S_5	50,00	10,00

Table 3. ER and Selected Feature Vector Size Reduction (ΔΔMFCC)

	Size reduction (%)	RR improvement (%)
S_1	40,00	10,00
S_2	30,00	10,00
S_3	40,00	15,00
S_4	40,00	25,00
S_5	60,00	10,00

Table 4. ER and Selected Feature Vector Size Reduction (MFCC+ΔMFCC+ΔΔMFCC)

	Size reduction (%)	RR improvement (%)
S_1	43,33	10,00
S_2	43,33	15,00
S_3	40,00	20,00
S_4	53,33	25,00
S_5	60,00	25,00

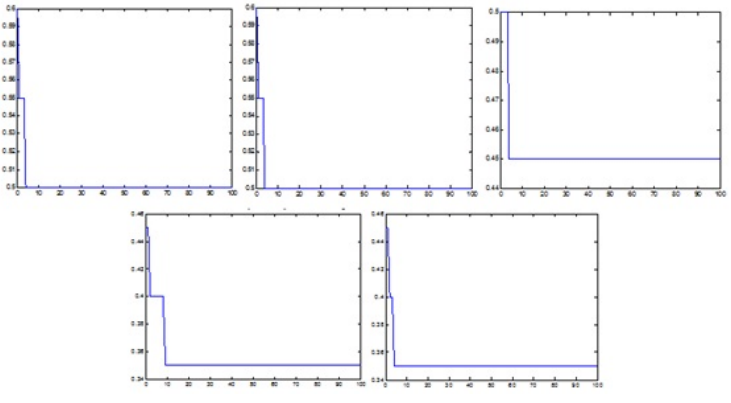


Fig. 5. ER using MFCC for the five sets S_1 (top left) to S_5 (bottom right)

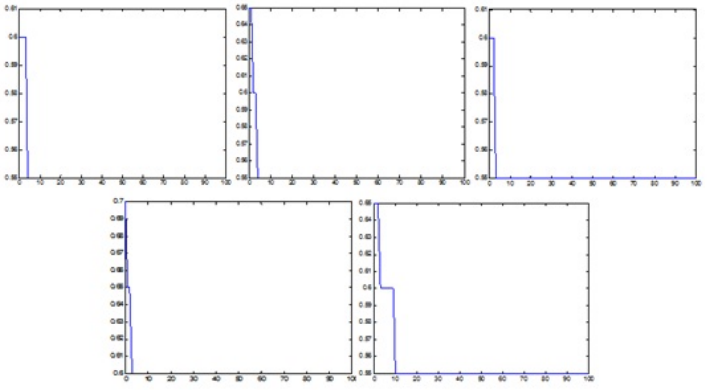


Fig. 6. ER using ΔMFCC for the five sets S_1 (top left) to S_5 (bottom right)

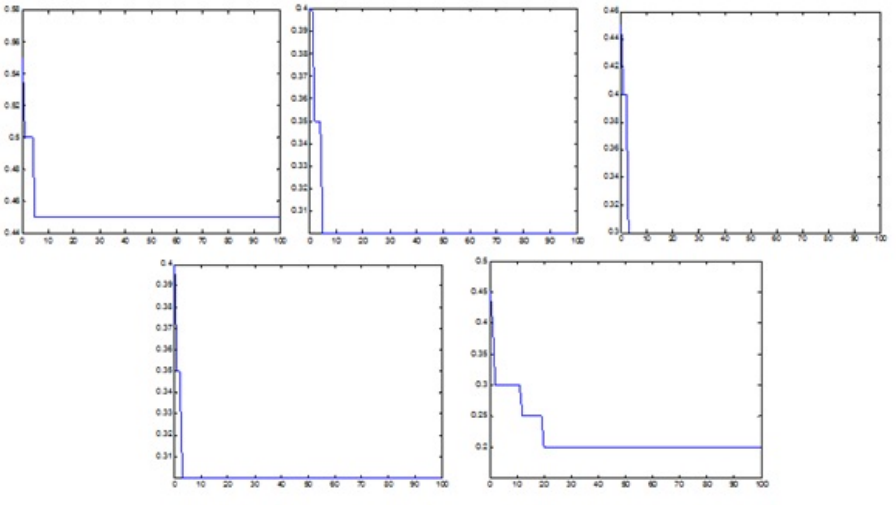


Fig. 7. ER using ΔΔMFCC for the five sets S_1 (top left) to S_5 (bottom right)

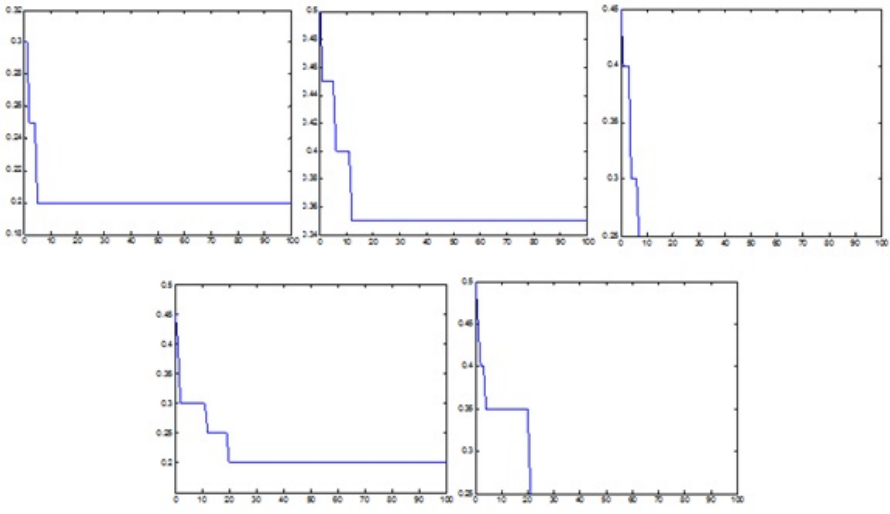


Fig. 8. ER using MFCC+ΔMFCC+ΔΔMFCC for the five sets S_1 (top left) to S_5 (bottom right)

While Figures 5 to 8 show the evolution of the error rate over the generations. We can see that our GA can reduce the dimensionality of features between 30% (the worst case) and 60% of original features (in the best case). On other hand, our the proposed genetic algorithm has improve the classification rate (reduce the error rate) between 5% (in the worst case) and 25% (in the best case) with better results for ΔΔMFCC compared to MFCC or ΔMFCC. Even using small feature vector, the proposed genetic algorithm can obtain better classification accuracy with smaller. The best result is obtained using a subset selected from a combination of all features vectors (MFCC+ΔMFCC+ΔΔMFCC) which confirm that the inclusion of new parameters improves speaker's discrimination.

5 Conclusions and Future Works

In this paper, we addressed the problem of optimizing acoustic feature set by GA-based feature selection algorithm. The GA algorithm adopts classifier performance and the number of the selected features as heuristic information, and selects the optimal feature subset in terms of smallest feature vector size and the best performance of system classifier. The experimental results on QSDAS data sets showed that our GA is able to select the more informative features without loosing the performance; the algorithm can obtain better classification accuracy with smaller feature vector which is crucial for real time applications and low resources devices systems. The feature vectors size is reduced over 60% that led to a less complexity of our ASR system and reduce the ER up to 25%. For future works, we prepare another paper on the use of a multi-objective genetic algorithm by separating the two objectives (feature vector size and classification error rate).

References

1. Day, P., Nandi, A.K.: Robust text-independent speaker verification using geneticprogramming. IEEE Transactions on Audio, Speech, and Language Processing 15(1), 285–295 (2007)
2. Jensen, R.: Combining rough and fuzzy sets for feature selection.Ph.D. thesis, Universityof Edinburgh (2005)
3. Zamalloa, M., Rodríguez, L.J., Peñagarikano, M., Bordel, G., Uribe, J.P.: Comparing genetic algorithms to principal component analysis and linear discriminant analysis inreducing feature dimensionality for speaker recognition. In: GECCO 2008, pp. 1153–1154 (2008)
4. Zhao, Q., Zhang, D., Zhang, L., Lu, H.: Evolutionary discriminant feature extraction withapplication to face recognition. EURASIP Journal on Advances in Signal Processing, 1–13 (2009)
5. Harrag, F., El-Qawasmeh, E., Salman Al-Salman, A.M.: Extracting Named Entities from Prophetic Narration Texts (Hadith). In: Zain, J.M., Wan Mohd, W.M.b., El-Qawasmeh, E. (eds.) ICSECS 2011, Part II. Communications in Computer and Information Science, vol. 180, pp. 289–297. Springer, Heidelberg (2011)
6. Rajavarman, V.N., Rajagopalan, S.P.: Feature Selection in Data-Mining for Genetics Using Genetic Algorithm. Journal of Computer Science 3(9), 723–726 (2007)
7. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
8. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
9. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters 10(5), 335–347 (1989)
10. Nemati, S., et al.: Text-independent speaker verification using ant colony optimization-based selected features. Expert Systems with Applications 38, 620–630 (2011)
11. Reynolds, D.A., Rose, R.C.: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Transactions on Speech and Audio Processing 3(1), 72–83 (1995)
12. Mitchell, M.: An introduction to genetic algorithms. A Bradford book. The MITpress, Cambridge (1999) (fifth printing)
13. Chandra, E., Nandhini, K.: Learning and Optimizing the Features with Genetic Algorithms. International Journal of Computer Applications 9(6), 1–5 (2010)
14. Langley, P.: Selection of Relevant Features in Machine Learning. In: Proceedings AAAI Fall Symposium Relevance (1994)
15. Casillas, J., Cordon, O., Del Jesus, M.J., Herrera, F.: Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. Information Sciences Journal 136, 135–157 (2001)
16. Rokach, L.: Genetic Algorithm-based Feature Set Partitioning for Classification Problems. Journal Pattern Recognition 41(5), 1676–1700 (2008)
17. Harrag, A., Mohamadi, T.: QSDAS: New Quranic Speech Database for Arabic Speaker Recognition. AJSE Journal 35(2C), 7–19 (2010)

On How Percolation Threshold Affects PSO Performance

Blanca Cases, Alicia D'Anjou, and Abdelmalik Moujahid

Computational Intelligence Group of the University of the Basque Country
P. Manuel de Lardizábal n. 1, 20018 San Sebastián, Spain

Abstract. Statistical evidence of the influence of neighborhood topology on the performance of particle swarm optimization (PSO) algorithms has been shown in many works. However, little has been done about the implications could have the percolation threshold in determining the topology of this neighborhood. This work addresses this problem for individuals that, like robots, are able to sense in a limited neighborhood around them. Based on the concept of percolation threshold, and more precisely, the disk percolation model in 2D, we show that better results are obtained for low values of radius, when individuals occasionally ask others their best visited positions, with the consequent decrease of computational complexity. On the other hand, since percolation threshold is a universal measure, it could have a great interest to compare the performance of different hybrid PSO algorithms.

Keywords: Particle Swarm Optimization parameters, Percolation theory, Hybrid PSO.

1 Introduction

Percolation theory appears in very different random structures, including spread of diseases, fire propagation in forests, phase transition in solids, diffusion in disordered media, etc. There are number of good reviews of the percolation theory [10,11]. This theory is particularly well adapted to describe global physical properties, such as the connectivity and conductivity behaviour of geometrically complex systems. This work analyzes the role of percolation threshold, a universal measure of connectivity in graphs, to analyze the convergence of swarms algorithms as a function of the expected number of neighbors at initial step. And therefore, to develop a framework to compare the performance of hybrid PSO algorithms. The analysis of the performance of different hybrid PSO algorithms has been addressed in different works [4,7,3]. However, in this work, we focus on the concept of percolation threshold as a useful tool to define the parameter space for which the performance of the basic PSO algorithm is enhanced.

The basic model of PSO by Eberhart, Kennedy and Shi [8,6] is an stochastic optimization method for D-dimensional functions. Given a population of P individuals i, $1 \leq i \leq P$, uniformly distributed on an D-dimensional hypercube of

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 509–520, 2012.

size S in points $x_i = (x_{i1}, \ldots, x_{id}, \ldots, x_{iD})$, each individual moves to next position according to a velocity vector $v_i = (v_{i1}, \ldots, v_{id}, \ldots, v_{iD})$ both calculated according to (1):

$$\begin{aligned} v_{id} &= w * v_{id} + c_1 * rand\,(1.0) * (p_{id} - x_{id}) + c_2 * Rand\,(1.0) * (p_{g_i d} - x_{id}) \\ x_{id} &= x_{id} + v_i \end{aligned} \tag{1}$$

where values $rand(1.0)$ and $Rand(1.0)$ are drawn at random according to a uniform distribution in the interval $[0, 1]$. Velocity $v_i\,(t+1)$ is a randomized linear combination of three forces: (i) inertia, weighted by parameter $w \in [0, 1]$, (ii) the personal attraction c_1 to the position p_i where the minimum value of a function $f\,(p_i)$ was found, and (iii) the social attraction to the position p_{g_i} of the best goal remembered by the neighbors of individual i in a given topology.

In many works, the neighborhood of each individual includes the whole population and the best goal $p_g = p_{g_i}$ is visible for all the agents in a blackboard architecture. In these cases, the gain in efficiency runs in parallel to the lose of autonomy of agents in the model. For realistic purposes, as in swarm robotics, the notion of Euclidean neighborhood is a building block since sensors and effectors have physical limits. Some authors [12,14] have carried out social network statistical studies to address the effects of neighborhood topology in PSO algorithms, concluding that the topology used conditions strongly the convergence of these algorithms.

In this work, based on the so called continuum or disk percolation model [1], we investigate the effect of the percolation threshold according to the topology of geometric graphs. A geometric graph is a 2D representation of a graph: given a neighborhood radius R and a square board of size S, let $\{x_1, \ldots, x_P\}$ be the initial positions of individuals composing the swarm selected at random following a uniform distribution. The geometric graph is built connecting each node to all neighboring nodes in radius R. Percolation disk theory allows the calculus of the radius R, that ensures a mean of a neighbors per node (called the area or degree) with parameters of population of P individuals and world size S, according to (2).

$$R = \sqrt{\frac{aS^2}{\pi P}} \tag{2}$$

We study experimentally the convergence of PSO algorithms focusing on the concept of percolation. The percolation concept comes from chemistry and refers to the way a liquid flows through a porous medium, like a large piece of pumice stone and sets what is the probability that the center of the stone is wet when we immerse it in water. Other example is making coffee: if ground coffee is fine grained, under a determined threshold for the diameter of grain, water stagnates, while over this threshold water falls through coffee. In terms of random graphs, percolation is identified to the phenomenon of passing from numerous small connected components to a connected one as a function of the number of expected neighbors a. The critical value of a receives the name of percolation threshold a_c.

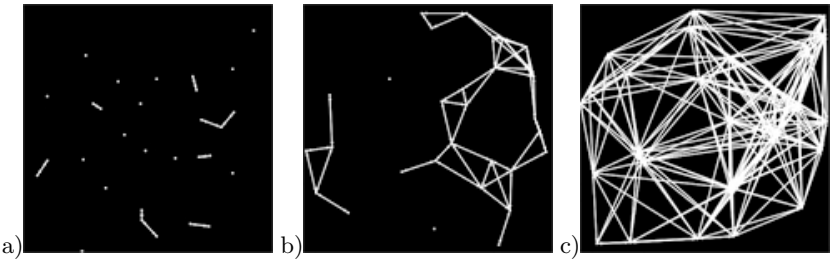


Fig. 1. Random geometric graphs with parameters a) $a = 0.8$, $R = 18.426$, b) $a = 4.512$, $R = 43.760$, c) $a = 29$, $R = 110.941$

Percolation is a complex phenomenon which suggests that agents in a swarm only need a minimum of social information to perform the optimization task, being redundancy a source of futile complexity. It has been proved [1] with confidence 99.99% the following bound for the critical percolation threshold a_c and the corresponding critical value R_c, obtained as the average degree over the squares of any size S:

$$4.508 \leq a_c \simeq 4.512 \leq 4.515 \tag{3}$$

$$43.741 \leq R_c \simeq 43.758 \leq 43.775 \tag{4}$$

We will show experimentally that neighborhoods of critical radius R_c are enough to reach goals in a minimum number of steps. The following definitions and parameters were set in order to compare results with those of [2,5,13].

- Neighbors are determined by the euclidean distance d: let x_i be the position of individual i, $N_R(x_i) = \{x_{i_1}, \ldots, x_{i_{n_i}}\}$ is the set of neighboring individuals in radius R, that is $d(x_i, x_{i_j}) \leq R$, $1 \leq j \leq n_i$.
- The neighbors best value of function f known by agent i in its neighborhood is defined as: $p_{g_i} = min_f \{p_{i_j} : x_{i_j} \in N_R(x_i)\}$, that is, the position p_{i_j} remembered by neighbors (at current position x_{i_j}) minimizing goal function f. The social best of x_i is defined as the mean neighbors best value at final step, $\frac{\sum p_{g_i}}{P}$.
- Experimental parameters are exactly those given in [5]: a swarm of $P = 30$ individuals and square side $S = 200$ and center (0,0). Velocity module $|v| = \sqrt{v_1^2 + v_2^2}$ is limited by $x_{max} = v_{max} = 100$, meaning that individuals move a step forward in the direction of vector v with step length $s(v) = v_{max}$ if $|v| > v_{max}$ and $s(v) = |v|$ otherwise. Clerc's [2] optimal parameters $c_1 = c_2 = 1.49445$ and constant constriction factor $w = 0.729$ where used.
- Benchmark functions are those used in [5,13]. The model has been implemented in Netlogo 4.1, running in concurrent mode (individuals are updated once the run step ends) with a float precision of 16 decimals. An experiment consists in 100 runs for each function, 20 for each instance of parameter R.

– We compare results for Clerc's parameters set [2,5] $w = 0.729$, $c_1 = c_2 = 1.49445$ and Trelea's parameters $w = 0.6$, $c_1 = c_2 = 0.7$. Since disk percolation threshold is a 2D measure, the dimension will be $D = 2$ for the scope of this work. The domain of the benchmark functions will be $x_{max} = 100 = v_{max}$, as well as the goal, 0.01 for all functions. In this way results concerning percolation threshold can be easily compared.

2 Analysis of Goals as a Function of the Expected Degree at Initial Step

Applying equation (2), we explore the success of PSO algorithm as a function of radius $R(a)$ according to a longitudinal study of the mean degree expected at initial step.

– Dataset A: $0.000 \leq a \leq 0.9$, meaning that only a fraction of the individuals has neighbors: from 0.0 to 0.9 with step 0.1. Table 1 shows the corresponding neighborhood radius $R(a)$. Each function is run 20 times, so for each $R(a)$ 100 runs were completed.
– Dataset B: $1 \leq a \leq 30$, from the minimum (1 neighbor) to the maximum (29 neighbors) plus one, 30 neighbors with step 1. Neighborhood radius R varies in the interval $20.601 \leq R \leq 112.838$.
– Best global value registers at each run step the best position remembered by any agent, that is $best - global - val\,(t) = min\,\{f\,(p_i\,(t)) : 1 \leq i \leq P\}$, updated every time that an individual finds a smaller value $x_i\,(t+1) < best - global - val\,(t)$.
– Mean best neighbors value is the average social information over all the individuals, $best - neighbors - val_i\,(t) = min\,\{f\,(p_j\,(t)) : \}$, updated every time that an individual finds a smaller value $x_i\,(t+1) < best{-}global{-}val\,(t)$.

Varying the benchmark function and the radius of neighborhood as parameters, each pair 20 runs, with goal precision 0.01 and a maximum number of iterations $P^2 = 900$, results in table 2. Even including the case of neighborhood radius $R = 0$, the performance of the PSO algorithm does better for graphs with initial mean degree $0 \leq a \leq 0.9$: with the only exception of f_1 the mean best global values are successful reaching the goal 0.01. As was expected, the objective function has to do in the performance of the algorithm: it is harder to optimize function f_1, but never ceases to amaze that respecting to the social force, less neighbors means better performance of PSO. In a further section we will analyze accurately the performance of the algorithm for values $a = 0$ and $0 < a \leq 0.9$. The mean iterations (over a maximum of $P^2 = 900$) averaged over all runs is similar for datasets A and B.

Figure 2 shows mean values of the best global as a function of neighborhood radius $R\,(a)$ calculated according to (2). As it can be appreciated, the best result corresponds to a disk radius $R\,(a) = 6.515$ where the degree $a = 0.100$ is the first one being explored. This means that initial configurations, where 10% of individuals have a neighbor, give the best performance. Note that $a < \log\,(\log\,(P)) =$

Table 1. Values of neighborhood's radius $R(a)$ as a function of expected degree a

a	0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.800	0.900
R	0.000	6.515	9.213	11.284	13.029	14.567	15.958	17.236	18.426	19.544

Table 2. Percentage of goals, average best global value and iterations over all runs for datasets A, $0 \leq a \leq 0.9$, $0 \leq R \leq 19.544$ and B, $1 \leq a \leq 30$, $20.601 \leq R \leq 112.838$..

Function	Goals		Best global		Steps			
	A	B	A	B	A goal	B goal	A all	B all
Spherical f_0	95.50%	100.00%	0.01	0.00	92	56	129	56
Rosenbrock f_1	52.00%	9.00%	0.09	0.06	588	370	738	852
Rastrigin f_2	95.00%	100.00%	0.01	0.00	82	58	123	58
Griewank f_3	100.00%	100.00%	0.00	0.00	2	2	2	2
Schaffer f_6	100.00%	100.00%	0.00	0.00	5	4	5	4
Total	88.50%	81.80%	0.02	0.02	108	38	199	195

0.169 since $P = 30$ and hence each run has computational complexity at most of the order of $P \log(\log(P))$. Over the critical radius R_c the mean global best stops increasing to become constant.

As one might expect, Fig. 3 shows that the only memory of the best personal value gives comparatively a poor score. Even so, this score is over the 60%. The importance of social information is crucial: surprisingly the smallest number of initial neighbors gives the majority of the goals. Near of the percolation threshold $R_c \simeq 43,758$ the score starts remaining constant. Finally, the mean iterations needed for success, with a limit of $P^2 = 900$, reach the minimum at $R = 6.515$. From this value, the mean over all the runs slightly increases becoming constant, near to 195 steps, for values upper to $R_c \simeq 43.758$. On the other hand, the average over successful runs decreases until the threshold R_c is passed reaching a constant value near in mean to 36 steps.

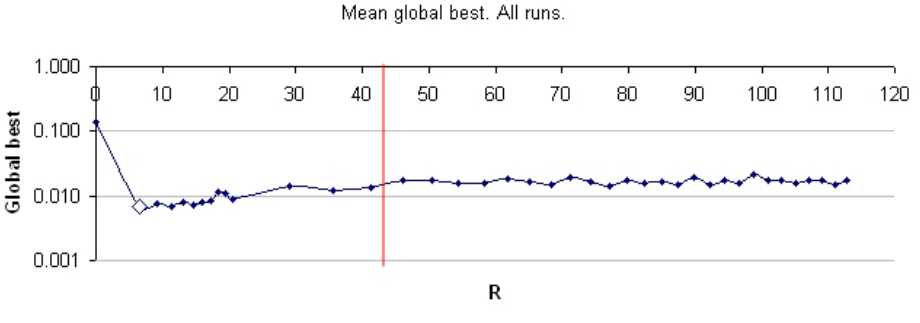


Fig. 2. Mean global best as a function of neighborhood radius R shows that PSO behaves better under critical $R_c \simeq 43,758$. Non successful runs are included. A minimum is obtained at R=6.515 growing until R_c is reached.

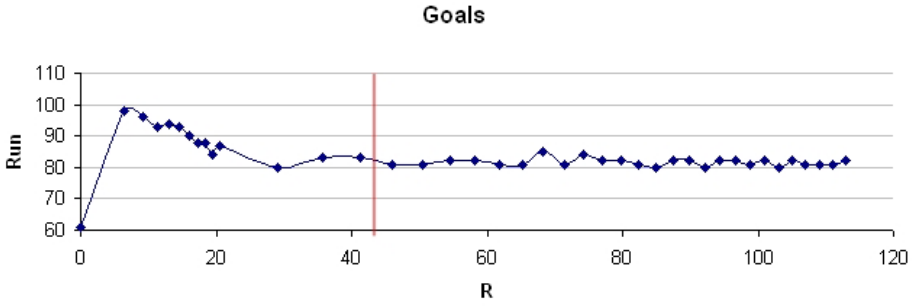


Fig. 3. Number of goals reached as neighborhood R increases

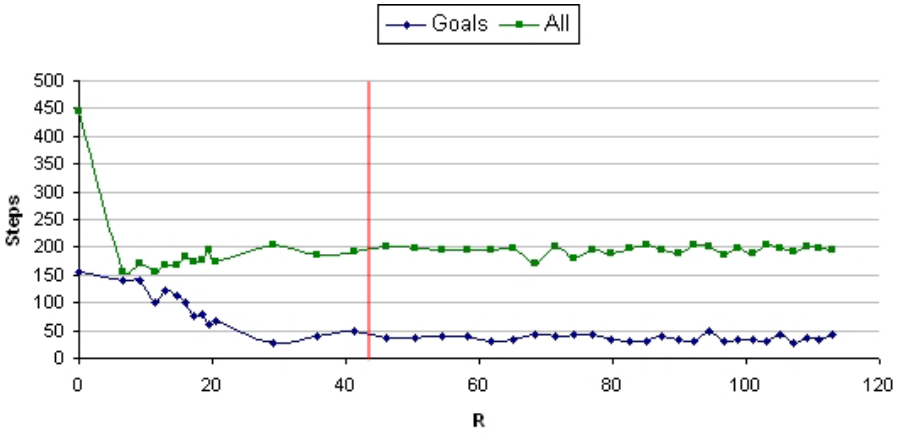


Fig. 4. Mean number of iterations. Square marks represent the mean over all runs and diamonds over successful runs.

The mean value of social best, that is the average value of the best personal value of neighbors, was measured at final step. The mean over all runs is presented in Fig. 2. Beyond value $R = 61.804$ the mean reaches value 0 rounded to 3 decimals. This occurs because some individuals fly outside the limits of the board having 0 neighbors. This curve, as will be shown in next section, seems to be very sensible to the variations of R, decreasing drastically at critical radius R_c.

3 Exploring the Disk Percolation Threshold

Values of R around critical radius R_c were finely explored varying degree a along the interval $[4.37, 4.515]$ just before the critical threshold $a_c = 4.512$ with step 0.001. Hence $R(a)$ varies in the interval $[43.066, 43.775]$ in intervals of 0.01. Each instance of parameter R was run 100 times, 20 for each function f_i. Table 3 shows

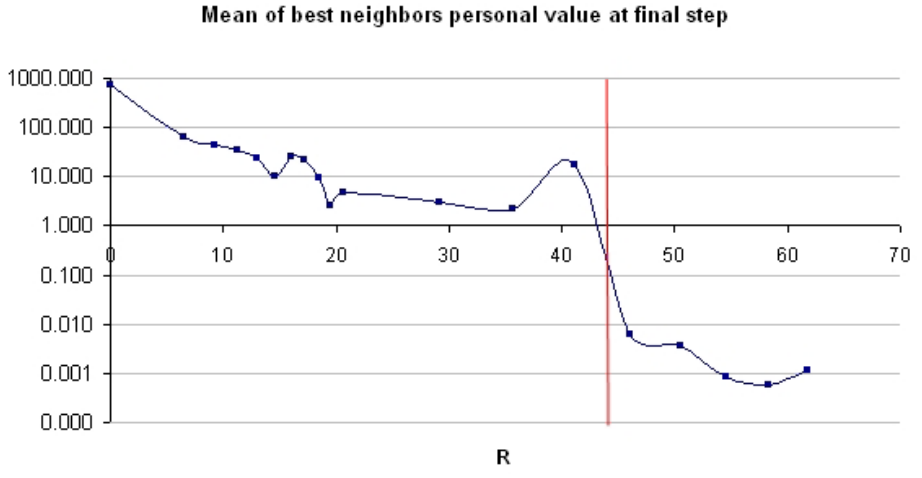


Fig. 5. The mean of social best at final step

the means of goals, the best global value reached and final step. Comparing to table 2, the total of successful runs equals the goals of dataset A, 88.5%, as well as the mean best global value, 0.02. The difference with dataset A is in the number of steps: at critical radius R_c the performance of PSO algorithm decreases to mean of 42 iterations in successful runs, near to the 38 steps reached in dataset B. Mean steps, 197, are just in the middle of the 199 obtained for dataset A and the 195 of dataset B. As a conclusion, the performance of PSO algorithm with a_c between 4 and 5 neighbors is the same as with large values, even when the neighborhood comprises all agents.

In closing, the curves of mean social and personal values are represented in Fig. 3 together with the global best value and the percentage of goals. Pearson correlations were calculated between all the series being the most significant the negative correlation between goals and global best, -0.385. We conjecture that the lack of correlation between variables is a characteristic of the interval around the critical radius.

Table 3. Percentage of goals and mean best global values. With the only exception of f_1 the mean best global values are successful reaching the goal 0.01.

	Goals reached.	Best global all.	Steps Goals	Steps all
Spherical f_0	95.50%	0.01	57	57
Rosenbrock f_1	52.00%	0.09	509	861
Rastrigin f_2	95.00%	0.01	60	60
Griewack f_3	100.00%	0.00	2	2
Schaffer f_6	100.00%	0.00	4	4
Total	88.50%	0.02	42	197

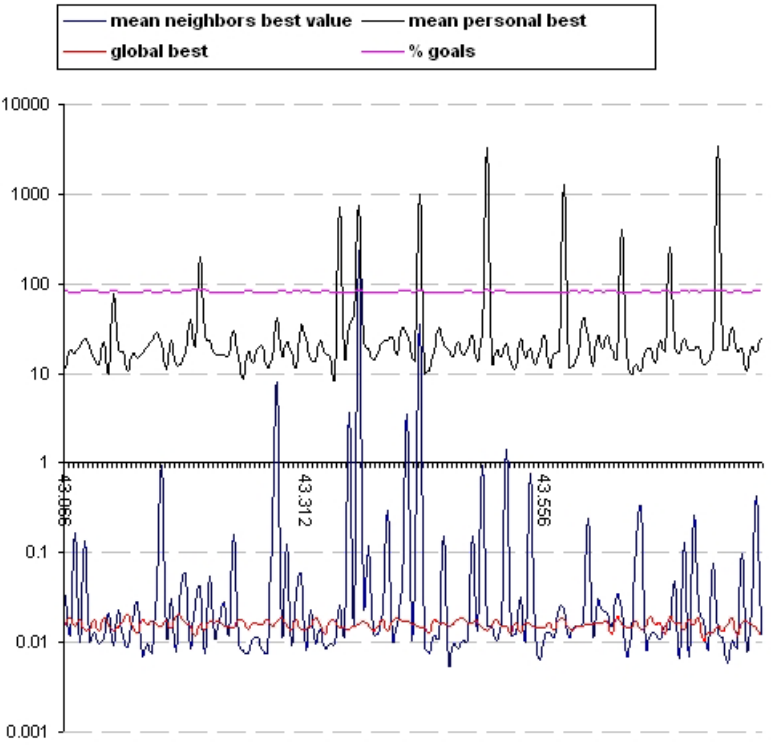


Fig. 6. Values just before $R_c \simeq 43,758$: Social and personal information show a great variability, while goals and global best vary slightly

4 Exploring R in the the Interval [0, 0.9]

As it has been reported in Fig. 2 the minimum value of global best occurs at $a = 0.1$, $R(a) = 6.515$. For the same radius, the Fig. 3 shows a maximum of successful runs. This suggest that PSO does better with less, but some, social contact: degree $a = 0.1$ means that only one in ten individuals has one neighbor at initial step. The interval $R \in [0, 0.9]$, or equivalently $a \in [0, 0.001909]$ was explored to investigate the pass of having no neighbors to have a minimum of social contact. A total of 91 instances of R with 20 runs per function were executed. Table 4 shows surprising results: the 94.13% of the runs were successful in the interval [0.75, 0.9] with the only handicap of longer iterations, reaching a mean of 258 steps.

Figure 4 shows that high values of social information are related to increasing goals and decreasing global values. To corroborate this tendency, the Pearson coefficient of correlation was calculated in table 5. A very significant positive correlation exists between the percentage of goals with R and the mean social best: the major the dispersion of neighbors the major the goals. Conversely, the

Table 4. Exploration of $R \in [0, 0.9]$ subdivided in intervals:

function	Goals				Steps all
	0.0-0.25	0.26-0.50	0.51-0.75	***0.75-0.90***	**0.75-0.90**
Spherical f_0	60.58%	91.20%	99.60%	***100.00%***	**262**
Rosenbrock f_1	2.88%	19.80%	50.60%	***70.67%***	**750**
Rastrigin f_2	60.19%	90.00%	99.80%	***100.00%***	**265**
Griewank f_3	100.00%	100.00%	100.00%	***100.00%***	**3**
Schaffer f_6	100.00%	100.00%	100.00%	***100.00%***	**9**
Total	64.73%	80.20%	90.00%	***94.13%***	**258**

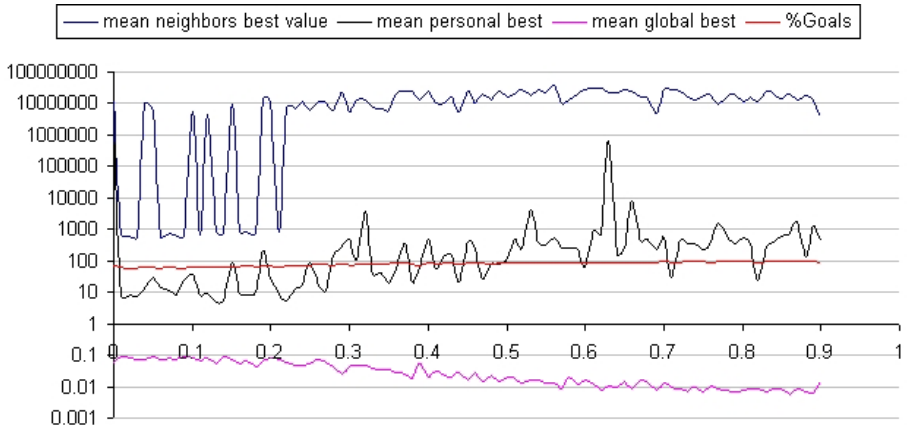


Fig. 7. Comparison of mean goals and the mean social, personal and global best for small values of $R \in [0, 0.9]$. The minimal contact with neighbors, for radius over $R = 0.6$ ensures the convergence.

goals go in a very significant negative correlation with global best: the less the best global value the more the goals . The mean best global value decreases as radius R and social best increase.

5 Comparing Different Domain Through Degree a

Experiments in previous sections were made over the same domain conditions of f_0 and f_6: $x_{max} = 100$, $S = 200$. Disk percolation theory gives us a way to relate the expected number of neighbors a to neighborhood radius R of individuals in spite of different values of parameters of the world, population P and square side S, were supplied. In table 6 we report values of R for different degree values $a_1 = 0.00151$, $a_2 = 0.8$, $a_3 = 29$ and at a critical radius $a_c = 4.512$, calculated according to (2). Table 6 shows that conditions $x_{max} = 100 = \frac{S}{2}$ and $P = 30$ give for each a a corresponding radius $R\,(a)$ in the intervals studied in previous sections: $R(0.00151) = 0.8 \in [0.75, 0.9]$ analyzed in section 4, $R(0.8) = 18.426$ in

Table 5. Pearson coefficients of correlation for dataset $R \in [0, 0.9]$

Correlation	R	Social best	Personal best	Global best	Goals	Steps
R	1	-	-	-	-	-
Social best	0.591	1	-	-	-	-
Personal best	-0.122	0.024	1	-	-	-
Global best	-0.908	-0.712	0.049	1	-	-
Goals	0.956	0.720	-0.058	-0.947	1	-
Steps	-0.96	-0.668	0.042	0.915	-0.967	1

Table 6. Radius values computed for each of the benchmark functions for different values of a

f	a_c	S	P	R_c
f_0	0.00151	200	30	0.801
f_1	0.00151	60	30	0.240
f_2	0.00151	10.24	30	0.041
f_3	0.00151	1200	30	4.803
f_6	0.00151	200	30	0.801
f_0	4.512	200	30	43.760
f_1	4.512	60	30	13.128
f_2	4.512	10.24	30	2.241
f_3	4.512	1200	30	262.561
f_6	4.512	200	30	43.760

f	a_c	S	P	R_c
f_0	0.800	200	30	18.426
f_1	0.800	60	30	5.528
f_2	0.800	10.24	30	0.943
f_3	0.800	1200	30	110.558
f_6	0.800	200	30	18.426
f_0	29	200	30	110.941
f_1	29	60	30	33.282
f_2	29	10.24	30	5.680
f_3	29	1200	30	665.648
f_6	29	200	30	110.941

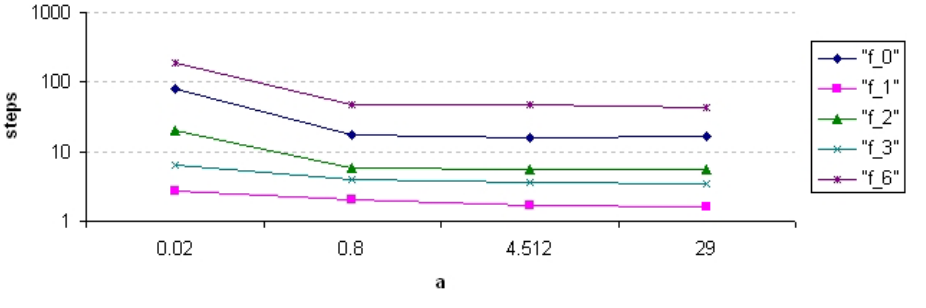


Fig. 8. Mean steps with Trelea's parameters $w = 0.6,\ c_1 = c_2 = 0.7$

dataset A and $R(29) = 110.941$ in dataset B of section 2 and the critical radius $R_c = R(a_c)$ in section 3. We wonder if a change of parameters values will report different patterns of PSO convergence.

For each value of parameter a the experiment was repeated 500 times, 100 for each function f_i. All the runs converged with the goal precision given in [5,13]. Hence, the only variable to compare is the mean number of steps, given in Fig. 5. The same pattern of convergence appears: at small values of

$a < R(a) \in [0.75, 0.9]$ all the experiments converge, but more steps are needed. From the percolation threshold, a universal measure, the number of steps slightly decreases.

6 Conclusions

We show in this work that percolation threshold is a tool to analyze the convergence of swarms as a function of the expected number of neighbors at initial step. While traditionally PSO algorithm is based on the external calculus of the global best goal by reasons of convergence, we present here experiments suggesting that, for one hand, very small amounts of information, translated to very low values of a are enough to provoke the convergence of PSO swarms. Moreover, better scores of convergence are found. On the other hand, a low degree $a_c = 4.512$ is enough to both purposes: convergence to the minimum and short runs. The steps needed for convergence become constant from this critical value. Hence, more neighbors do the same than a_c neighbors. Disk percolation threshold is a measure for 2D systems. Efforts will be made in investigating the calculus of percolation thresholds for higher dimension. An idea to do that is to compare the area of the circle of radius R to the surface of the square $\pi R^2/S^2$ looking for the value of radius R' such that a D-dimensional hypersphere of this radius maintain the same proportion with respect to the volume of the hyperboard.

This work can be extended in a further development, comparing previous hybrid PSO approaches and using more benchmark functions. Moreover, we think it is of great interest to study the implications of percolation threshold in fuzzy neurocomputing areas [9], since percolation is considered as a second order state transition phenomenon, having hence a crisp nature.

References

1. Bollobás, B., Riordan, O.: Percolation. Cambridge University Press, UK (2006)
2. Clerc, M., Kennedy, J.: The particle swarm explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002); cited By (since 1996) 2079
3. Corchado, E., Abraham, A., Ponce Leon Ferreira de Carvalho, A.C.: Hybrid intelligent algorithms and applications. Inf. Sci. 180(14), 2633–2634 (2010)
4. Corchado, E., Graña, M., Wozniak, M.: Editorial: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
5. Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, vol. 1, pp. 84–88 (2000); cited By (since 1996) 660
6. Eberhart, R.C., Shi, Y.: Particle swarm optimization: Developments, applications and resources. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, vol. 1, pp. 81–86 (2001); cited By (since 1996) 890
7. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)

8. Kennedy, J.: Particle swarm: Social adaptation of knowledge. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, pp. 303–308 (1997); cited By (since 1996) 436
9. Pedrycz, W., Aliev, R.A.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
10. Sahimi, M.: Applications of Percolation Theory. Taylor and Francis, UK (1994)
11. Stauffer, D., Aharony, A.: Introduction to Percolation Theory. Taylor and Francis, UK (1992)
12. Toscano-Pulido, G., Reyes-Medina, A.J., Ramírez-Torres, J.G.: A statistical study of the effects of neighborhood topologies in particle swarm optimization. SCI, vol. 343, pp. 179–192 (2011)
13. Trelea, I.C.: The particle swarm optimization algorithm: Convergence analysis and parameter selection. Information Processing Letters 85(6), 317–325 (2003)
14. Zhang, L., Mu, H.P., Jiao, C.Y.: Particle swarm optimization with highly-clustered scale-free neighborhood model. In: Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010, vol. 2, pp. 661–663 (2010)

Pollen Grains Contour Analysis on Verification Approach

Norma Monzón García[1], Víctor Alfonso Elizondo Chaves[2], Juan Carlos Briceño[2], and Carlos M. Travieso[1]

[1] Signals and Communications Department, Institute for Technological Development and Innovation in Communications, University of Las Palmas de Gran Canaria, Campus University of Tafira, 35017, Las Palmas de Gran Canaria, Las Palmas, Spain
{ctravieso}@dsc.ulpgc.es
[2] Computer Science Faculty, University of Costa Rica, University Campus Rodrigo Facio, 11501, San José, Costa Rica
{victor.elizondo,juan.briceno}@ucr.ac.cr

Abstract. Earth's biodiversity has been suffering the effects of human contamination, and as a result there are many species of plants and animals that are dying. Automatic recognition of pollen species by means of computer vision helps to locate specific species and through this identification, study all the diseases and predators which affect this specie, so biologist can improve methods to preserve this species. This work focuses on analysis and classification stages. A classification approach using binarization of pollen grain images, contour and feature extraction to locate the pollen grain objects within the images is being proposed. A Hidden Markov Model classifier was used to classify 17 genders and species from 11 different families of tropical honey bee's plants achieving a mean of 98.77% of success.

Keywords: Pollen grains, Pollen classification, contour features, HMM, contour extraction, noise elimination.

1 Introduction

The recognition of pollen grains is an issue that has gained the attention on tropical countries, on both the theoretical and practical life. This topic has many applications, for example, species recognition, population estimation of species according to its biodiversity, knowledge of the extinction's patterns or region's reproduction. Within the field of pattern recognition, the pollen grains recognition is one of the areas that has embodied to this type of classification, not only by its shape or contour but also by its color, taking this to a higher level of complexity and variety when classifying patterns.

In the field of automatic pollen species classification, there have been a high number of research proposals, gaining a lot of progress in the species classification field. One of the first works in which texture features and neural networks were used for pollen grains identification task is shown in [1]. Another approach is presented by

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 521–532, 2012.

France et al. in [2], which addresses the problem based on improving the quality of the image processing; involving the processes of detection and classification of pollen grains. There have been many studies which model the features of pollen grains, in order to obtain a better classification process, which works as basis to understand the improvements made so far in the field. In [3], the shape and ornamentation of the grains were analyzed, using simple geometric measures. Measurement of textures using concurrence matrices were used by P.Li et al. in [4], using this measurements for automated identification process with optical images of pollen grains. Some others researchers have combined both techniques described above in their experiments as in [5] and [6], which used an approach of a Multi-Layer Perceptron neural network. In [7] brightness and shape information were used as vector features to improve its classification system. Besides these approaches, there are a vast number of methods used nowadays. An exciting approach is found in [8], which consists in a combination of statistical reasoning, feature learning and expertise knowledge from the application. Feature extraction was applied alternating 2D and 3D representations. Iterative refinement of hypotheses was used during the classification process. Another interesting method describes a full 3D volume recorded database of pollen grains using a confocal laser scanning microscope as found in [9]; this method extracted 14 invariant gray-scale features based on integration over the 3D Euclidian transformation used for classification. A more recent work with an ambitious approach is shown in [10]. It describes an automatic optical recognition and classification of pollen grains system. This method is capable to locate pollen grains on slides, focus and capture them in pictures, and finally identify the species applying a trained neural network.

There is no doubt the field of computer vision in biology has been made a lot of progress; nevertheless, more knowledge is needed combining different types of features so the number of species that systems can recognize and classify increments considerably. This work introduces a hybrid classification approach using Hidden Markov Models to classify successfully the contour of pollen grains images due to the results of previous works in different scenarios that has demonstrated that HMM is successful in the image contour classification area. The hybrid proposed system is based on image signal processing and decisions support systems, taking the advantages of both techniques to create a useful bioinformatics system to classify species. Moreover, this work introduces a contour analysis of pollen grains using a verification approach where the images are analyzed and classified by a HMM classifier. Two approaches have been used, identification for a first stage and verification for a second stage. The first stage consists of a learning process where the systems let us identify and classify species; towards that classification, a verification process is done to confirm the correct recognition of tropical honey bee's species, in comparison to previous work where identification was the core approach. This approach has given a 72.92% of success rate in the identification stage, but its success rate has considerable increased to 98.77% thanks to the novelty use of a verification process in the recognition system, being the first work in the area to use a verification stage. We consider that the verification stage we have incorporated to the system makes it unique and accurate due to the reliability in species recognition. This stage allows the system to recognize and classify successfully species of tropical honey bee's plants that are present in the database used. Compared to previous works, if a pollen grain is not documented in the database, those systems tend to fail in the identification stage.

Nevertheless, the incorporation of a verification process let the system successfully indicate that the pollen grain does not belong to a trained class; this way we can understand bee's behavior and if those bee's start to produce honey with different pollen grains. Furthermore, this work presents an angular parameterization which provides a contour invariant characterization compared to other approaches, removing the problems of scale, translation and rotation in images. Regarding the biological aspect, working with pollen grains contour supposed to have 2 relevant features, the variability and the similarity between its species. Likewise, the proposed method in this paper encompasses the classification using an HMM classifier due to the benefits it presents in pattern recognition as in temporal shape recognition, speech recognition, handwriting and gestures patterns, bioinformatics and others. The HMM statistical grounding gives the freedom to manipulate the training and verification processes as required by the problem, and give a mathematical and statistical analysis of the results and processes. Moreover, HMM classifiers provide a modular approach thanks to the modularity of its architecture, allowing the combination of HMMs into larger HMMs improving significantly the classification results. Furthermore, HMM allows incorporating prior knowledge into its architecture, letting initialize the model close to something believed to be correct and constraining the training process. The remainder of this paper is organized as follows. How the images were preprocessed is being described in Section 2. Section 3 describes the feature extraction process followed to obtaining the relevant information of the pollen grain contours. The HMM classification system used is described in Section 4. Then, Section 5 contains the experimental settings applied in this research; as are the database, the experiment methodology and the results obtained. Finally in Section 6, the conclusions of this work are shown.

2 Image Preprocessing

As can be seen in figure 1, it has been used a colorant to make the samples of each pollen grain more visible. Thus, the preprocessing was vital in order to obtain a good parameterization. The preprocessing steps are described above, those steps include: Binarization Process, Noise Elimination Process and Contour Extraction.

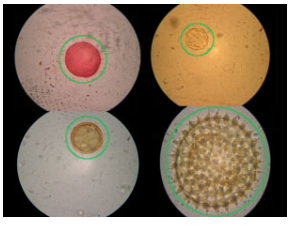


Fig. 1. Images to work with. Various kinds of pollen grains surrounded in green circles

Most pollen grain contour detection algorithms were proposed from binary images, which make it appropriate to transform a gray scale (or color) image to a binary one, allowing reducing data volume to process. One of the used methods to obtain a binary image is through its histogram, obtaining the number of pixels related to each level of gray which appears in the image. The binarization process must choose an adequate

value in the range of gray levels (a threshold), all the gray levels inferior to that threshold would transformed to black and all the gray levels superior to that threshold would be transformed to white; being this process known as Otsu Method[11].

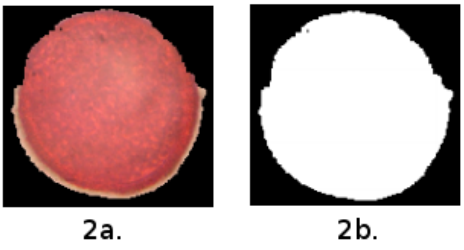


Fig. 2. Binarization Process. 2a) Pollen grain image, 2b) Binarized image

Once the images are binarized, the next step would correspond to eliminate the noise in the image, which would be removing any part of the image that does not belong to the pollen grain, like small bubbles created due to the liquid used to give color to pollen grains. The noise elimination process consists in using as reference the binarized image of the pollen grain. In this point, it is necessary to locate which part of the image corresponds to the pollen grain. To achieve this, we obtain the center pixel value of the whole image; if this value is equal 1 it means the pixel taken corresponds to the inner pixels of the pollen grain; if not, then we move to the right and take a new pixel value until we find the pollen grain. Once the pollen grain has been localized, it is used a selection algorithm which is useful to eliminate all the pixels that does not belong to the pollen grain. This algorithm consists in applying a mask using the binarized image of the pollen grain; which consist in taking the image and selecting the objects found in the image. Among those objects, the object which covers more than 90% of the image is the dominant, so the rest of them are excluded, obtaining a cleaner image and the pixels that correspond just to the pollen grain object. These results can be seen in figure 3.

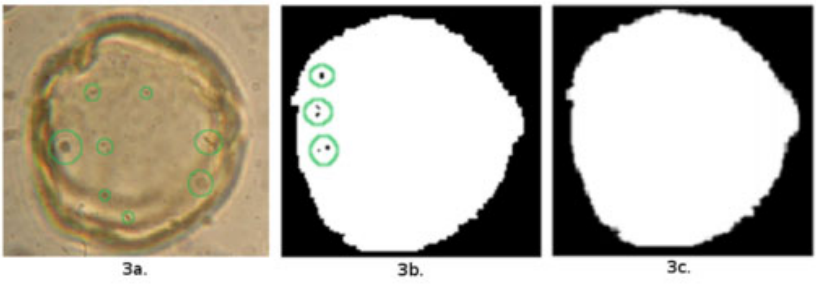


Fig. 3. 3a) Pollen grain image with colorant bubbles surrounded in green circles, 3b) Binarized image with some noise surrounded in green circles, 3c) Image without noise

Contour extraction is used in computer vision to recognize objects in 2 dimensions [6] [12], region segmentation and object's border extraction. To obtain the border of an image, there are a vast number of methods to achieve this goal like Sobel method,

Prewitt method, Roberts's method, Canny Method, among others. In order to obtain the border of the pollen grain, it has been applied the Canny algorithm, finding out that it is not efficient enough in terms of how fast the border extraction is done.

Therefore, the method of extraction was changed to morphological operations. In this case, the image of the pollen grain and the interior pixels are removed, so the border can be obtained. To achieve this, the pollen grain pixels are taken an analyzed this way: If a pixel has all its 4-connected neighbors with a 1 value, then this pixel is set to 0. This helps us to process and leave only the boundary of the image with the pixel value in 1. This method is more efficient than Canny algorithm in terms of how fast the border extraction is done, and allows to obtain the contour of the image due to the image skeletonization; obtaining the results seen in figure 4.

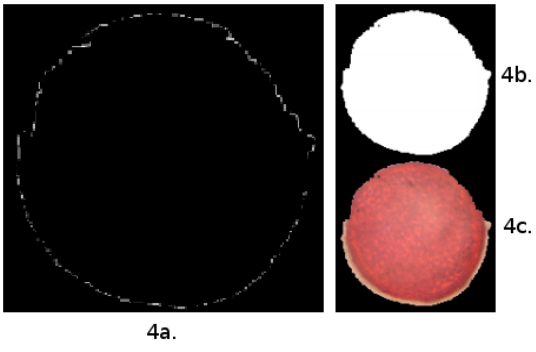


Fig. 4. 4a) Pollen grain contour, 4b) Binary noiseless image, 4c) Original Pollen grain image

3 Feature Extraction

In order to make a good feature extraction process, a set of steps has been proposed to locate a set of features of every image. Once the contour of the pollen grain has been obtained, the first thing that should be done is make the contour points selection as shown in figure 5b and obtain a set of points organized in a clockwise manner keeping the circular form of the contour.

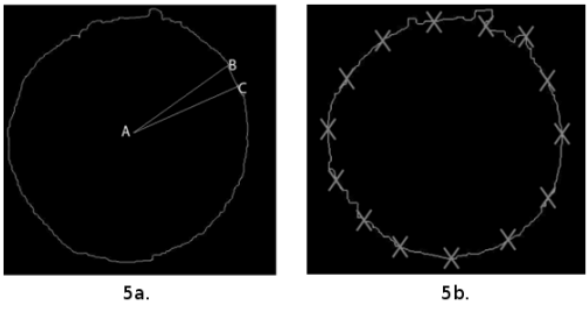


Fig. 5. 5a) Location of angles, 5b) Set of points

After the set of points has been calculated, the centroid of the object is obtained with formula (1), (2) and (3) where the *n* vertices are defined by *(xi,y_i)* and the centroid correspond to *(Cx, Cy)*. With this information, a triangle can be formed using the centroid of the pollen grain been this point the A vertex of the triangle, and the B and C vertex would be 2 neighbor points of the contour. Using the Pythagoras' theorem and Cosine' theorem, the inner angles of the triangle can be obtained.

$$C_x = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{1}$$

$$C_y = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{2}$$

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i) \tag{3}$$

Once the angles are obtained, those values are stored and a new triangle is formed; keeping the centroid as the A vertex, the new B vertex value would be the same value of the C vertex of the last triangle, and the new C vertex value would be the next neighbor point of the B vertex. In other words, the B and C vertex are 2 neighbor points of the contour taking those points in a clockwise manner as shown in figure 5a.

The angles used in this feature extraction are angular rather than polar, taking the advantage of obtaining information of the contour independent of the size, translation and rotation of the pollen grains. Polar angles are not a good choice since polar angles gives information dependent of the contour size, being this a relevant factor due to the importance of a good classification in which the size does not predominate. This parameterization is used to treat discrete systems like the Hidden Markov Model (HMM) classifier.

4 Classification System

The classification system proposed to accomplish the recognition of the pollen grains from the features obtained is based in the Hidden Markov Model (HMM). A HMM is a string of states $\boldsymbol{q}$, jointed with a stochastic process which takes values in an alphabet S which depends of $\boldsymbol{q}$. These systems evolves in time passing randomly from one state to another and issuing in each moment a random symbol of the S alphabet.

When the system is in the state $\boldsymbol{q_{t-1} = i}$, it has a probability $\boldsymbol{a_{ij}}$ of moving to the state $\boldsymbol{q_t = j}$ in the next instant of time and the probability $\boldsymbol{b_j(k)}$ of issuing the symbol $\boldsymbol{o_t = vk}$ in time $\boldsymbol{t}$. Only the symbols issued by the state $\boldsymbol{q}$ are observable, nor the route or the sequence of states $\boldsymbol{q}$; that's why the HMM obtain the appellative of "Hidden" since the Markov process is not observable.

We have worked with an HMM called "left to right" or Bakis, which is particularly appropriate for sequences. These "left to right" HMM's turn out to be especially

appropriate for hand edge because the transition through the states is produced in a single direction, and therefore, it always advances during the transition of its states. This provides for this type of model the ability to keep a certain order with respect to the observations produced where the temporary distance among the most representative changes.

5 Experimental Settings

In order to implement the classification system based on HMM classifier, a set of experimental settings has been configured. These settings includes the Database used in all the experiment scenarios, the methodology followed to achieve a good classification result, and the analysis of the results obtained.

5.1 Database

The database used has been obtained directly from the Research Center of Tropical Bees, CINAT (Centro de Investigación de Abejas Tropicales from its Spanish acronym), of the National University of Costa Rica (UNA); located in Heredia, Costa Rica. This database contains a total of 426 pollen grain sub-images corresponding to 17 genders and species from 11 different families of tropical honey plants from Costa Rica. This database has been used in previous research work done as in [14].

5.2 Experiment Methodology

The experiment methodology has been structured using a supervised classification technique which consists in using a set of pre-established knowledge to determine the best classification scenario. Therefore, this experiment is divided in two stages. The first stage describes the learning process based in HMM whereas the principal objective is to obtain the best model, which is the one capable of classify the contour of the pollen grain in every herbage. And the second stage represents the process of verification of the pollen grain contour used by the models generated in the learning stage, as shown in figure 6.

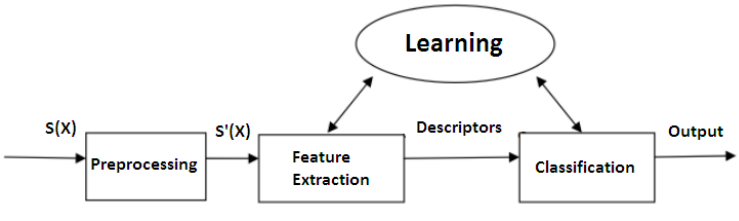


Fig. 6. Classifier procedure

Learning Process: It has been trained 41 models (one for each species), so the training was done with 17 different pollen grain contours coming from each different species. This training was done with the training function to which it sends as a parameter the observation information obtained in the feature extraction process. This function defines a state transition matrix and an initial observation symbols matrix.

The number of iterations in the HMM is 1681 using 17 types of pollen grain contours, this values corresponds to the number of observations obtained of each pollen grain contour. The behavior of the HMM classifier consists in comparing the input feature information vector corresponding to a determined class, with a set of pattern features vector. At least, there should be one pattern feature vector for each of the classes. The feature vectors of each class are introduced to a single HMM. For example; in class 1 there should be an input of the feature vectors corresponding to individual 1, in class 2 there should be an input of the feature vectors corresponding to individual 2, and so on. This way, a single HMM that contains the information of all the individuals is generated. Once the HMM is ready, the test mode can be pursued. In this mode, the system receives as input a feature vector and the system compares it with model generated in the learning stage. This comparison takes the maximum of the HMM with the output and the identified class. If the output exceeds this maximum, the input feature vector belongs to an individual in the database; otherwise the input feature vector belongs to an individual outside of the database.

Verification Process: The HMM classifier is treated as a bi-class classifier. It should be created as many models as there are classes, being each model composed by the feature vector of a class to which it is considered as positive and all the remaining feature vectors to which there are considered as negative. In the test mode of the verification process, the image to be classified is preprocessed according to the method described in Section 2, and then the features of this image are extracted according to the process described in section 3. The feature vector of the image to be verified is calculated according to the parameterization method chosen in each case. If the output exceeds a determined threshold, the pollen grain to be verified would be verified as from the class to which the individual is said to belong. Otherwise, the individual is rejected. The threshold obtaining process is based in the False Rejection Rate and the False Acceptance Rate. According to these concepts, the ROC curves are created, which indicates the system reliability. A high FRR indicates a decision threshold too strict provoking dissatisfaction from the system users due to the negation of authorized personal. Otherwise, a high FAR indicates a decision threshold too permissive, generating a big security problem. The important thing is to obtain an optimal threshold [13]. Representing the values of the rates in accordance to the decision thresholds, it can be obtained a TEER (Threshold of Equal Error Rate) and an EER (Equal Error Rate) of the system. The TEER refers to the threshold of EER, whereas the FAR and FRR are equal and the EER is the system error. This means that the TEER and EER are generated where the cutoff point between FAR and FRR is. Once the EER is calculated, the reliability of the system can be calculated through the following expression:

$$System\ Reliability\ (\%) = 100\ EER(\%) \tag{4}$$

6 Results and Analysis

In this section, it is presented serious different tests made with all the feature vectors and the classifier. Showing both the results of the identification process and the results of the verification process; providing an approach of which feature extraction

methods and configurations of the classifiers offer a better Total Success Rate (TSR). Once all the results are obtained, it can be verified that the pollen grain contour system can identify and verify the individuals in a database, offering an acceptable success rate. The tests performed were made with a database of 43 classes with 17 samples of each class. In the tests made with the classifier, it has been used 10 samples for the system training and the remaining 7 samples for test purposes. The results of the tests are presented in tables were the column TSR shows the success rate in terms of mean and variance.

6.1 Results for the Identification Process

The experiments performed and its results obtained with each of the feature vectors calculated in the feature extraction process for the identification system using HMM are exposed in this section. The tests performed for the identification process were repeated a total of 16 times.

With this method, it was selected a determined number of contour points separated the same distance between them. The tests were performed for different number of contour points. The number of points selected for the tests comprises a range of 200 to 400 contour points, taking into account that the number of points is expressed in the form of which it takes 1 point and skips 16 points. The states of the HMM has been varied in a range of 100 to 200 states. This variation in the number of states of the HMM and the number of contour points selected has been used to create a high number of test scenarios, indicating that overpassing those boundaries, the results tends to decrease considerably making these values a good set to reach variation in the results during experimentation of the proposed system.

Based on those experiments, it has been found that the best combination results are those who have an approximate relation of 2 contour points by each HMM state. It has been obtained a large number of results related to the variation in the number of states of the HMM and the number of selected contour points. From these results, it can be noticed that while the number of points chosen increase the results decrease, not being optimal results; as we can see in table 1. Among all the experiments done, it has been found that the combination of 280 selected contour points with 140 number of states for the HMM has given the best results in the identification process of the classes of the pollen grains as shown in table 1.

Table 1. Identification of pollen grains contour

Number of Points Chosen	Number of States HMM	TSR Mean%±std
350	180	54.17±5.89
300	160	70.83±5.90
280	140	72.92±2.95

In the classification process, it has been used a classifier based on HMM. Knowing that in most cases, by not specifying that all, the pollen grains have a circular form regardless of the termination of its class noting 2 different types of pollen shapes. This classification has given a good result which it helps to explain some issues: a)The number of points to be selected is a relevant factor since choosing a lower number of points in comparison to the optimal, the shape of the pollen grain could be lost converting it into a simple circle; and choosing a higher number of points would give us a lot of details in the pollen grains, causing worst results due to the memorization of the shape and not being able to recognize the shape pattern of pollen grains of the same species. This way, choosing the number of points to be selected is a crucial task so this number can create a balance between generalization and memorization allowing the system to correctly identify and classify the species. And, b) By introducing a higher number of points, the results would be impoverished due to the inclusion of noise to the classifier, giving bad results as seen in table 1.

6.2 Results for the Verification Process

The experiments for the verification process were performed only with the method were the best result has been obtained. Thanks to the identification system, the best result scenario has been obtained due to the combination of 280 selected contour points with 140 number of states for the HMM. Therefore, this result is used as a basis for the verification process. As a reminder, while in the identification process it was created just a single model with all the classes in it, for the verification process it is created a model for every class in the database where each of this models corresponds to a different class in the database. Each one of these models is evaluated, and the results are obtained varying the output threshold, obtaining the FAR and FRR for every value of the threshold. These rates are obtained analyzing the number of pollen grains images falsely accepted and the number of pollen grains images falsely rejected depending on the threshold. From the representation of FAR and FRR, the EER and the TEER can be obtained. As shown in figure 9 and table 2, the results of EER, TEER and the Reliability of the system for the performed tests for all the classes in the database, are calculated according to the equation given in section 5.2, it can be considered that a verification process can be done with a high level of credibility.

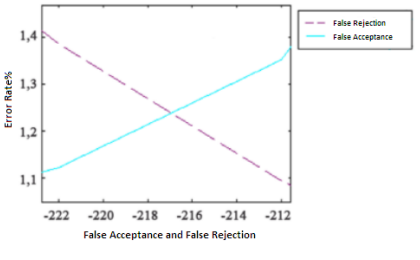


Fig. 7. Cutoff points for the verification

Table 2. EER, TEER and Reliability of the system obtained with HMM

EER	TEER	Reliability
1,23%	-217	98,77%

7 Conclusions

This work has introduced a classification system through the pollen grain contour, being capable of performing the recognition and verification of the pollen grain. The preprocessing stage has been a little bulky in the process of finding the appropriate threshold for the binarization of the image. The ideal method would be to calculate the threshold for each image, but this couldn't be done due to the obtainment of the images captures. The solution presented in this paper was performed by obtaining the mean of the thresholds establishing this mean as the binarizarion threshold. In the feature extraction stage, 2 methods of parameterization have been proposed; obtaining angular angles and polar angles. For this 2 methods, it has been obtained a better result using the angular angles because these values does not depend on the size of the pollen grain, gaining a better TSR result. The success rates in identification are around 72.95% with a variance of 2.95 and a trust interval of 0.5; having for the verification process a reliability of 98.77% working with our database. This shows that an approximate relation of 2 contour points by each HMM state, is the optimal choice to model the contour of pollen grains, letting the system identify and validate the different species of tropical honey bees. It is important to mention that this classification method proposed gives a better result rate than the methods presented previously by other authors like [4], [5], [6] and [7]. Moreover, this paper has contributed with a taxonomical quantitative model, showing that the pollen grain contour classification system can simplify the features in the moment of its classification obtaining a sufficient discriminatory result.

Acknowledgements. This work has been supported by Spanish Government, in particular by *"Agencia Española de Cooperación Internacional para el Desarrollo"* under funds from D/027406/09 for 2010, D/033858/10 for 2011 and A1/039531/11 for 2012.

References

1. Li, P., Flenley, J.: Pollen texture identification using neural networks. Grana 38(1), 59–64 (1999)
2. France, I., Duller, A., Duller, G., Lamb, H.: A new approach to automated pollen analysis. Quaternary Science Reviews 18, 537–536 (2000)
3. Treloar, W.J., Taylor, G.E., Flenley, J.R.: Towards Automation of Palynology 1: Analysis of Pollen Shape and Ornamentation using Simple Geometric Measures, Derived from Scanning Electron Microscope Images. Journal of Quaternary Science 19(8), 745–754 (2004)

4. Li, P., Treloar, W.J., Flenley, J.R., Empson, L.: Towards Automation of Palynology 2: The Use of Texture Measures and Neural Network Analysis for Automated Identification of Optical Images of Pollen Grains. Journal of Quaternary Science 19(8), 755–762 (2004)
5. Zhang, Y., Fountain, D.W., Hodgson, R.M., Flenley, J.R., Gunetileke, S.: Towards Automation of Palynology 3: Pollen Pattern Recognition using Gabor Transforms and Digital Moments. Journal of Quaternary Science 19(8), 763–768 (2004)
6. Rodriguez-Damian, M., Cernadas, E., Formella, A., FernandezDelgado, M., De Sa-Otero, P.: Automatic detection and classification of grains of pollen based on shape and texture. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 36(4), 531–542 (2006), doi:10.1109/TSMCC.2005.855426
7. Rodriguez-Damian, M., Cernadas, E., Formella, A., Sa-Otero, R.: Pollen classification using brightness-based and shape-based descriptors. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, August 23-26, vol. 2, pp. 212–215 (2004)
8. Boucher, A., Thonnat, M.: Object recognition from 3D blurred images. In: Proceedings of 16th International Conference on Pattern Recognition, vol. 1, pp. 800–803 (2002)
9. Ronneberger, O., Burkhardt, H., Schultz, E.: General-purpose object recognition in 3D volume data sets using gray-scale invariants - classification of airborne pollen-grains recorded with a confocal laser scanning microscope. In: Proceedings of 16th International Conference on Pattern Recognition, vol. 2, pp. 290–295 (2002)
10. Allen, G.P., Hodgson, R.M., Marsland, S.R., Flenley, J.R.: Machine vision for automated optical recognition and classification of pollen grains or other singulated microscopic objects. In: 15th International Conference on Mechatronics and Machine Vision in Practice, M2VIP 2008, December 2-4, pp. 221–226 (2008)
11. González, R.C., Woods, R.E.: Digital image processing, 2nd edn. Prentice Hall, Upper Saddle River (2002)
12. Chen, Y.W., Chen, Y.Q.: Invariant Description and Retrieval of Planar Shapes, Using Radon Composite, Features. IEEE Transaction on Signal Processing 56(10), 4762–4771 (2008)
13. Bricego, M., Murino, V.: Investigating Hidden Markov Models Capabilities in 2D Shape Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), 281–286 (2004)
14. Ticay-Rivas, J., Pozo-Baños, M., Travieso, C., Arroto-Hernández, J., Pérez, S., Alonso, J., Mora-Mora, F.: Pollen Classification basedon Geometrical Descriptors and Colour Features using Decorrelation Stretching Method. In: Proceedings of 12th INNS EANN-SIG International Conference, EANN 2011 and 7th IFIP WG 12.5 International Conference, AIAI 2011, vol. 2, pp. 342–349. Springer, Heidelberg (2011) ISSSN: 1868-4238, ISBN: 978-3-642-23959-5

Modelling Stress Recognition in Conflict Resolution Scenarios

Marco Gomes, Davide Carneiro, Paulo Novais, and José Neves

Department of Informatics, University of Minho
pg18373@alunos.uminho.pt, {dcarneiro,pjon,jmn}@di.uminho.pt

Abstract. The current trend in Online Dispute Resolution focuses mostly on the development of technological tools that allow parties to solve conflicts through telecommunication means. However, this tendency leaves aside key issues, namely the context information that was previously available in traditional Alternative Dispute Resolution processes. The main weakness of this approach is that conflict resolution may become focused solely on objective issues. In order to overcome this inconvenience, we move forward to incorporate context and behavioural information in an Online Dispute Resolution platform. In particular, we consider the estimation of the level of stress and the prediction of the stress state evolution. As a result, the conflict resolution platform or the mediator may predict to what extent a party is affected by a particular matter, allowing one to adapt the conflict resolution strategy to a specific scenario in real time.

Keywords: Hybrid Artificial Intelligence Systems, Online Dispute Resolution, Stress, Cognitive Activation Theory of Stress.

1 Introduction

Online Dispute Resolution (ODR) is a form of dispute resolution that takes place partially of wholly in a digital environment [3]. The use of technological solutions in this field is nowadays well established. However, the current trend continues to focus mainly on the development of technological tools. As a result, the actual ODR systems leave aside important issues that are present in traditional dispute resolution processes, namely, the context-dependency issue. This issue has a preponderant role in human behaviour. The omission of context and behavioural information can influence the course of action and, consequently, the outcome of a conflict resolution scenario. The use of a synergistic combination of multiple Artificial Intelligence techniques [9] [10] and, more particularly Ambient Intelligence techniques, can help to suppress this lack.

In order to address this challenge this work aims to develop mechanisms that operate in the virtual environment of ODR to collect context information and perceive the activities being performed. Basically, the underlying intent is to extend the traditional technology-based conflict resolution, in which a user simply interacts with the system, with a new component: an intelligent environment. These environments are pervasive and transparent, i.e. a person should not perceive the environment in any other way than by the actions it executes. This is important since when people are aware that they are under monitorization, they tend to behave differently.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 533–544, 2012.

With this approach it is expected to provide useful context knowledge to support the lifecycle of the conflict resolution model. The main objective is to capture context information that can be used by conflict resolution platforms to achieve better and more satisfactory outcomes for the involved parties [7]. This will include the estimation of the level of stress, the attention level, behavioural patterns, among others. Being able to make assumptions about the information that can be used to characterize a person's situation is extremely relevant to assist a conflict resolution platform finding the path to achieve successful outcomes [5]. From the point of view of the (electronic or human) mediator, the access to this information is also vital in order to plan the right strategy and perceive how each issue affects each party, much like it is done in traditional conflict resolution processes, when parties meet face-to-face.

2 The Stress Interpretation and a Multimodal Approach

Many experts endorse the original definition of stress concept to the proposed by Hans Selye [21], who coined the term as it is presently used. He defined the stress like a non-specific response of the body to any demand placed upon it. Selye conceived of external demands as *stressors* (the load or stimulus that triggered a response) and the internal body changes that they produced as the *stress response* (triggered by a load or a stimulus). He was the first person to document the chemical and hormonal changes that occur with stress. However the definitions provided were not conclusive for the entire scientific community. The concept of *stress* is still an open discussion in the scientific community. The main reason for this lies in the multiplicity of factors and the subjective nature of stress phenomenon, which led to multiple interpretations. With so many factors that can contribute to stress it can be difficult to define it. In this open discussion some argue that the *stress* concept is elusive because it is poorly defined [11] and others prefer to not define the concept until stress research reach a consensual significance. Meanwhile, scientists have circumvented the problem of a clear and agreed definition of stress by defining it empirically. It's difficult to measure stress if there is no agreement on what the definition of stress should be, but this does not prevent the advancement of science in this area but this has not stopped the advance in this field. Researches start to focus upon cognitive and behavioural causes for stress, and stress became viewed as a mind-body, *psychosomatic*, or psycho-physiologic phenomenon. A free interpretation of this phenomenon could refer stress as a physico-physiologic arousal response occurring in the body as result of stimuli, and these stimuli become a *stressor* by virtue of the cognitive interpretation of the individual. This will be our interpretation of stress that will serve as a cientific starting point to modelling stress in the conflict resolution scenarios.

Some experimental results [18] demonstrate that single-modality features are not sufficiently precise for stress recognition, while the integration of multiple-modality features are very helpful for accurate stress recognition. The following generic diagram (Fig. 1) represents a multimodal approach to the stress recognition problem. It consists of two portions, the leftmost portion, from left to the "stress" node, depicts the elements that can alter human stress. These elements are included in "context" node and represent the generalization of two main categories (sources of contextual information): the

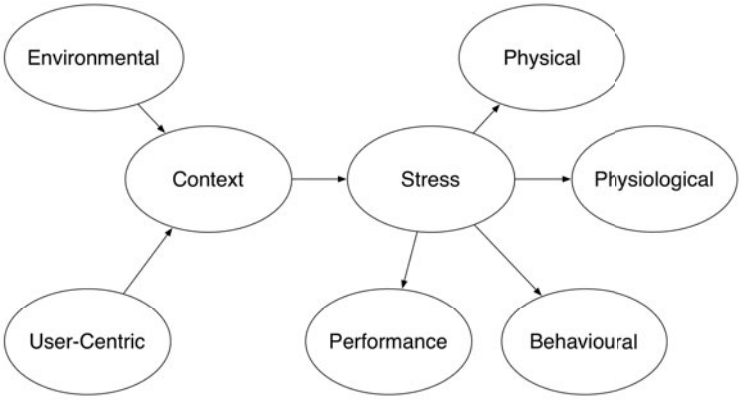


Fig. 1. A generic diagram for representing multimodality space in the recognition stress model. Due the space limit we don't draw all the features considered

"user-centric" context and "environmental" context. On the other hand, the rightmost portion of the diagram, from the "stress" node to the right portion, depicts the observable features that reveal stress. These features include quantifiable measurements on the user's physical appearance, physiology, behaviours and performance.

The "context" node is divided in two types according to the source of contextual information, namely, the "user-centric" and the "environmental" context. User-centric information is composed of two categories: the background and the dynamic behaviour. The background is composed by several attributes that can be extracted from the user's profile. These attributes are the age, gender, working area, social status, personality traits among others. The dynamic behaviour reflects the contextual attributes related to the user's activity within the conflict resolution platform. These can be depicted by the intention of the user (if he/she really wants to achieve an outcome) and by the activity (if the user is an active or passive party in the process). The "environmental" information fuses the physical environment characteristics, social environment information and computational environment measurements. Physical environment includes attributes such as the time, temperature, location, noise level, and luminance. High levels of noise, extreme temperatures and low levels of luminance are well known potential stressors. The social environment is related to the social conditions when a user interacts with the system, like the population density when a user interacts with the platform (number of surrounding people per unit of area). The computation environmental context can be characterized by the measurement of the electromagnetic field and the number of surrounding electronic devices.

Among the features that can reveal stress, those that can characterize the "behavioural" node are the user's conflict style [6] (a conflict style can be a coping strategy in response to stressful conflict [24]), his/her interactions with the computer, the mouse/touch screen pressure from clicks/touches, his/her agitation level (through the sensory data from the accelerometer placed in mobile devices or by analyzing movement), as well as input frequency and speed. Also, the "performance node" is depicted in terms of accuracy

and response, where the accuracy feature is related to the amount of touches in particular areas in the platform interface and the response feature corresponds to the analysis (qualitative and temporally) of user's responses to the conflict resolution demands. The physiological variables provide observable features about the user's stress state [20]. These features can be Galvanic Skin Response (GSR), that assesses the electrical proprieties of the skin in response to different kinds of stimuli, and General Somatic Activity (GSA) that assesses the minute movement of human body and many others such as respiration, pupilographic activity among others. Physical appearance includes the visual features that characterize the user's eyelid movement such as pupil movement (e.g. eye gaze, papillary response), facial expression or head movement. In addition, this is flexible enough to allow the insertion and modification of variables. For example, the variables under the behavioural node may vary with different applications, and the variables under the physiological node may change, depending on the availability of the requiring measuring devices.

3 Dynamic Stress Recognition Model

The ability to recognize human stress can have a determinant role in conflict resolution process. Having the information about the level of stress of the disputant parties is quite important in order for a mediator to correctly understand how each issue or event is affecting each party.

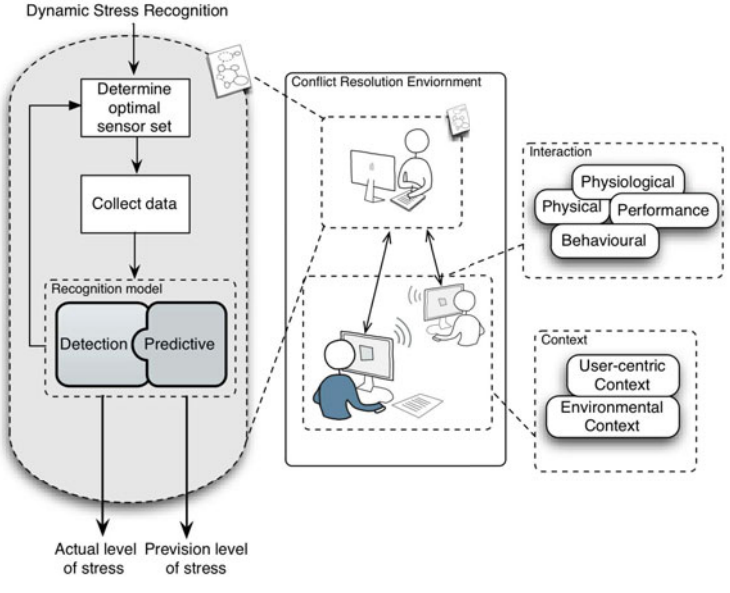


Fig. 2. An overview of the dynamic stress recognition system embedded in an intelligent environment

Various approaches have been developed to recognize and detect user stress [22] [15] [4] [18]. The approaches or systems differ from each other in either the evidence modalities, or detection techniques, or both. Overall, our approach differs from the cited ones in that it employs the dynamic techniques to unify detection with stress prediction, utilizes evidences from multiple modalities, and is validated with psychology theories.

In order to enrich the conflict resolution process with this ability, one must consider the detection and prediction of stress evolution in real-time. These two complementary modules compose the stress recognition model. The first one, tries to measure in each time instance the stress, and the second gives an outlook as stress is evolving, based on the Cognitive Activation Theory of Stress (CATS). Based on this information, the (electronic or human) mediator is able to perceive how the state of each party is evolving. Figure 2 streamlines the mains procedures in applying the dynamic stress recognition to conflict resolution environment. At each time t, the platform performs three procedures - sensor optimal set, stress recognition, and returns information. Specifically, the system decides an optimal sensory action set to collect data with a sensor selection strategy. The collect data are propagated through the stress recognition model. Through this model, the stress detection module computes the stress value at time t and the stress prediction model (based in CATS) determines the tendency of stress evolution for time $t+1$. After the provision of information to the users, their stress levels may change and new evidences need to be collected. Thus the system goes to the next step and repeats the three procedures.

3.1 The Stress Detection Module

Sensor measurements inherently incorporate varying degrees of uncertainty and are, occasionally, spurious and incorrect. It is extremely important to select just the necessary set of sensors that can provide the more accurate and consistent data in each moment. To fulfil this restriction, the system must select, in each phase of the stress recognition process, a subset of sensors that is the most relevant and cost effective. Determining the most informative and cost effective subset of sensors requires an evaluation of all possible subsets of sensors, which is computationally intractable. To address the problem of sensor selection on the application layer we base our approach in Yongmian Zang and Quiang Ji [28] sensor selection methodology. Specifically, the authors propose four steps: (1) a Bayesian Network to represent the sensor and their dependencies; (2) a statistical measure to quantify the pairwise sensor synergy; (3) the construction of a synergy graph; and (4) a greedy algorithm is applied to identify the near-optimal subset of sensors. Some experiments were performed [28] and demonstrated that the least upper bound can approximate closely of the mutual information value. This permits to circumvented the computationally intractable problem of assess the mutual information value of all variables. The presented sensor selection approach reduces the computation time with minimum loss accuracy, comparing with others methods like the random methods and the brute force approach. In the other hand, this methodology, as presented for Yongmian Zang and Quiang Ji [28], have some limitations. Namely, the assumption of all sensors have the same cost benefice and the non-existence of any conditionally dependence between them. These limitation can be considered strong barriers to implementing a real-time conflict resolution system. Thus, some adjustments will exist in the future work.

Processing of Sensor Data. The management of the raw data captured by the selected sensors is a challenge in stress recognition, as well in many other scenarios. The differences among evidences (observable features that are capable of providing clues about the person's stress state) and within the same evidence can be from temporally and spatially separated experiences. To overcome this, the processing of sensor data must be well managed in order to make the information and accuracy losses minimum in the process of transforming raw data. In ODR systems events can occur within very different temporal windows (e.g. a party may wait for a proposal a few minutes or a few hours or days). In that sense we defined dynamic temporal windows. Instead of predetermining a static value for the duration of the windows, we support that the temporal windows must be adaptive, i.e this approach is event-oriented. It is clear that independently of the size of the temporal window, the system continues to select and record the sensor measurements. The difference resides in the times in which the other stages of the stress recognition system are fed. In our view, the most appropriate (considering the ODR characteristics) time is the interval between events that occur in the platform. In the remaining of the document, the letter t will denote the previously referred temporal window.

Feature Extraction. For each feature, a different extraction technique can be used. However, some techniques can be applied to more than one feature. For example, when there is a loss of data an interpolation method (curve fitting or regression analysis) could be used to fill the gap. Due to the specific sensory data that derives into multiple and different values, we just highlight some of the features considered. The values of physiological features like the GSR and the Pupillary Activity (PA) can be obtained by the analysis of the mean value of the GSR and PA measurements. Visual feature extraction starts with eye detection and tracking, which serves as the basis for subsequent eyelid movement monitoring, gaze determination, and face orientation estimation. Behavioural features can be elicit through the analysis of the negotiation activity and the parties' interactions with the platform. Context features can be extracted from the party's profile using a simple form to capture the age, gender, working area, social status, personality traits, among others. After this process, all the features are normalized to a value in the range [0, 1] using max-min normalization. Not all the values of features need to update further stages of the stress recognition model. This is due to the fact that these values may not vary much between time $t-1$ and t and, also, because some features assume static values. For example, the contextual information has some attributes that will not vary during the session in the conflict resolution platform, like the gender, the age, the work occupation among others. Thus being, the extraction techniques will only act after the verification of a significant variation in these parameters. This verification must be performed by an Analysis of Variance technique (e.g. ANOVA) to performs the role of selecting, with a certain threshold, which values of features should be taken into account in computing the variable value.

The Figure 3 gives an overall vision of how sensory data is processed, from the near-optimal sensor selection to the analysis of variance. The main function is to provide normalized data to compute the stress value. In the next section, the variables (modalities) will be the normalized values of all features analysed in this process.

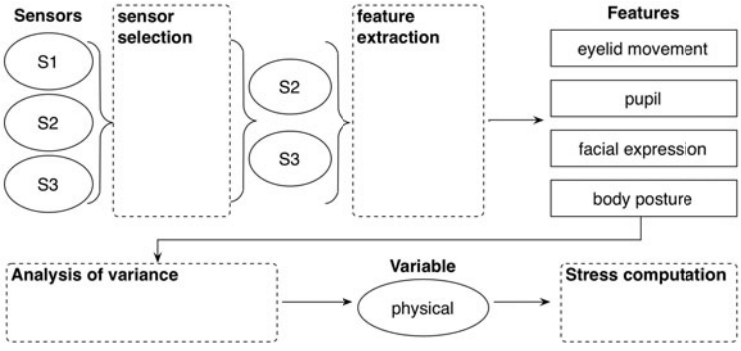


Fig. 3. The conceptual elements of the stress computation process

Decision Level Integration. To integrate all the multimodal evidences referenced in this work (see Sect. 2) we have chosen a decision level integration strategy. This strategy allows asynchronous processing of the available modalities, provides greater flexibility in modelling, and allows adaptive channel weighting between different modalities based on environmental conditions [27]. Our strategy uses a weighted sum to estimate the stress value with a temporal approach.

$$stress_t = \sum_{i=1}^{t-1} \left[c(i)_{wc} + b(i)_{wb} + p(i)_{wp} + ph(i)_{wph} + phy(i)_{wphy} \right] \quad (1)$$

In (1) we use the first characters of the modality (variable) name to refer the component that represents the sum of the normalized values from all the extracted features, i.e. c referring to context, p referring to performance, and so on. In addition, we subscript a variable by a step t to refer to the variable at time t. The w's, followed with the letters that represents the variables, denotes the weight of each component. At the beginning, the weights are equally distributed: every variable's weight is 0.2. Meanwhile, in the scientific literature the influence of each modality in the estimation of stress is not specified and has not been adequately explored yet. But this is not an irrelevant factor, in terms of accuracy of stress recognition systems. The relevance comes from the strong relationship between the interpretation of an individual's stress with the person and situation dependencies issues. To suppress this lack, we propose a novel approach to calculate the degree of influence by using Artificial Neural Networks (ANN). For each modality, an ANN (trained with synthetic datasets) is used in order to calculate the values of the weights. The input data includes all the values of all variables in time $t-1$ and the output must be an estimation of the degree of influence of each variable. This value must be within the range [0,1] and the sum of all estimations values at time t must be 1. The resulting values represent the rate variation for each variable used in the computation of the stress value at $t-1$. Indeed, this approach tries to give a dynamic performance in the calculation of the weights. Thus, we fulfil the requirement of respecting the person and case dependencies by suppressing, heuristically, the lack of ground-truth in this issue. However, optimization and validation are needed. So, we aren't closed to other options.

3.2 A Stress Prediction Module Based in CATS

The Cognitive Activation Theory of Stress (CATS) presents a formal set of definitions that tries to reduce the group of terms, which may cover the same stress phenomenon. Therefore, CATS offers meanings formulated in symbolic logic, which is a natural advantage (from a computational point of view) in an area so subjective as the stress phenomenon. This is *cognitive* stress theory because CATS assumes that the stress response depends on acquired expectancies of the outcomes of stimuli and available responses, which can be regarded as acquired (learned) relations between stimuli, and between responses and stimuli. It is an activation theory since it is based on neurophysiological activation and arousal concepts.

According to CATS, the stress response (alarm) depends on the individuals appraisal (evaluation) of the stress situation. Formally, CATS stipulates that the alarm occurs when there is a discrepancy (D in Eq. 2) between what is expected or the *normal* situation (set value) and what is happening in reality (actual value). The *normal* situation is understood as the classification obtained by collecting reference values from the tests performed by normal persons in a normal (non-pathological) state. Symbolically, it is the difference (Eq. 2) between the value a variable should have (set value SV), and the real value (actual value AV) of the same variable.

$$D = (SV - AV); \tag{2}$$

In our approach the predicted value at time $t+1$ is the discrepancy (D) value between the stress actual value (AV) obtained at time t and the CATS set value (SV) at $t+1$, during the events (e.g. send a proposal, receive a proposal, reject, accept, etc.) sequence on the resolution conflict lifecycle. The actual value of stress (AV) is calculated by the stress detection module (see Sect. 3.1) and the set value (SV) by Eq. 5. The level of alarm depends on expectancy (Eq. 3) of the outcome of stimuli and the specific responses available for coping. When the subject has learned (stored) that one stimulus ($S1$) predicts the occurrence of another stimulus ($S2$) this is referred to as stimulus expectancy. The stimulus expectancy (3) is expressed as ${}_{S1}E_{S2}$ and is equivalent to the stimulus expectancy $S1 - S2$, which means that $S1$ implies $S2$.

$${}_{S1}E_{S2} = (S1 \rightarrow S2); \tag{3}$$

$$H({}_{S1}E_{S2}) = SL_{S1} +_{S1} CT_{S2} +_{S1} PP_{S2}. \tag{4}$$

In our approach expectancies are quantified by several dimensions: acquisition strength, saliency of the event, and perceived probability. The acquisition strength of an expectancy (H) expresses that expectancies are acquired, according to the general principles of learning theory. This is calculated through the saliency of the event (SL), the contiguity of the event presentations (CT) and how often the events are occurring together (perceived probability PP). Primarily, the salience (SL) of an event can be achieved through an attention analysis, i.e., gathering the level of attention/interest when a party is experiencing the event. We can get this by using some elicited visual features like the gaze direction, the degree of eye open, and the size of pupil relative to

luminance to classify the attention/interest level that arises from the event. The contiguity of the events represents the distance (temporal or spatially) between two events. Through a machine learning technique, and using a dataset extracted of our ODR prototype, we are able to distinguish between a contiguous and non-contiguous event. The perceived probability expresses the probability of the expected event, as it is perceived by the individual. This is a subjective evaluation of the probability that we interpret as the probability of how often the events are occurring together. This is obtained by the conditional probability of two events occur successively in the same probability space. Finally, the H value is normalized by a max-min normalization technique (Eq. 5).

$$SV = H_{norm}({}_{S1}E_{S2}) = \frac{(SL_{S1} +_{S1} CT_{S2} +_{S1} PP_{S2}) - H_{\min}}{H_{\max} - H_{\min}}; \tag{5}$$

The discrepancy (D) can take three distinct values, a positive ($(SV - AV) > 0$), neutral ($(SV - AV) = 0$), and negative ($(SV - AV) < 0$), that are mapped into the following classification: increase, maintain, and decrease, respectively. So, the predictive module of the stress recognition process returns a stress outlook (decrease, maintain, or increase) for the next step of the conflict resolution process. However, our approach should be further developed in order to explore all the formalisms available in CATS.

4 Stress-Aware Conflict Resolution

The stress recognition model described previously is being used in the context of ODR. In fact, current research on this field mainly focuses on tools that can facilitate the compilation and exchange of information. This leaves aside very important information. Namely, information about the stress levels or the emotional state is simply not considered and is difficult to be transmitted over the telecommunication means currently used (e.g. chat rooms, e-mail). This not only makes it more difficult for parties to share their ideas but also to perceive feedback from our counterparts [17], making communication less effective. Moreover, when parties lack this information, they tend to disregard the fears and desires of each other (as they dont actually see “the person behind the screen”), being more prone to hurt or offend, thus making conflict resolution harder. In that sense, providing feedback about the state of each other is essential in order for parties to maintain a sense of reality and mutual respect. Based on this, we are applying this stress recognition model in our conflict resolution platform. The group of sensors and associated features currently being used includes:

- Touch intensity - higher levels of touch intensity are associated with increased levels of stress;
- Touch accuracy - this is a measure of the amount of touches in active controls versus touches in active areas. When people are nervous or stressed, the accuracy tends to decrease (one of the manifestations of stress are tremors);
- Accelerometer - when people are under stress, they tend to show more abrupt and sudden movements. This can be detected by accelerometers;
- Image processing - the amount of movement of a user can also be analyzed through image processing, allowing to estimate the level of agitation, thus stress;

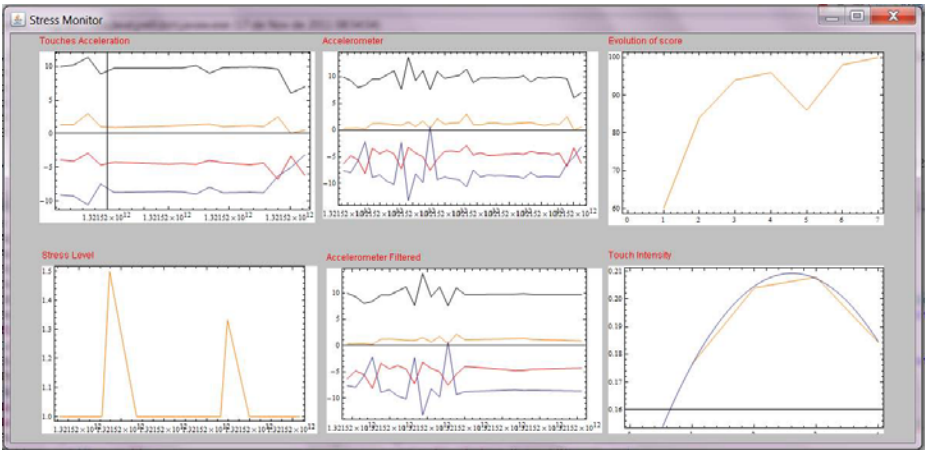


Fig. 4. Prototype of the interface of the Stress Monitor

- Touch patterns - calm touch patterns are generally different from stressed ones. Although touch patterns vary from an individual to another, they are affected by stress in in similar ways. This allows classifying touches as calm or stressed.

In order to have access to this information, we are using portable devices that are equipped with touch screens and sensors and act as interfaces for the conflict resolution platform. Moreover, video cameras placed in front of and around the user allow obtaining additional information. Evidently, not all these sensors may be available at the same time (e.g. some portable devices have capacitive screens, which dont support pressure measurement). In that sense, the afore mentioned techniques are used to select, in each moment, the best set of sensors to use and the weight of each one.

Figure 4 shows the interface of the stress monitor. In this case, only the accelerometer and the touch screen were available. This allows compiling several types of information. Considering the accelerometer, it produces three visualizations of the data: (1) the raw data of the acceleration; (2) the acceleration without the data corresponding to touches and (3) the acceleration during touches. The first one allows seeing all the acceleration information (three axes and absolute value). The second one allows seeing this information without the acceleration due to touches. In fact, variations in acceleration are expected during touches, which will influence the determination of the levels of stress. In that sense, this visualization is more accurate. Finally, it is also possible to examine these variations of acceleration: higher variations in the acceleration during touches are related to increased levels of stress. The interface also shows the touch pattern for the last touch and the corresponding quadratic curve, that allows to compare it with already classified curves and classify the touch as stressed or calm. There is also a screen that shows the evolution of the score. This score is computed based on the performance of the user in doing a given task. In this case, the tasks consist of relatively simple mental calculations. Some stressors are considered to induce stress in the user, specifically the vibration of the device, annoying sounds and a decreasing time to perform the task. All

of this influences the performance of the user in completing the tasks, which is reflected on the score, and thus, on the level of stress. The quality of the estimation of the level of stress depends on the amount and quality of the information available at each time.

5 Conclusions

In this work we focused on how to estimate the stress state evolution from the users. This information can then be used by either the platform or even a mediator that is conducting process, to perceive how each issue or event is affecting each party. This, we believe, will increase the rate of success of conflict resolution procedures and bring them closer to the rich communicative environment that we have, when we communicate face-to-face. In future work we intend to incorporate additional components to the context and behavioural characterization, so that we can have a more accurate approach. Moreover, in a later phase, we intend to work with the School of Medical Sciences, to use, for instance, electroencephalograms. This will be useful not only for validating this approach but also to more accurately calibrate it.

References

1. Andrade, F., Novais, P., Carneiro, D., Zeleznikow, J., Neves, J.: Using BATNAs and WATNAs in Online Dispute Resolution. In: Nakakoji, K., Murakami, Y., McCready, E. (eds.) JSAI-isAI 2009. LNCS, vol. 6284, pp. 5–18. Springer, Heidelberg (2010)
2. Antunes, L., Pinto, H.S. (eds.): EPIA 2011. LNCS, vol. 7026. Springer, Heidelberg (2011)
3. Bol, S.H.: Ethan katsh and janet rifkin, online dispute resolution, resolving conflicts in cyberspace. Artif. Intell. Law 11(1), 69–75 (2003)
4. Bonarini, A., Mainardi, L., Matteucci, M., Tognetti, S., Colombo, R.: Stress recognition in a robotic rehabilitation task. In: Proceedings of the ACM/IEEE Human-Robot Interaction Conference (HRI 2008). Full-day Workshop on Robotic Helpers: User Interaction, Interfaces and Companions in Assistive and Therapy Robotics, pp. 41–48 (2008)
5. Carneiro, D., Gomes, M., Novais, P., Neves, J.: Automatic classification of personal conflict styles in conflict resolution. In: Winkels [26], pp. 43–52
6. Carneiro, D., Gomes, M., Novais, P., Neves, J.: Developing dynamic conflict resolution models based on the interpretation of personal conflict styles. In: Antunes, Pinto, (eds.) [2], pp. 44–58 (2011)
7. Carneiro, D., Novais, P., Neves, J.: Toward seamless environments for dispute prevention and resolution. In: Novais, (eds.) et al. [19], pp. 25–32
8. Charniak, E.: Bayesian networks without tears. AI Magazine 12(4), 50–63 (1991)
9. Corchado, E., Abraham, A., de Carvalho, A.C.P.L.F.: Hybrid intelligent algorithms and applications. Inf. Sci. 180(14), 2633–2634 (2010)
10. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
11. Cox, R.: Sport psychology: concepts and applications. W.C. Brown (1985)
12. Eriksen, H.R., Olff, M., Murison, R., Ursin, H.: The time dimension in stress responses: relevance for survival and health. Psychiatry Res. 85(1), 39–50 (1999)
13. Grundlehner, B., Brown, L., Penders, J., Gyselinckx, B.: The design and analysis of a real-time, continuous arousal monitor. In: International Workshop on Wearable and Implantable Body Sensor Networks, 156–161 (2009)

14. Gyselinckx, B., Vullers, R., Hoof, C.V., Ryckaert, J., Yazicioglu, R.F., Fiorini, P., Leonov, V.: Human++: Emerging technology for body area networks. In: 2006 IFIP International Conference on Very Large Scale Integration, pp. 175–180 (October 2006)
15. Healey, J., Picard, R.W.: Detecting stress during real-world driving tasks using physiological sensors. IEEE Transactions on Intelligent Transportation Systems 6(2), 156–166 (2005)
16. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Springer Publishing Company, Incorporated (2007)
17. Larson, D.A.: Technology mediated dispute resolution. In: Proceeding of the 2007 Conference on Legal Knowledge and Information Systems: JURIX 2007: The Twentieth Annual Conference, p. 5. IOS Press, Amsterdam (2007)
18. Liao, W., Zhang, W., Zhu, Z., Ji, Q., Gray, W.: Toward a decision-theoretic framework for affect recognition and user assistance. International Journal of Human-Computer Studies 64(9), 847–873 (2006)
19. Novais, P., Preuveneers, D., Corchado, J.M. (eds.): Ambient Intelligence - Software and Applications - 2nd International Symposium on Ambient Intelligence, ISAmI 2011, Salamanca, Spain, April 6-8. Advances in Intelligent and Soft Computing, vol. 92. Springer, Heidelberg (2011)
20. Picard, R.W.: Affective computing. MIT Press, Cambridge (1997)
21. Selye, H.: The stress of life, vol. 5. McGraw-Hill paperbacks, McGraw-Hill (1956)
22. Shi, Y., Nguyen, M.H., Blitz, P., French, B., Fisk, S., De la Torre, F., Smailagic, A., Siewiorek, D.P., al' Absi, M., Ertin, E., Kamarck, T., Kumar, S.: Personalized stress detection from physiological measurements. In: International Symposium on Quality of Life Technology (2010)
23. Thomas, K., Kilmann, R.: Conflict and conflict management (1974), http://www.kilmann.com/conflict.html
24. Tidd, T.S., Friedman, R.: Conflict style and coping with role conflict: An extension of the uncertainty model of work stress. International Journal of Conflict Management 13(3), 236–257 (2002)
25. Ursin, H., Eriksen, H.R.: The cognitive activation theory of stress. Psychoneuroendocrinology 29(5), 567–592 (2004)
26. Winkels, R. (ed.): Legal Knowledge and Information Systems - JURIX 2011: The Twenty-Third Annual Conference on Legal Knowledge and Information Systems, Vienna, AT, December 14-16. Frontiers in Artificial Intelligence and Applications, vol. 235. IOS Press (2011)
27. Wu, L., Oviatt, S.L., Cohen, P.R.: Multimodal integration - a statistical view. IEEE Transactions on Multimedia 1, 334–341 (1999)
28. Zhang, Y., Ji, Q.: Efficient sensor selection for active information fusion. Trans. Sys. Man Cyber. Part B 40, 719–728 (2010)

Multilayer-Perceptron Network Ensemble Modeling with Genetic Algorithms for the Capacity of Bolted Lap Joint

Julio Fernández-Ceniceros, Andrés Sanz-García, Fernando Antoñanzas-Torres, and F. Javier Martínez-de-Pisón-Ascacibar

EDMANS Research Group, University of La Rioja, Logroño, Spain
http://www.mineriadatos.com

Abstract. The assessment of failure force in bolted lap joints is a critical parameter in the design of steel structures. This kind of bolted joint shows a highly nonlinear behaviour so traditional analytical models are not very reliable. By contrast, other classical technique like finite element analysis provides a powerful tool to solve nonlinearities but usually with a high computational cost. In this article, we propose a data-driven approach based on multilayer-perceptron network ensemble model for failure force prediction, using a data set generated via finite element simulations of different bolted lap joints. Numeric ensemble methods combine multiple predictors to obtain a single output through average. Moreover, a procedure based on genetic algorithms is used to optimize the ensemble parameters. Results show greater generalization capacity than single prediction model. The resulting ensemble includes the advantages of finite element method whereas reduces the complexity and requires less computation.

Keywords: Genetic Algorithms, Multilayer-perceptron Network, Ensemble Model, Finite Element Method, Bolted Connection, Lap Joint.

1 Introduction

Currently, there are growing trends to use soft computing (SC) for many industrial and building applications [5]. One important reason behind the growing acceptance of SC is its guiding principle: exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness and low solution cost [18]. Other advantage is the existence of combinations of individual SC methods such as genetic algorithms (GA) for obtaining higher quality models, optimizing learning processes, solving competitively real-world problems, etc [6]. We apply a combination of SC techniques in order to avoid the high computational cost of a widely known hard computing (HC) technique, finite element method (FEM) [1].

In steel construction many studies have been carried out during the last decades to investigate the behaviour of bolted connections because they are

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 545–556, 2012.

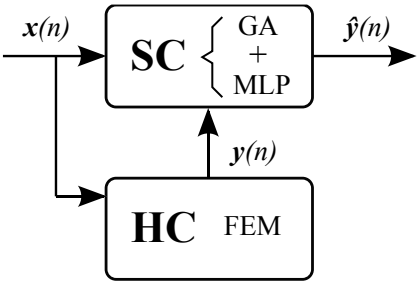


Fig. 1. Transposed hard computing and soft computing structure

critical elements in structures. These works are mainly divided into analytical models [17], experimental tests [4], and numerical implementations like the FEM [15]. The first ones are based on the classical theory of steel structures and they do not take into account nonlinear effects such as material plasticity, stress concentrations or contacts between plates and bolts [19]. On the other hand, experimental tests provide realistic results but they are limited to the test program features so it is not possible to get a generalized model. Finally, a solution to the analytical model is to use a HC technique like FEM which is capable to reproduce the connection behaviour accurately because it can solve complex elasticity and structural analysis problems [16].

However, in spite of the advances in computational field, attempts to achieve more realistic simulations have resulted as high computational times, making these models uncompetitive compared to analytical models. This greater computer time consumption is mainly caused by nonlinearities inherent in the physics of bolted connections [9]. We apply these techniques to get a very reduced computational cost than FEM in exchange for slightly lower accuracy. Moreover, we avoid the difficulties with the correct setting of some aspects of the finite element (FE) model strain conditions.

By following the fusion SC and HC categories of Ovaska et al. [18], the research presented here belongs to hybrid transposed of SC and HC approach with very low fusion grade (Fig. 1). The mathematical mapping is as follows:

$$\hat{\boldsymbol{y}}(k) = f_{SC}(\theta_{SC}; \boldsymbol{x}(k), \boldsymbol{y}(k)) = f_{SC}(\theta_{SC}; \boldsymbol{x}(k), f_{HC}(\theta_{HC}; \boldsymbol{x}(k))) \quad (1)$$

where k is the discrete sample index, $\boldsymbol{x}$ and $\boldsymbol{y}$ represent input and output vectors, $f_{SC}(\cdot;\cdot)$ and $f_{HC}(\cdot;\cdot)$ are arbitrary SC and HC algorithms, respectively, and finally θ_{SC} and θ_{HC} the corresponding system parameters such as the coefficients. In particular, this work implements $f_{HC}(\cdot;\cdot)$ as a numerical method to solve sets of partial differential equations for a nonlinear elastic solids and $f_{SC}(\cdot;\cdot)$ is the multilayer-preceptron networks ensemble (MLPE) model. The parameters θ_{SC} define the structures of the three-layer multilayer-preceptron (MLP) networks and the ensemble model: value of the learning rate, momentum ratio for neurons, number of neurons in the hidden layer, the activation function

for the hidden neurons, and even the criterion used to measure the proximity of the prediction to the target. It is interesting to note that the relatively high number of coefficients combination makes difficult to set the optimal ensemble model and many efforts to increase the accuracy generate overfitting after the training procedure. Therefore, an optimization strategy has to be applied to find the best combination. The present work utilizes the GA, an algorithm for a wide array of optimization problems within economics, engineering, ecology, social networks, and machine learning. This is an evolutionary strategy where individuals encode the parameters θ_{SC} that define the optimal prediction model.

In summary, this article considers the combination of an ensemble formed by MLP with GA. This model is applied for the prediction of the failure force in steel bolted connections. The remainder of this paper is scheduled as following. In section 2, the statement of the problem is presented. The modelling strategy combining GA and MLPE is drawn in section 3. Section 4 describes the case study of the GA-MLPE modelling to predict the failure force in steel bolted connections, where a briefly explanation about the FE model and the parametric study is shown. Finally, conclusions and future work are outlined in Section 5.

2 Problem Statement

In order to apply the methodology based on fusing GA and MLPE, the bolted lap joint problem has been chosen. Bolted connections are critical elements in steel structures and an easy procedure is necessary to improve the existing analytical models and to reduce the time consumed in the numerical models. The main objective in this article is to generate a substitute model of FE, based on ensemble techniques.

The configuration of bolted lap joint is according to the Fig. 2. The connection is composed of two inner plates, two outer plates, and two preloaded bolts. When the inner plates are subjected to a tension load, they are prevented from sliding by the bolts until the value of frictional force is exceeded. At this moment, a displacement of the inner plate is against the shank of the bolt; tensile force

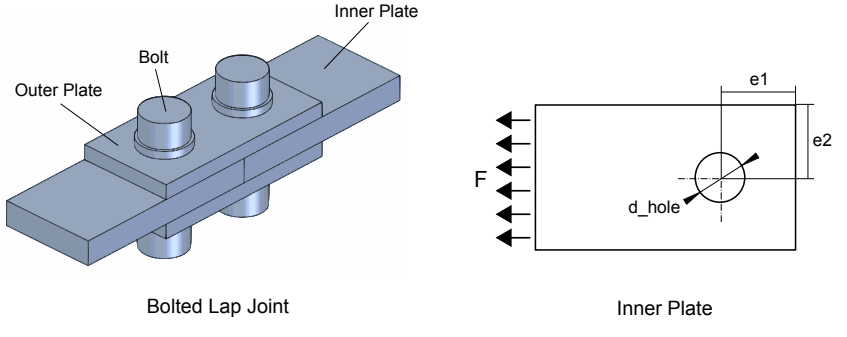


Fig. 2. Bolted Lap Joint. Geometrical parameters.

continues to build on the connection until there is a second displacement of the bolt against the outer plates; from there on, no further displacement is possible and the inner plate begins to be crushed against the bolt until failure occurs. The failures modes of bolted lap joint are following:

- *Bearing failure*: it is considered to constitute failure by the excessive deformation of material behind the bolt, regardless of whether the connection has reserve strength.
- *Net-section failure*: a critical fracture in connections with relatively narrow plate widths.
- *Tear-out failure*: it is associated with connections which have a relatively small end distance and pitch, while having a comparatively large edge distance and gauge to avoid net section failure mode.
- *Shear bolt failure*: a brittle fracture occurs in the bolts when the shear load exceeds its capacity.

The complexity of calculating a bolted lap joint lies in the nonlinearities associated with the structural behaviour of the connection as well as the different failure modes depending on the geometrical parameters. Consequently, we suggest there is a neccesity of creating prediction models from a set of FE simulations to provide the failure force.

3 Non-linear Modelling Using Soft Computing Techniques

3.1 Multilayer-Perceptron Neural Network

Artificial neural networks (ANN) are the best-known biologically information processing devices that are modelled as neuron with its inputs (dentrites) and outputs (axons). Once correctly trained, ANN can solve complex and challenged problems. The generalization capacity to tackle different cases with similar characteristics is its strong point [14].

MLP is an artificial feed-forward neural network model where information moves forward from the input nodes, through all hidden nodes, to the output nodes without loops [2]. MLP has theoretically the ability to solve any high dimensional classification problem and MLP is even able to model practically any continuous complex function. Mathematically, we can express the neural network output as follows:

$$\hat{\boldsymbol{y}}\left(k\right) = \alpha_0 + \sum_{i=1}^{n} \boldsymbol{\alpha}_i \cdot \phi_i \left(\sum_{j=1}^{l} \boldsymbol{w}_{ij} \cdot \boldsymbol{x}_j\left(k\right) + \boldsymbol{w}_{i0} \right) \tag{2}$$

where $\boldsymbol{x}_j\left(k\right) = \left[x_1\left(k\right), x_2\left(k\right), \ldots, x_l\left(k\right)\right]^T$ is the input vector at instant k with length l, $\boldsymbol{\alpha}_i$ and $\boldsymbol{w}_{ji}$ are the weighting coefficients of the network ($\boldsymbol{\alpha}_0$ and $\boldsymbol{w}_{i0}$ are the output and input layer bias respectively, ϕ_i is the nonlinear activate function, n is the number of hidden neurons, and $\hat{\boldsymbol{y}}\left(k\right)$ represents the estimation

of the output $\boldsymbol{y}$. In regression analysis with a single y value, there is a single neuron in the output layer.

The goal of the training process, where error back-propagation (BP) algorithm is usually applied, is to find the values of the weights $\boldsymbol{\alpha}_i$ and $\boldsymbol{w}_{ji}$ that will cause the MLP output to be equal to the target values as closely as possible. Finally, it is also widely known only one hidden layer is required to approximate any continuous function if the number of connection weights is enough [13].

3.2 Multilayer-Perceptron Network Ensemble Model

The multilayer-perceptron network ensemble (MLPE) model is the subject of this research. The model has been developed by Hansen and Salamon [12] and used in several applications [11]. The method fits perfectly to achieve superior predictions than a single neural network. The main reason of why an ensemble model should perform better is to consider the performance of a single expert versus that from a group of multiple experts [20]. In addition, the model solves the over-fitting and under-fitting in the nonlinear modelling when only a limited amount of training data is available. Combining networks by ensemble modelling do not have such a sound theoretical basis but we get good generalization ability avoiding fitting problems, and it is an alternative to the classical approaches such as jittering, early stopping, weight decay, or bayesian learning.

Table 1. Bagging as an ISLE-based algorithm

$F_0(x) = 0$
For $m = 1$ to M {
  $p_m = \arg\min_{P} \sum_{i \epsilon S_m(\eta)} L(y_i,\ F_{m-1}(x_i) + T(x_i;\ p))$
  $T_m(x) = T(x;\ p_m)$
  $F_m(x) = F_{m-1}(x) + \upsilon \cdot T_m(x)$
}
write $\{T_m(x)\}_1^M$

In this work, we use one of the earliest ensemble models. Bagging, short for Bootstrap Aggregation [3], generate firstly the individuals by bootstrap replicates in order to create the training data for each individual model in the ensemble. Each bootstrap replicate is obtained by sampling randomly from the training dataset with replacement. Finally, if we are using MLP networks to build the ensemble, we firstly determine all single networks and then, their outputs are combined by average or another weighted combination (regression). The reason the MLPE gives significant gains in accuracy from a single MLP is because bagging increases the differences between single models reducing the generalization error (regression).

Breiman´s bagging was formalized by Friedman and Popescu (2003) [10] as an algorithm with the Importance Sampling Learning Ensembles (ISLE) approach

as ahows Table 1. The variable υ is equal so there is no memory, $T_m(x)$ are the base learners (MLP model), $F_{m-1}(x_i)$ represents the ensemble of base learners up to step $m-1$, N is the size of the samples, $\eta = \frac{N}{2}$ means that each model is trained using half-size samples without replacement, and the loss function is $L(y, \hat{y}) = (y - \hat{y})^2$.

3.3 MLPE Model Optimization Using GA

The determination of the MLPE structure and its setting is considered an open issue in machine learning modelling. Expert knowledge and trial-error are the common practices to obtain a good model, but in most of the cases these procedures need a tedious repetitive modelling phase. Additionally, this approach does not guarantee that the model performance is the optimal. A second method is based on testing each possibility sequentially in order to determine if it is the solution. This is widely known as direct search (DS), exhaustive search or even brute force method. In terms of computational cost, DS increases exponentially, and becomes impractical for most of the optimization cases.

In this article, we choose an advanced computational SC technique such as GA to solve this problem. There are three possible ways for combining GA and MLPE: GA-based tuning of connecting weights and bias values, GA-based tuning of topologies, and GA-based preprocessing of data with interpretation of the outputs. For building the prediction model for the capacity of bolted lap joint, we generate a semi-fixed architecture where only the number of neurons in the hidden layer changes. Based on simple rules and few parameters dictating the GA operations, the GA explores solution space and efficiently finds the best one. In particular, the optimal MLPE model will be obtained through the minimization of the fitness function based on the root mean-squared error (RMSE) of the predictions:

$$J = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y_{FEM}(k) - \hat{y}(k)\right)^2} \tag{3}$$

The working principle of the chosen GA-based approach is described as follows:

1. Selection of the parameters to optimize.
2. Generation of an initial population of solutions in a random way over its search space. The population size depends on the complexity of the problem.
3. The objective function (fitness values) and constraints of each solution in the population are evaluated:

$$J_{min} = \operatorname{argmin}\sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y_{FEM}(k) - \hat{y}\left(\boldsymbol{x}(k), \{\alpha_i, w_{ij}; i, j \leq n, l\}\right)\right)^2} \tag{4}$$

4. The parents of the next population are selected by an operator named reproduction where the ones from the population are chosen depending on their

fitness values. Other genetic operators like crossover or mutation are applied in order to create the rest of solutions for the next population.

5. After we create another whole generation and if the optimum value has not been reached yet, steps $2-4$ are repeated until the desired accuracy in the solution or a maximum number of generations is achieved.

4 Case Study: Bolted Lap Joint

In this section, the MLPE with GA combination is applied to predict the failure force in bolted lap joints through empirical experiments.

4.1 Development of the FE Model

An advanced 3D FE model is developed in order to simulate the bolted lap joint behaviour. This numerical model represents the HC technique used and has been created using ABAQUS v.6.11, a general purpose FE software.

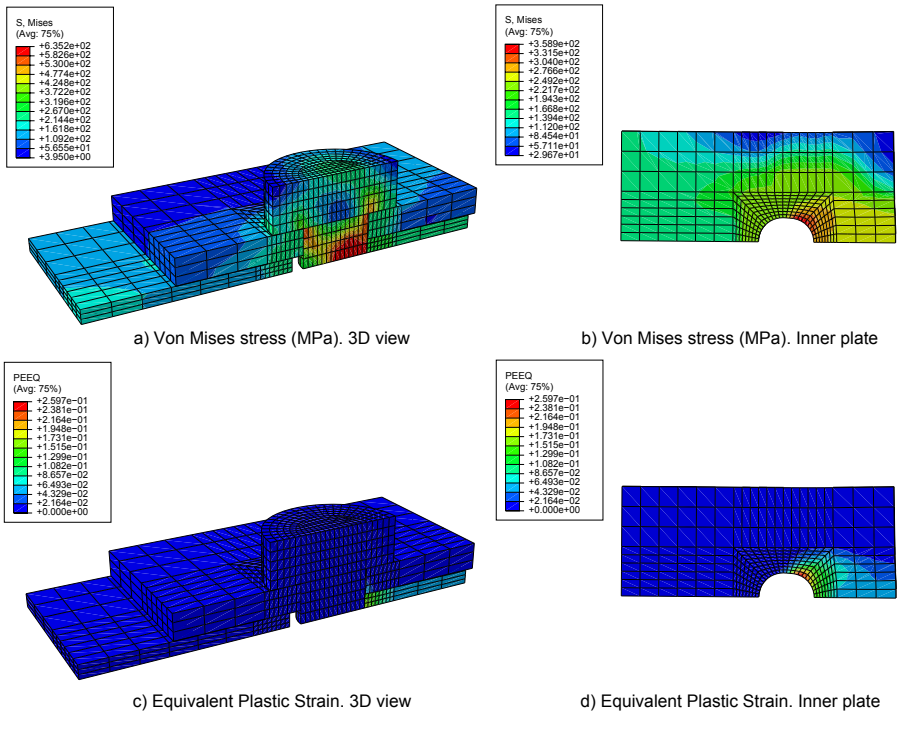


Fig. 3. Von Mises Stress (MPa) and plastic equivalent strain distributions (%)

Table 2. Material properties of the elements (plates and bolts)

Element	Steel Grade	Minimum Yield Strength (MPa)	Tensile Strength (MPa)	Minimum Elongation (%)
Plates	S235 JR	235	360	26
Bolts	8.8	640	800	12

Step 1. Material Behaviour of the FE model. The model takes into account the nonlinear material behaviour and the steel in the plates and the bolts is modelled as a bilinear approximation (elastic-plastic with strain hardening). Stress and strain levels are shown in Table 2, according to standard EN 10025-2:2004 [7] for S235 JR steel and to Eurocode 3 [8] for grade 8.8 preloaded bolts.

Step 2. Boundary Conditions of the FE model. The boundary conditions for the model are established taking into consideration its three-way symmetry with regard to the X, Y and Z axis planes. Therefore, only one eighth of the model is simulated and important savings in computational time are made. The simulation of contacts between plates and bolts is carried out by the General Contact algorithm provided by ABAQUS. This algorithm automatically establishes the interactions between the contact surfaces being suitable to simulate highly non-linear processes. The normal behaviour of the bolt shank with regard to the bolt hole is modelled by the hard contact property. When the surfaces are in contact, any pressure between them can be transmitted, with the surfaces separating if the pressure drops to zero. Tangential behaviour is modelled by Coulomb's basic friction model, in which two surfaces in contact can withstand shearing stresses of a certain magnitude at their interface before one starts to slide over the other. Finally, the preload of the bolt is simulated by a cooling in the bolt shank, so that the decrease in shank length is equivalent to that caused by preload.

Step 3. Finite Element Analysis of the Joint. Explicit analysis method is used because it is suitable for nonlinearities, such as contacts or geometrical discontinuities and it shows a good convergence. Figure 3 shows the results about Von Mises stress and plastic equivalent strain in a bolted lap joint FE model. The failure mode of the joint to the configuration shown in the Fig. 2, is produced by bearing of the inner plate against the bolt shank. In this case, the maximum stress (360 MPa) and the maximum strain (26 %) of the S235 JR steel are achieved in the area around the bolt hole. Finally, the value of the maximum strength supported by the joint is 22918 N.

4.2 Parametric Study

A parametric study is carried out in order to study the influence of several geometrical and mechanical parameters affecting the overall behaviour of the

Table 3. Statistics of input parameters in the dataset

Attributes	Description [units]	Range	Mean	Sd.
d_ bolt	Bolt diameter [mm]	$6-12$	8.656	1.700
e1	End distance [mm] (Fig. 2)	$8-25$	17.59	4.339
e2	Edge distance [mm] (Fig. 2)	$8-35$	22.61	7.136
t_ inner	Inner plate thickness [mm]	$2-12$	7.003	2.849
t_ outer	Outer plate thickness [mm]	$1-6$	3.487	1.436
friction	Friction coefficient between elements	$0.1-0.5$	0.300	0.114
preload	Clamping force in the bolts [kN]	$1-30$	17.88	9.686

connection. This study is generated combining seven input parameters (Table 3) by random latin hypercube. Table 3 summarizes the main properties of the general dataset. It shows, for each attribute or input parameter, the description, the range of its values, the mean, and the standard deviation. Finally, we check the existence of data-fractures or even noise.

For each combination of the input parameters a FE simulation is carried out in order to obtain the failure force of the joint. Owing to the heavy computation involved in each FE simulation, the dataset created to develop the ensemble model is only composed of 1155 instances. The correlation coefficients between the attributes were less than 0.62 and the scatterplot showed plenty of cases that did not fall on the line. Only if we observe p_max and f_max we state that there is a strong relation between these two attributes.

4.3 Experiments and Results

In this work, the data used for modelling are from the previous FE simulations. We carried out, first, a data normalization into a range of [0, 1] instead of a statistical standarization. In our opinion, the normalization will work well because the data are positive or zero. Then, the normalized dataset was divided in three subsets where the ratios were adjusted to cover the main characteristics of the overall data. That is, the dataset was split into training (60 %), validation (25 %), and testing (15 %) dataset.

The generation of the models was developed on a Linux operating system SUSE 10.3 OS running in a Quad-Core Opteron server. The statistical analysis package R (`http://www.r-project.org`) and the Weka suite (`http://www.cs.waikato.ac.nz/ml/weka`), a collection of machine learning algorithms for data mining tasks, are the tools used to processing data and training the models. In order to prove the proposed method is the best one, we compared it with other bagging techniques applied to: linear regression, M5P/model trees, and support vector machines (SVM). Some initial trials have been conducted for adjusting the main setting parameters of these algorithms, but in these three cases the best configuration is found by using direct search (DS).

Table 4 summarizes the errors predicting failure force in bolted lap joints corresponding to the mean and the standard deviation (sd) of the best ten (10)

Table 4. Computational cost, and training, validation and checking errors

Algorithm	RMSE_{tr}^{mean}	RMSE_{tr}^{sd}	RMSE_{V}^{mean}	RMSE_{V}^{sd}	RMSE_{T}	$t(hours)$
Bagg-LR	0.051	0.003	0.053	0.004	0.102	38
Bagg-M5P	0.047	0.001	0.049	0.002	0.059	42
Bagg-SVM	0.037	0.002	0.041	0.003	0.052	96
GA-MLPE	0.021	0.005	0.026	0.004	0.027	47

models. Hold-out method is used as the validation method for determining how accurately a learning algorithm will be able to predict data. For each algorithm, we obtain training, validation, and testing root mean squared errors (RMSE), and total computational time of the modelling phase. The training data are directly used to the modelling, and the validation data estimate how accurately a predictive model will perform in practice. In case of GA-MLPE, we prevented model overfitting using the validation data to stop the training process. Finally, testing data, that are hidden during the whole modelling procedure, are used to check the generalization capacity of the model. The GA optimization procedure was carried out with a maximum number of generations of 100 and a population size of 30 individuals represented in a vector of real values. The genetic operators applied to create the next population are binary tournament for selecting the best of two individuals chosen randomly: a uniform crossover that generates a random value between 0 and 1 for each gene, and a non uniform mutation operator with a probability of 0.15.

In Table 4, it can be seen, as a consequence of low $RMSE$, that results show high correlation between FEM simulation and four ensembled models. The best algorithm will be the one with the less MSE_T, i.e. the best generalization ability. GA-MLPE with a $RMSE_t^{mean}$ value of 2.1 %, $RMSE_V^{mean}$ value of 2.6 %, and finally $RMSE_T$ value of 2.7 % represents the best model of the four (4) obtained. We also observe in Table 4 that the computational complexity is excessively high for a DS instead of a evolutionary strategy. This comparison is important since these models are also capable of eficiently solving joints in steel construction software. If the MLPE model would not possess an important advantage over the others then it has no sense the developing of the model. Finally, if the models of the Table 4 are compared with single models like MLP or SVM, we will observe that the bagging techniques improve the accuracy of the results in all of them because reducing the bias of the prediction but not the variance.

In conclusion, DS is working accurately but with a excesive computational cost unlike optimization with GA. DS is highly susceptible to finding the global minimum if the search limits are correctly chosen. On the other hand, it has no ability to stop the process after it finds the global minimum.

5 Conclusions and Future Work

In this work, a combination of nonlinear FE model and MLPE has been applied to study the performance of failure force prediction in steel bolted connections. The regression model has gone through an evolutionary optimization procedure,

using a dataset of 1155 instances which cover a practical range of steel plates and bolts. The difference between the predicted values from the MLPE and those obtained from the FE analysis are within the range of 2 % and 3 %. In our opinion, the main advantage of the MLPE model compared to the FE model is the integration in structure software due to its its high speed in computation. The evolutionary strategy used in this work would be extended to predict the failure force in bolted connections made of other materials like low-carbon steel, carbon fiber, plywood, etc. This will require further data generation for new materials using FEM.

Acknowledgments. The authors thank to the *Autonomous Government of La Rioja* for its support through the *Third Rioja Plan of Research and Development* for the project FOMENTA 2010/13 and to the *University of La Rioja* through FPI fellowships, to the Santander Bank for the project API11/13.

References

1. Annicchiarico, W., Cerrolaza, M.: Structural shape optimization 3d nite-element models based on genetic algorithms and geometric modeling. Finite Elements in Analysis and Design 37, 403–415 (2001)
2. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
3. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
4. Bursi, O.S., Jaspart, J.P.: Benchmarks for finite element modelling of bolted steel connections. Journal of Constructional Steel Research 43(1-3), 17–42 (1997)
5. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
6. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
7. European Committee for Standardization: EN 10025-2: 2004. Non-alloy structural steels: grades, mechanical properties and nearest equivalent grades
8. European Committee for Standardization: EN 1993-1-8 Eurocode 3. Design of steel structures part 1-8. Design of joints
9. Fernández, J., Pernía, A., de Pisón, F.M., Lostado, R.: Prediction models for calculating bolted connections using data mining techniques and the finite element method. Engineering Structures 32(10), 3018–3027 (2010)
10. Friedman, J.H., Popescu, B.E.: Importance sampled learning ensembles. Tech. rep., Stanford University, Department of Statistics (2003)
11. Garcia-Pedrajas, N., Hervas-Martinez, C., Ortiz-Boyer, D.: Cooperative coevolution of artificial neural network ensembles for pattern classification 9(3), 271–302 (2005)
12. Hansen, L.K., Salamon, P.: Neural network ensembles 12(10), 993–1001 (1990)
13. Hornik, K., Stinchcombe, M.B., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359–366 (1989)
14. Jones, M.T.: Artificial Intelligence: A Systems Approach. Infinity Science Press, LLC (2008)

15. Ju, S.-H., Fan, C.-Y., Wu, G.H.: Three-dimensional finite elements of steel bolted connections. Engineering Structures 26(3), 403–413 (2004)
16. Kim, T.S., Kuwamura, H., Cho, T.J.: A parametric study on ultimate strength of single shear bolted connections with curling. Thin-Walled Structures 46(1), 38–53 (2008)
17. Loureiro, A., Gutiérrez, R., Reinosa, J., Moreno, A.: Axial stiffness prediction of non-preloaded t-stubs: An analytical frame approach. Journal of Constructional Steel Research 66(12), 1516–1522 (2010)
18. Ovaska, S.J., Kamiya, A., Chen, Y.: Fusion of soft computing and hard computing: computational structures and characteristic features 36(3), 439–448 (2006)
19. Salih, E.L., Gardner, L., Nethercot, D.A.: Numerical investigation of net section failure in stainless steel bolted connections. Journal of Constructional Steel Research 66(12), 1455–1466 (2010)
20. Yang, Y.-Y., Mahfouf, M., Pnoutsos, G.: Development of a parsimonious ga-nn ensemble model with a case study for charpy impact energy prediction. Advances in Engineering Software 42, 435–443 (2011)

A Hybrid Classical Approach to a Fixed-Charged Transportation Problem

Camelia-M. Pintea, Corina Pop Sitar, Mara Hajdu-Macelaru, and Pop Petrica

North University, 430083 Baia-Mare, Romania
{cmpintea,sitarcorina,macelarumara,pop_petrica}@yahoo.com

Abstract. Some of the most complex problems, nowadays, are transportation problems. A capacitated fixed-charge transportation problem is the problem we are trying to solve using hybrid classical approaches. The problem is considered with fixed capacities for each distribution centers and customers have particular demands. The model, as an economical model, minimizes the total cost as some distribution centers are selected in order to supply demands of all the customers.

In order to find feasible solution for the mentioned problem, we are using some variants of *Nearest Neighbor* search algorithm. The problem as a whole is a two stages supply chain network: first we have to choose the distribution centers and next the customers based on their demand. The new approach is that we are starting from given customers' demands and select the best distribution centers. Some hybrid variants of *Nearest Neighbor* based on different probabilities are investigated and tested on large sizes data. Based on the numerical results we found a suitable hybrid version for the specified transportation problem.

Keywords: Hybrid algorithms, Nearest Neighbor, Transportation Problem.

1 Introduction

Transportation problems are combinatorial optimization problems. Researchers use different heuristics [4,16,18] approximation [12] or hybrid techniques [17] in order to solve this type of complex problems. Modeling transportation problems is difficult. The algorithms should take into account the large number of constraints, high dimensions, several uncertainties, a large number of parameters and many local optimal solutions.

A transportation problem is basically a network problem [1]. As we know from [2], in a transportation problem the transportation cost is directly proportional to the number of units transported. Some distribution and transportation problems can be described as fixed cost transportation problems [3, 4].

A particular fixed-charge transportation problem is described in this paper. The problem involves two supply chains. The first supply chain is starting from a manufacturer who delivers items to certain distribution centers. The second chain is

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 557–566, 2012.

from the distribution centers to a list of given customers, with their own demands. The objective function is to minimize the whole transportation cost, from manufacturer to customers, based on several intermediary costs and other parameters. The objective function takes into account some fixed costs and some transportation costs. A mathematical model of the problem is detailed and the parameters and the constraints are clearly illustrated.

The use of conventional, linear algorithms for solving this mathematical programming model of the fixed-charged transportation problem is limited due to the complexity of the problem and the large number of variables and constraints.
Hybrid models based on *Nearest Neighbor* are used further due to the complexity of the objective function.

The paper is organized as follows. In the second section is described the fixed-charged transportation problem and the mathematical model. Section 3 describes the proposed models and his hybrid variants and Section 4 provides experimental comparative results. In the last section some conclusions and further research are provided.

2 The Fixed-Charge Transportation Problem

The present paper consider two stages of a supply chain network: distribution centers, denoted DCs, and customers. In Figure 1 is a model of a supply chain network.

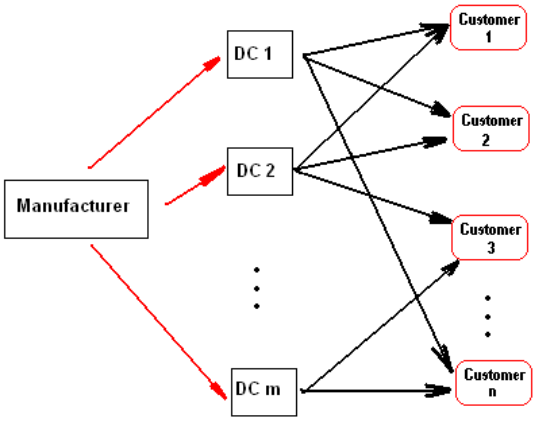


Fig. 1. A two stage supply chain network

2.1 Description of the Fixed Transportation Problem

As we can see from Figure 1, there are potential distribution centers (DCs) which are candidate for the manufacturer. Also, there are several customers whose particular demands should be satisfied by distribution centers. In [14] is supplied also an elementary example of the problem.

The manufacturer is assumed with no capacity limitation in production. As in [14] we assume that each potential distribution center has distinct and different capacity in order to support the customers. It is considered a transportation cost from the manufacturer to a distribution center and an opening cost for each potential distribution center.

The problem should find the total cost selecting potential places as distribution centers which supply demands of all the customers.

The total cost considers the following: fixed costs and transportation costs.

Fixed costs are considered: the opening cost for potential distribution centers and fixed cost for transportation from distribution centers to customers.

There are also two types of transportation cost: from manufacturer to each distribution center and transportation cost from all possible distribution centers to the customers.

2.2 Mathematical Model of the Fixed Transportation Problem

In the mathematical model are considered m potential distribution centers and a number of n customers, each one with a particular demand.

First, let see who the fixed costs are.

- The opening cost for a potential distribution center i, is denoted f_i.
- The fixed cost for transportation from distribution center i, to the customer j, is denoted f_{ij}.

The transportation cost per unit is following.

- The transportation cost from manufacturer to a potential distribution center i, is denoted c_i.
- Each potential distribution center i, can transport demands to a customer j, one of the n customers, at a transportation cost per unit c_{ij}.

The problem considers two stages of a supply chain network. That is why two main objectives have to be considered in order to minimize the total cost.

- First objective is to find which candidate places are to be opened as distribution centers.
- Second objective is to find which customers are served from opened distributers.

Other notations of the mathematical model of the problem are following.

- The known quantity to be transported on the route *(i, j)* from distribution center i to customer j is x_{ij}
- The capacity of a distribution center i is denoted a_i.
- The number of units demanded at customer j is b_j

The mathematical model of the objective problem is to minimize the following function *Z:*

$$Z = Z_{tc} + Z_{fc} \tag{1}$$

where Z_{tc} is the total cost of transportation and Z_{fc} is the total of the fixed costs.

Z_{tc} , the total cost of transportation, includes the transportation cost from manufacturer to distribution centers and from distribution centers to customers.

$$Z_{tc} = \sum_{m}^{i=1} c_i x_i + \sum_{m}^{i=1} \sum_{n}^{j=1} c_{ij} x_{ij} \tag{2}$$

where

$$x_i = \sum_{n}^{j=1} x_{ij}, i = 1...m, \tag{3}$$

$$b_j = \sum_{m}^{i=1} x_{ij}, j = 1...n \tag{4}$$

and there are two constraints:

$$x_{ij} \geq 0, \forall i = 1..m, \forall j = 1..n \tag{5}$$

$$x_i \leq a_i, i = 1...m \tag{6}$$

Z_{fc}, the total of the fixed costs, includes the opening costs of distribution centers and the fixed costs from distribution centers to customers.

$$Z_{fc} = \sum_{m}^{i=1} f_i y_i + \sum_{m}^{i=1} \sum_{n}^{j=1} f_{ij} y_{ij} \tag{7}$$

where

$$y_i = \begin{cases} 1, \sum_{n}^{j=1} x_{ij} \geq 0 \\ 0, \sum_{n}^{j=1} x_{ij} = 0 \end{cases} \forall i = 1...m \tag{8}$$

$$y_{ij} = \begin{cases} 1, x_{ij} \geq 0 \\ 0, x_{ij} = 0 \end{cases} \forall i = 1...m, \forall j = 1...n \tag{9}$$

3 Proposed Hybrid Models Based on Nearest Neighbor

First, we introduce the *nearest neighbor searching* technique, as in [12].

Let S be a given set of *n data points* in a metric space *X*. The problem is to process these points so that given any *query point* $q \in X$, the data point nearest to *q* can be found quickly.

The *Nearest neighbor searching problem* was also called the *closest-point problem* and the *post office problem* [13]. Nearest neighbor searching was used in a large number of applications as knowledge discovery and data mining [5], machine learning [7], multimedia databases [9], classification [6], data compression [8], statistics [11] and document retrieval [10].

The easiest technique in order to solve the nearest neighbor searching problem is to compute the distance from the query point to all points from the given search domain, keeping track of the "best so far" point.

In order to find a good solution for the fixed-charge transportation problem, we use a *nearest neighbor search* algorithm.

The common steps of *Nearest Neighbor* algorithm, used also for the traveling salesman problem are following [15].

1. Stand on an arbitrary vertex as current vertex
2. Find out the lightest edge connecting current vertex and an unvisited vertex V
3. Set current vertex to V
4. Mark V as visited
5. If all the vertices in domain are visited then terminate, else go to step 2

For the fixed-charged transportation problem, with two stages of a supply chain network, we have two possibilities to apply *Nearest Neighbor* algorithm. First, when we are choosing the potential distribution center and second when is choosing the best edge from the distribution center to customers.

There are introduced to variants of algorithms. At first, nearest neighbor is used in both supply chains and the second variant is a hybrid model.

3.1 The *Nearest Neighbor* Transportation Model

a) The first supply chain is between the manufacturer and distribution centers.

The Nearest Neighbor (NN) procedure for choosing the distribution center follows.

1. The manufacturer take the demand
2. All distribution centers are marked unvisited
3. The manufacturer choose an unvisited distribution center with the *smallest capacity*
4. If the demand is greater than this capacity, mark visited the current distribution center; modify demanding subtracting the DCs capacity and go to step 3
5. If the demand is smaller or equal with DCs capacity, mark visited the current distribution center and terminate procedure

When procedure ends we will know exactly the list (L) of distribution centers selected for the next supply chain.

There are used two variants for the first supply chain.

- DX-randomly chosen the potential distribution centers
- DY-with the best probability based on the total request and distribution centers capacities. We tested several probabilistic variants as:

$$p_i = \frac{a_i}{request} \tag{10}$$

$$p_i = \frac{a_i}{xcont_i \cdot request} \tag{11}$$

$$p_i = \frac{xcont_i}{a_i} \tag{12}$$

where $xcont_i$ is the number of nonzero quantities to be transported on the route *(i, j)* from distribution center *i* to customer *j*, $x_{ij}>0$.

b) The second supply chain starts from the customers' demands and try to reach the better edges to distribution centers.

In the *Nearest Neighbor* procedure used for choosing the distribution center starting from a given customer is as follows.

1. The customer *i* give the demand
2. All distribution centers, from the list *L*, are marked unvisited
3. The customer choose an unvisited distribution center with the smallest x_{ij}, the quantity to be transported on the route *(i, j)* from distribution center *i* to customer *j from list L*
4. If the demand is less than the capacity of chosen distribution center, mark visited the current distribution center; modify distribution center capacity subtracting the demand and go to step 3
5. If the demand is greater than this capacity, mark visited the current distribution center and go to step 3
6. If the demand is less or equal to this capacity, mark visited the current distribution center and terminate procedure

3.2 Hybrid Transportation Models

For hybrid transportation models are considered several variants.

In order to choose a *distribution center,* in the hybrid *H-NN* variants, the first supply chain is using *DX* or *DY,* as in Section 3.1.

For the second supply chain is considered *Nearest Neighbor* technique for choosing the best way from de distribution centers to customers.

The computational experiments performed further were guided the objective to determine the most promising hybrid based *Nearest Neighbor* model.

4 Computational Experiments

In order to achieve the fixed-charged problem' objectives, was implemented a java program. The computer used for testing data was an AMD 2600, 1.15 GHz and 1024MB RAM. The program was compiled under Linux. The randomly generated instances, as in [14], are illustrated in Table 1.

Table 1. Generating data table with problems characteristics

Problem Size	Total supply	Total demand	Variable costs	Fixed costs	Opening costs
10 x 10	30.000	10.000	[3,8]	[50,200]	[150,600]
10 x 30	45,000	15,000	[3,8]	[400,1600]	[1200,4800]
30 x 100	90,000	30,000	[3,8]	[400,1600]	[1200,4800]

For each instance, all algorithms were executed for a maximum amount of time. The time is in milliseconds and is relevant just in comparison with the models run on the already mentioned computer. The stopping criterions are following: for dimension 10 x 10 is 15 ms, 150 ms for size 10 x 30 and 500 ms for 30 x 100.

The comparative results are shown in Figure 2 for data sets 10 x10, Figure 3 for 10 x 30 data sets and for Figure 4 for 30 x 100 data sets. There are considered for each problem three different data sets (*Series 1-3*).

The notations are:

- *NN - Nearest Neighbor* for both supply chains choosing depots and the customer starting with the one with the minimum capacity.
- *HNN - Nearest Neighbor* for both supply chains choosing depots and the following variants:
    - *DX* and *DY* - random choosing the distribution center and choosing with a best probability (10) / (11) / (12) (Section 3.1) the distribution center in the first supply chain

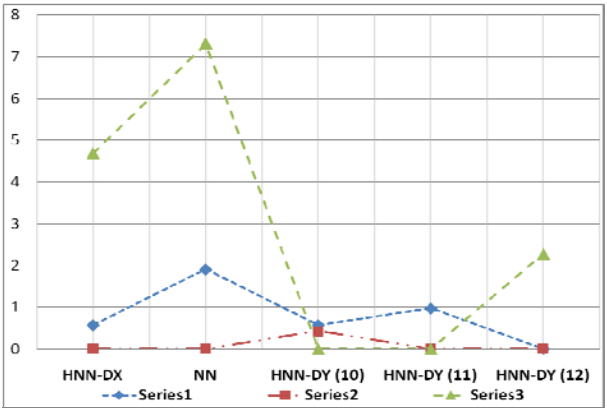


Fig. 2. Gap values for hybrid Nearest Neighbor variants on 10 x 10 data sets

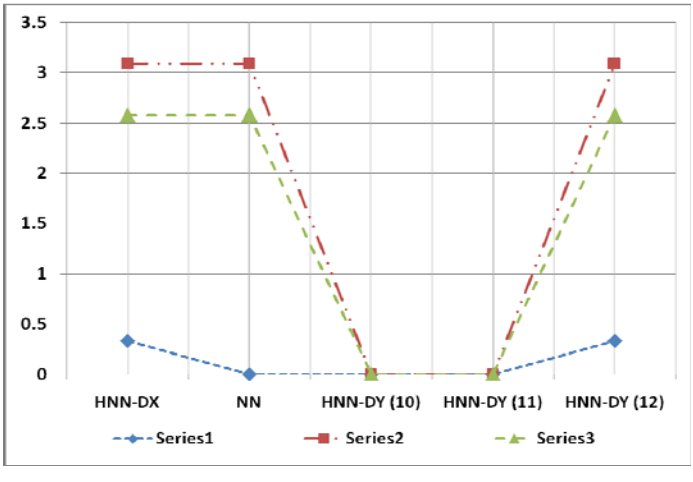


Fig. 3. Gap values for hybrid Nearest Neighbor variants on 10 x 30 data sets

As we see from Figures 2-4, based on gap's values, *HNN-DY(11)* and *HNN-DY (12)* are suitable for this type of transportation problem. The results of an unpaired t-test for *HNN-DY(11)* and *HNN-DY(12)* is *t=0.214* and *standard deviation=0.300E+05*. The probability of the result, assuming the null hypothesis is 0.83. The difference is considered to be not statistically significant.

A conclusion is that for large data is beneficial to use the hybrid variant *HHN-DY(12)* and for smaller data *HHN-DY(11)*. Another significant conclusion is the real importance of the factor $xcont_i$, the number of nonzero quantities transported on the route *(i, j)* from distribution center *i* to customer *j*.

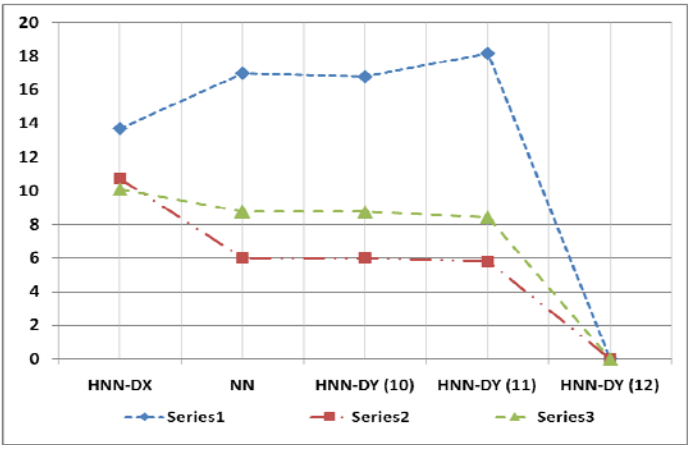


Fig. 4. Gap values for hybrid Nearest Neighbor variants on 30 x 100 data sets

5 Conclusions and Future Work

The paper describes some hybrid techniques for solving the fixed charged transportation problem. The problem is a two chain supply network. Classical *Nearest Neighbor* algorithm is used basically to find the best distribution centers when sending items from the manufacturer and from distribution centers to all customers. The demand for all customers should be satisfied. Different probabilities are used and tested on different large scale data in order to find the most suitable for choosing the best distribution center.

More investigations have to be made in order to use larger data, hybridizations and other local search procedures as well as a parallel architecture.

Acknowledgment. This research is supported by Grant PN II TE 113/2011, New hybrid metaheuristics for solving complex network design problems, funded by CNCS Romania.

References

1. Hitchcock, F.L.: The distribution of a product from several sources to numerous localities. J. of Mathematical Physic. 20, 224–230 (1941)
2. Diaby, M.: Successive linear approximation procedure for generalized fixed charge transportation problems. J. of Oper. Research Society 42, 991–1001 (1991)
3. Adlakha, V., Kowalski, K.: On the fixed-charge transportation problem. OMEGA: The International Journal of Management Science 27, 381–388 (1999)
4. Sun, M., Aronson, J.E., Mckeown, P.G., Drinka, D.: A tabu search heuristic procedure for the fixe charge transportation problem. European Journal of Operational Research 106, 441–456 (1998)

5. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advancesin Knowledge Discovery and Data Mining. AAAI Press/MIT Press (1996)
6. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley& Sons, NY (1973)
7. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning 10, 57–78 (1993)
8. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic, Boston (1991)
9. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. IEEE Computer 28, 23–32 (1995)
10. Deerwester, S., Dumals, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latend semantic analysis. J. Amer. Soc. Inform. Sci. 41, 391–407 (1990)
11. Devroye, L., Wagner, T.J.: Nearest neighbor methods in discrimination. In: Krishnaiah, P.R., Kanal, L.N. (eds.) Handbook of Statistics, vol. 2. North-Holland (1982)
12. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R.: An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. Journal of the ACM 45(6), 891–923
13. Knuth, D.: The Art of Computer Programming, vol. 3. Addison-Wesley, Boston (1997)
14. Molla-Alizadeh-Zavardehi, S., Hajiaghaei-Keshteli, M., Tavakkoli-Moghaddam, R.: Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prüfer number representation. Expert Systems with Applications 38, 10462–10474 (2011)
15. Gutin, G., Yeo, A., Zverovich, A.: Traveling Salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. Discrete Applied Mathematics 117, 81–86 (2002)
16. Chira, C., Dumitrescu, D., Pintea, C.-M.: Learning sensitive stigmergic agents for solving complex problems. Computing and Informatics 29(3), 337–356 (2010)
17. Pintea, C.-M., Crişan, G.-C., Chira, C.: Hybrid Ant Models with a Transition Policy for Solving a Complex Problem. Logic Journal of the IGPL (2011), doi:10.1093/jigpal/jzr004
18. Reyes, L.C., Zezzatti, C.A.O.O., Santillán, C.G., Hernández, P.H., Fuerte, M.V.: A Cultural Algorithm for the Urban Public Transportation. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 135–142. Springer, Heidelberg (2010)

Computing Optimal Solutions of a Linear Programming Problem with Interval Type-2 Fuzzy Constraints

Juan Carlos Figueroa-García[1,⋆] and Germán Hernandez[2,⋆⋆]

[1] Universidad Nacional de Colombia, Sede Bogotá
Universidad Distrital Francisco José de Caldas, Bogotá - Colombia
jcfigueroag@udistrital.edu.co
[2] Universidad Nacional de Colombia, Sede Bogotá
gjhernandezp@gmail.com

Abstract. This paper presents the computation of the set of optimal solutions of a Fuzzy Linear Programming model with constraints that involve uncertainty, by means of Interval Type-2 Fuzzy sets.

By applying convex optimization algorithms to a linear programming model with Interval Type-2 fuzzy constraints, an Interval Type-2 fuzzy set of optimal solutions derived from the uncertain constraints of the problem, is obtained. This set of optimal solutions is defined through four boundaries which determine its behavior. Finally, some theoretical considerations are made and explained through an application example.

Keywords: Interval Type-2 fuzzy sets, Fuzzy optimization, Soft constraints.

1 Introduction and Motivation

Optimization under uncertainty is a recent approach which can be defined in different ways: Interval-valued, Fuzzy, Possibilistic and Stochastic, among others. A new family of higher order uncertainty measures are the Type-2 Fuzzy Sets (T2FS) which deal with linguistic uncertainty and imprecision of fuzzy sets, so its use in optimization problems is an interesting topic to be addressed.

Fuzzy Linear Programming (FLP) is becoming an important tool for solving Linear Programming (LP) problems that involves uncertainty. Type-1 fuzzy sets problems can be solved using the well known method proposed by Zimmermann in [1] and [2], and Interval Type-2 fuzzy Sets (IT2FS) problems can be solved by one of the extensions proposed by Figueroa in [3], [4] and [5].

This paper is intended to define the set of optimal solutions of an Interval Type-2 Fuzzy Linear Programming model (IT2FLP) by using classical algorithms. This set is composed from a set of Interval Type-2 Fuzzy Constraints (IT2FC), which contains all possible optimums of the problem at different uncertainty degrees.

⋆ Juan Carlos Figueroa is Ph. D. Student in the Universidad Nacional de Colombia, Sede Bogotá and Assistant Professor in the Universidad Distrital Francisco José de Caldas, Bogotá - Colombia.

⋆⋆ Germán Hernandez is Associate Professor in the Engineering Dept. of the Universidad Nacional de Colombia, Sede Bogotá.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 567–576, 2012.

The paper is divided into five principal sections. In Section 1, a brief Introduction and Motivation is presented; Section 2 presents basic definitions of uncertain constraints; Section 3 describes the behavior of the optimal solutions of an LP problem with Fuzzy Right Hand Side (RHS) parameters; in Section 4, an application example is provided; and in Section 5 the concluding remarks of the work are given.

2 Interval Type-2 Fuzzy Constraints

The theory of Type-2 fuzzy sets has been developed by authors as Mendel (See [6], [7], [8], [9], [10] and [11]), and Melgarejo (See [12] and [13]) where its results are important tools for designing optimization models. In this way, the following definitions given by Figueroa are referred.

Definition 21 (IT2FS RHS - Figueroa in [3]). *Consider an RHS parameter of an FLP problem defined as an IT2FS $\tilde{b}$ over the closed interval $\tilde{b}_i \in [\underline{b}_i, \overline{b}_i]$, $\{\underline{b}_i, \overline{b}_i\} \in \mathbb{R}$, $i \in \mathbb{N}_n$. The membership function which represents the fuzzy space[1] of $\tilde{b}_i$ is:*

$$\tilde{b}_i = \int_{b_i \in \mathbb{R}} \left[\int_{u \in J_{b_i}} 1/u \right] \Big/ b_i, \quad i \in \mathbb{N}_n,\ J_{b_i} \subseteq [0,1] \tag{1}$$

Now, $\tilde{b}$ is bounded by two *Lower* and *Upper* primary membership functions[2] called $\underline{\mu}_{\tilde{b}}$ with parameters $\check{\underline{b}}$ and $\hat{\underline{b}}$ and $\bar{\mu}_{\tilde{b}}$ with parameters $\check{\overline{b}}$ and $\hat{\overline{b}}$ respectively. First two distance measurements $\triangle$ and ∇ are defined as follows.

Definition 22 (Figueroa in [3]). *Consider an IT2FLP problem with restrictions in the form $\leqslant$[3]. Then $\triangle$ is defined as the distance between $\check{\underline{b}}$ and $\check{\overline{b}}$, $\triangle = \check{\overline{b}} - \check{\underline{b}}$ and ∇ is defined as the distance between $\hat{\underline{b}}$ and $\hat{\overline{b}}$, $\nabla = \hat{\overline{b}} - \hat{\underline{b}}$.*

Henceforth, three cases can be identified in an IT2 RHS environment: Uncertain ∇, uncertain $\triangle$ and joint uncertain ∇ & $\triangle$. Uncertain ∇ is a case in which $\nabla > 0$ with $\triangle = 0$, uncertain $\triangle$ is a case in which $\triangle > 0$ with $\nabla = 0$, uncertain $\triangle$ is a case in which $\triangle > 0$ with $\triangle = \nabla$ and joint uncertain $\triangle$ & ∇ is a case in which $\triangle > 0$ and $\nabla > 0$ where $\triangle \neq \nabla$. Its graphical representation is displayed in Figure 1[4]. Note that the following statement is mandatory: $\{[\check{\underline{b}}_i, \hat{\underline{b}}_i] \cap [\check{\overline{b}}_i, \hat{\overline{b}}_i]\} > 0\ \forall\ i \in \mathbb{N}_n$.

With these definitions we can define an *Uncertain Space of Solutions* of a FLP problem, which is defined through linear fuzzy sets and its solution can be reached through convex optimization methods.

Figueroa in [5] defined the notion of an *Interval Type-2 Fuzy Constraint*, as follows.

Definition 23 (Interval Type-2 Fuzzy Constraint). *The set conformed by all crisp possible values of $x \in \mathbb{R}^n$ which satisfy an Interval Type-2 fuzzy partial order $\succsim, \precsim$ or $\approx$ with an Interval Type-2 membership function $\tilde{b}_i$ is an Interval Type-2 Fuzzy Constraint.*

[1] A Fuzzy Space is defined by the interval $b_i \subseteq \tilde{b}$.

[2] For simplicity effects, only linear fuzzy sets are considered.

[3] In the case of $\geqslant$, a symmetric reasoning can be used to solve the problem.

[4] FOU is the acronym for *Foot of Uncertainty.*

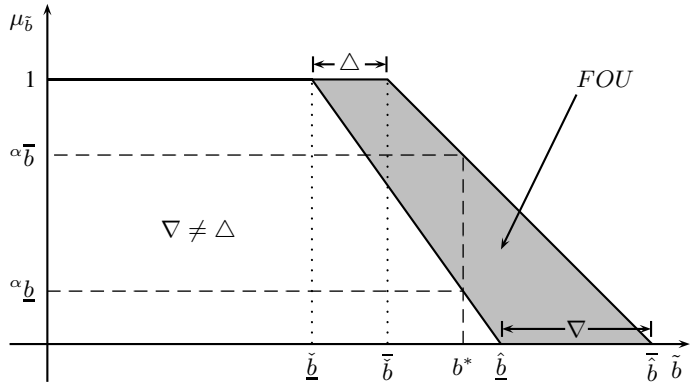


Fig. 1. IT2FS RHS with Joint Uncertain $\triangle$ & ∇

2.1 The IT2FLP Model

Given the notion of an IT2 constraint and by using the classical definition of a FLP, an uncertain constrained FLP model can be defined as follows:

$$\begin{gathered} Opt\,\{c(x)+c_0\} \\ s.t. \\ Ax \precsim \tilde{b} \\ x \geqslant 0 \end{gathered} \tag{2}$$

where $x \in \mathbb{R}^n$, $c(x) = c'x$, $c_0 \in \mathbb{R}$, $A_{n\times m} \in \mathbb{R}^{n\times m}$. Note that $\tilde{b}$ is an IT2FS defined by two primary membership functions $\underline{\mu}_b$ and $\bar{\mu}_b$.

Two possible partial orders $\precsim$ and $\succsim$ with IT2 fuzzy RHS exist, so the handle of each one defines the nature of the problem. In this approach, only linear membership functions are used because the main goal is to get linear models, easily to be optimized with classical algorithms. See the appendix to see the mathematical form of each partial order $\precsim$ and $\succsim$.

3 Computing the Optimality Bounds of an FLP with IT2 Constraints

An IT2FLP model involves an infinite amount of T1FS embedded on the FOU of each RHS regarding the opinions and perceptions of different analysts of the problem.

The Zimmermman's T1FS model computes the optimal bounds of the problem to construct a fuzzy set of optimal solutions that is also defined as a linear fuzzy set through the Zadeh's extension principle, so a new fuzzy set $\tilde{z}$ can be obtained through the computation of bounds of the original problem.

The fuzzy set of optimal solutions can be obtained by applying the Zadeh's extension principle and the computation of the optimal bounds of the problem. This means that $\tilde{z}$ is a function of the optimal values of the problem $z^* = c(x^*)$ regarding the bounds of $\tilde{b}$.

In the domain of x, the above is currently the problem of finding a vector of solutions $x \in \mathbb{R}^n$ such that:

$$\max_{x\in\mathbb{R}^n} {}^{\alpha}\left\{\bigcap_{i=1}^{m}\{{}^{\alpha}\breve{b}_i, {}^{\alpha}\tilde{b}_i, b_i\}\bigcap\tilde{z}\right\} \tag{3}$$

Here, $\breve{}$ denotes a Type-1 fuzzy set, ${}^{\alpha}\breve{b}_i$ and ${}^{\alpha}\tilde{b}_i$ are α-cuts on Type-1 and Type-2 fuzzy sets, respectively. The membership function of $\tilde{z}$ is defined as

$$\mu_{\tilde{z}}(z^*) = \sup_{z^*=c(x^*)} \min_{i} \left\{\mu_{\tilde{b}_i}(x^*) \,|\, x^* \in \mathbb{R}^n\right\} \tag{4}$$

Now, like $\mu_{\tilde{z}}$ is a linear function of x^*, $z^* = c(x^*)$ and $\tilde{b}_i$, then we can define its bounds as function of the bounds of $\tilde{b}_i$, as defined in Definition 21. Although Figueroa in [3], [4] and [5] defined some approximations to an optimal solution for IT2FLP problems based on primary α-cuts and the results of the Definition 22, our focus is the computation of $\mu_{\tilde{z}}$ since it is a key aspect of decision making of LP problems.

Zimmermann defined in [1] and [2] a method for FLP with soft constraints called himself as a *symmetric* approach. His method is based on the computation of the optimal solution of the problem by using the bounds of the Type-1 fuzzy sets $\breve{b}_i$. It is an efficient approach due to all computations are made using the simplex method with linear fuzzy sets.

In this way, we can extend this property to the IT2FLP by using the (4), and keeping in mind that a FLP can be solved by using convex optimization tools, we can compose $\tilde{z}$ through $\tilde{b}_i(Ax)$ and $z^* = c(x^*)$.

Note that $\tilde{b}_i$ is a fuzzy partial order in the form $\precsim$ or $\succsim$ (See Appendix) with well defined parameters $\underline{\breve{b}}, \overline{\breve{b}}, \underline{\hat{b}}$ and $\overline{\hat{b}}$. Mathematically speaking we can compute the bounds of the global optimization problem by using each one of these parameters. To do so, the following relations are defined

Theorem 31 (Bounds of $\tilde{z}$). *Let (2) be an optimization problem where $\tilde{b}$ is composed by $\underline{\mu}_{\tilde{b}}$ with parameters $\underline{\breve{b}}$ and $\underline{\hat{b}}$, and $\bar{\mu}_{\tilde{b}}$ parametrized by $\overline{\breve{b}}$ and $\overline{\hat{b}}$ (See Appendix). Then the fuzzy set $\mu_{\tilde{z}}$ is composed by $\underline{\mu}_{\tilde{z}}$ and $\bar{\mu}_{\tilde{z}}$ in the following way.*

$$\underline{\mu}_{\tilde{z}}(z;\underline{\breve{z}},\underline{\hat{z}}) = \begin{cases} 0, & z \leqslant \underline{\breve{z}} \\ \dfrac{z-\underline{\breve{z}}}{\underline{\hat{z}}-\underline{\breve{z}}}, & \underline{\breve{z}} \leqslant z \leqslant \underline{\hat{z}} \\ 1, & z \geqslant \underline{\hat{z}} \end{cases} \tag{5}$$

$$\bar{\mu}_{\tilde{z}}(z;\overline{\breve{z}},\overline{\hat{z}}) = \begin{cases} 0, & z \leqslant \overline{\breve{z}} \\ \dfrac{z-\overline{\breve{z}}}{\overline{\hat{z}}-\overline{\breve{z}}}, & \overline{\breve{z}} \leqslant z \leqslant \overline{\hat{z}} \\ 1, & z \geqslant \overline{\hat{z}} \end{cases} \tag{6}$$

where $\underline{\breve{z}}, \underline{\hat{z}}, \overline{\breve{z}}$, and $\overline{\hat{z}}$ are crisp parameters computed from $\tilde{b}$.

Proof. Firstly, we can assume from Definition 21 that:

$$\underline{\hat{b}} \geqslant \underline{\check{b}} \tag{7}$$

$$\overline{\hat{b}} \geqslant \overline{\check{b}} \tag{8}$$

$$\underline{\check{b}} \leqslant \overline{\check{b}} \tag{9}$$

$$\underline{\hat{b}} \leqslant \overline{\hat{b}} \tag{10}$$

Now, replacing b by each of the above parameters in the following crisp LP problem:

$$\begin{gathered} \max z = c(x) + c_0 \\ \textit{s.t.} \\ Ax \leqslant b \\ x \geqslant 0 \end{gathered} \tag{11}$$

We obtain the following optimal results as functions of z^*:

$$\underline{\check{b}} \rightarrow \underline{\check{z}} \tag{12}$$

$$\underline{\hat{b}} \rightarrow \underline{\hat{z}} \tag{13}$$

$$\overline{\check{b}} \rightarrow \overline{\check{z}} \tag{14}$$

$$\overline{\hat{b}} \rightarrow \overline{\hat{z}} \tag{15}$$

Then, the computation of each bound of $\mu_{\tilde{b}}(Ax)$ leads to a corresponding bound of $\mu_{\tilde{z}}$, and by mathematical induction is clear that:

$$\underline{\hat{z}} \geqslant \underline{\check{z}} \tag{16}$$

$$\overline{\hat{z}} \geqslant \overline{\check{z}} \tag{17}$$

$$\underline{\check{z}} \leqslant \overline{\check{z}} \tag{18}$$

$$\underline{\hat{z}} \leqslant \overline{\hat{z}} \tag{19}$$

So we can compose $\mu_{\tilde{z}}$ by its primary membership functions $\underline{\mu}_{\tilde{z}}$ and $\bar{\mu}_{\tilde{z}}$ in the following way:

$$\underline{\mu}_{\tilde{z}}(z; \underline{\check{z}}, \underline{\hat{z}}) = \begin{cases} 0, & z \leqslant \underline{\check{z}} \\ \dfrac{z - \underline{\check{z}}}{\underline{\hat{z}} - \underline{\check{z}}}, & \underline{\check{z}} \leqslant z \leqslant \underline{\hat{z}} \\ 1, & z \geqslant \underline{\hat{z}} \end{cases}$$

And its upper membership function is:

$$\bar{\mu}_{\tilde{z}}(z; \overline{\check{z}}, \overline{\hat{z}}) = \begin{cases} 0, & z \leqslant \overline{\check{z}} \\ \dfrac{z - \overline{\check{z}}}{\overline{\hat{z}} - \overline{\check{z}}}, & \overline{\check{z}} \leqslant z \leqslant \overline{\hat{z}} \\ 1, & z \geqslant \overline{\hat{z}} \end{cases}$$

Which concludes the proof.

On the other hand, an interesting question arises from the analysis of $\mu_{\tilde{z}}$: Is there exists the possibility of having a linear combination of $\tilde{b}_i$ that reaches out a solution outside $\mu_{\tilde{z}}$? To answer that, we define the following Corollary

Corollary 32 (FOU of $\tilde{z}$). *The footprint of uncertainty of $\tilde{z}$ can be defined as follows*

$$\text{FOU}(\tilde{z}) = \bigcup_{z \in Z} \left[\underline{\mu}_{\tilde{z}}(z^*), \overline{\mu}_{\tilde{z}}(z^*) \right] \tag{20}$$

$$\text{FOU}(\tilde{z}) = \bigcup_{z \in Z} J_{z^*} \tag{21}$$

where J_{z^*} is the primary membership function of $\tilde{z}$ composed by (29) and (30).

The Corollary 32 defines that the union of all possible Type-1 fuzzy sets is embedded into $FOU(\tilde{z})$. By using the definitions 21 and 22 it is easy to show that there is no choice to have a linear fuzzy set outside ∇ and $\triangle$, so the question has a simple answer: No. This is not possible if we consider that $\mu_{\tilde{b}}$ is composed by linear primary membership functions and $\mu_{\tilde{z}}$ is also a linear fuzzy set, so there is no any possibility to define a linear combination of $\tilde{b}_i$ that reaches a value outside $\mu_{\tilde{z}}$.

3.1 Bounds of $\tilde{z}$

Hitherto, $\mu_{\tilde{z}}$ is composed by all the possible linear combinations of $\tilde{b}_i$ viewed as Type-1 linear fuzzy sets embedded into the FOU of $\tilde{b}_i$. The membership function of $\tilde{z}$ is derived from $\tilde{b}_i$, so this uncertain fuzzy set of optimal solutions inherit their properties. This set involves all optimal values of x regarding $c(x)$ and $\tilde{b}$. This helps us to understand that all values of z are contained into the support of $\tilde{z}$, $supp(\tilde{z})$. According to Niewiadomski in [14] and [15], we obtain the bounds of $\tilde{z}$ through $supp(\tilde{z})$, as follows

$$supp(\tilde{z}) = [\underline{\check{z}}, \overline{\hat{z}}] \tag{22}$$

Therefore, $\tilde{z}$ is bounded through this interval, as well as its universe of discourse, so the analyst should think that he could not reach either a less optimal than $\underline{\check{z}}$ or a higher optimal than $\overline{\hat{z}}$.

On the other hand, all optimal linear combinations of $\tilde{b}_i$ are enclosed on $supp(\tilde{z})$ through $z^* = c(x^*)$ and (4). This means that any optimal solution of the problem is condensed into $\tilde{z}$ and the use of $\tilde{b}_i$ through its α-cuts is bounded by $[\underline{\check{z}}, \overline{\hat{z}}]$.

Under an engineering's point of view, the selection of a particular Type-1 fuzzy set embedded into the FOU of $\tilde{b}_i$ should reach a particular Type-1 fuzzy set $\check{z}$ embedded into the FOU of $\tilde{z}$. Moreover, any crisp solution obtained from $\tilde{z}$ (See Figueroa in [3], [4] and [5]) has attached a deffuzification degree regarding a Type-1 fuzzy set.

This means that the problem with uncertain constraints has a crisp optimal solution that can be reached from a Type-1 fuzzy set embedded into the FOU of the uncertain constraints which is a soft constraint itself, so this solution can be obtained from the computation of an α-cut of the soft constraints.

Although Figueroa in [3], [4] and [5] proposed some approaches for solving LP problems with IT2FC, the searching of an optimal solution is not a simple problem.

The sense of the words "Optimal Solution" in an uncertain scheme is a Type-2 fuzzy concept itself, and its definition should be represented by a Type-2 fuzzy set, just like we presented in this paper.

4 Illustrative Example

The following is a maximization example where the main idea is to compute $\tilde{z}$ from $z^* = c(x^*)$, (4) and $\tilde{b}$. All parameters of $\tilde{b}$, c and A are defined in matrix form.

$$A = \begin{bmatrix} 5 & 3 & 7 \\ 10 & 4 & 9 \\ 4 & 6 & 3 \\ 2 & 7 & 7 \\ 5 & 6 & 11 \end{bmatrix} ; \underline{\check{b}} = \begin{bmatrix} 50 \\ 70 \\ 40 \\ 60 \\ 40 \end{bmatrix} ; \underline{\hat{b}} = \begin{bmatrix} 72 \\ 104 \\ 65 \\ 95 \\ 80 \end{bmatrix}$$

$$\bar{\check{b}} = \begin{bmatrix} 60 \\ 80 \\ 55 \\ 75 \\ 57 \end{bmatrix} ; \bar{\hat{b}} = \begin{bmatrix} 95 \\ 110 \\ 77 \\ 102 \\ 98 \end{bmatrix} ; c = \begin{bmatrix} 12 \\ 17 \\ 9 \end{bmatrix}$$

By using the Theorem (31) and (11) to (15), we obtain the following results:

$$\underline{\check{z}} = 113.33 \tag{23}$$

$$\bar{\check{z}} = 157.16 \tag{24}$$

$$\underline{\hat{z}} = 189.68 \tag{25}$$

$$\bar{\hat{z}} = 223.5 \tag{26}$$

A graphical representation of the bounds of $\tilde{z}$ is presented in Figure 2.

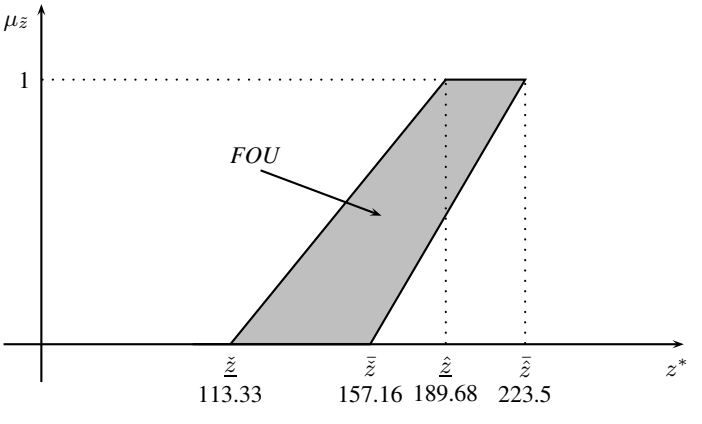


Fig. 2. Interval Type-2 Fuzzy Set $\tilde{z}$ and its FOU

5 Concluding Remarks

The following concluding remarks are proposed:

1. An analytical way to compute the bounds of the optimal values that a FLP can take is presented and a discussion about its definition is done.
2. An extension of the results of Zimmermann is provided and its scope is extended to a Type-2 fuzzy sets environment. The concept of a symmetric FLP problem is used to define the fuzzy set of optimal solutions of an IT2FLP model.
3. An infinite set of choices of b embedded in $\tilde{b}$, $b \in supp(\tilde{b})$ could be defined. Different experts or analysts can define several $\breve{b}$, so the IT2FS approach involves all their estimates and perceptions about a fuzzy constraint.
4. The problem of finding a global optima is still an ongoing problem. Some ideas about Type-reduction (See Melgarejo in [16], [12], [17], and Mendel in [6], [10], [18] and [19]) and interval optimization (See Lodwick in [20], [21] and [22], and Kreinovich in [23] and [24]) can be helpful on the way to design a general optimization method for IT2FLP.

Finally, it is important to recall that the scope of this paper is to define $\tilde{z}$ as a function of $\tilde{b}$ and its parameters. This is a starting point for researchers who want to optimize uncertain and upcoming problems. The use of linguistic uncertainty in optimization arises as the next step in operations research and decision making.

5.1 Further Topics

The theory of Generalized Interval Type-2 Fuzzy Sets (GT2 FS) arises as a new challenge. This one has an additional degree of freedom that should be considered in the modeling process, this feature is the secondary membership function $f_x(u)/u$ which induces to new directions.

References

1. Zimmermann, H.J.: Fuzzy programming and Linear Programming with several objective functions. Fuzzy Sets and Systems 1, 45–55 (1978)
2. Zimmermann, H.J., Fullér, R.: Fuzzy Reasoning for solving fuzzy Mathematical Programming Problems. Fuzzy Sets and Systems 60, 121–133 (1993)
3. Figueroa, J.C.: Linear programming with interval type-2 fuzzy right hand side parameters. In: 2008 Annual Meeting of the IEEE North American Fuzzy Information Processing Society (NAFIPS) (2008)
4. Figueroa, J.C.: Solving fuzzy linear programming problems with interval type-2 RHS. In: 2009 Conference on Systems, Man and Cybernetics. IEEE (2009)
5. Figueroa, J.C.: Interval type-2 fuzzy linear programming: Uncertain constraints. In: 2011 IEEE Symposium Series on Computational Intelligence (2011)
6. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall (1994)
7. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. IEEE Transactions on Fuzzy Systems 14, 808–821 (2006)

8. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Transactions on Fuzzy Systems 10, 117–127 (2002)
9. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems: Theory and design. IEEE Transactions on Fuzzy Systems 8, 535–550 (2000)
10. Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. Fuzzy Sets and Systems 122, 327–348 (2001)
11. Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. Fuzzy Sets and Systems 17, 643–658 (1999)
12. Melgarejo, M.A.: A Fast Recursive Method to compute the Generalized Centroid of an Interval Type-2 Fuzzy Set. In: Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), pp. 190–194. IEEE (2007)
13. Melgarejo, M.A.: Implementing Interval Type-2 Fuzzy processors. IEEE Computational Intelligence Magazine 2, 63–71 (2007)
14. Niewiadomski, A.: On Type-2 fuzzy logic and linguistic summarization of databases. Bulletin of the Section of Logic 38, 215–227 (2009)
15. Niewiadomski, A.: Imprecision Measures for Type-2 Fuzzy Sets: Applications to Linguistic Summarization of Databases. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 285–294. Springer, Heidelberg (2008)
16. Melgarejo, M., Peña, C.A., Sanchez, E.: A genetic-fuzzy system approach to control a model of the hiv infection dynamics. In: IEEE (ed.) 2006 International Conference on Fuzzy Systems, pp. 2323–2330. IEEE (2006)
17. Melgarejo, M., Bernal, H., Duran, K.: Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set. In: 2008 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), vol. 27, pp. 1–6. IEEE (2008)
18. Mendel, J.M., Liu, F.: Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set. IEEE Transactions on Fuzzy Systems 15, 309–320 (2007)
19. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Information Sciences 132, 195–220 (2001)
20. Lodwick, W.A., Jamison, K.D.: Theoretical and semantic distinctions of fuzzy, possibilistic, and mixed fuzzy/possibilistic optimization. Fuzzy Sets and Systems 158, 1861–1872 (2007)
21. Lodwick, W.A., Bachman, K.A.: Solving Large-Scale Fuzzy and Possibilistic Optmization Problems. Fuzzy Optimization and Decision Making 4, 257–278 (2005)
22. Lodwick, W.A.: Fuzzy Optimization: Recent Advances and Applications. Springer Series Studies in Fuzziness and Soft Computing (2010)
23. Kreinovich, V., Lakeyev, A., Rohn, J., Kahl, P.: Computational complexity and feasibility of data processing and interval computations. Kluwer Academic Publishers, Dordrecht (1997)
24. Kearfott, R.B., Kreinovich, V.: Beyond convex? global optimization is feasible only for convex objective functions: A theorem. Journal of Global Optimization 33, 617–624 (2005)

Appendix: Membership Functions for $\precsim$ and $\succsim$

In order to define $\precsim$ and $\succsim$ in (2), the lower membership function for $\precsim$ is:

$$\underline{\mu}_{\tilde{b}}(x;\underline{\check{b}},\underline{\hat{b}}) = \begin{cases} 1, & x \leqslant \underline{\check{b}} \\ \dfrac{\underline{\hat{b}} - x}{\underline{\hat{b}} - \underline{\check{b}}}, & \underline{\check{b}} \leqslant x \leqslant \underline{\hat{b}} \\ 0, & x \geqslant \underline{\hat{b}} \end{cases} \tag{27}$$

And its upper membership function is:

$$\bar{\mu}_{\tilde{b}}(x;\bar{\bar{b}},\bar{\hat{b}}) = \begin{cases} 1, & x \leqslant \bar{\bar{b}} \\ \dfrac{\bar{\hat{b}} - x}{\bar{\hat{b}} - \bar{\bar{b}}}, & \bar{\bar{b}} \leqslant x \leqslant \bar{\hat{b}} \\ 0, & x \geqslant \bar{\hat{b}} \end{cases} \tag{28}$$

The Lower membership function for $\succsim$ is:

$$\underline{\mu}_{\tilde{b}}(x;\underline{\check{b}},\underline{\hat{b}}) = \begin{cases} 0, & x \leqslant \underline{\check{b}} \\ \dfrac{x - \underline{\check{b}}}{\underline{\hat{b}} - \underline{\check{b}}}, & \underline{\check{b}} \leqslant x \leqslant \underline{\hat{b}} \\ 1, & x \geqslant \underline{\hat{b}} \end{cases} \tag{29}$$

And its upper membership function is:

$$\bar{\mu}_{\tilde{b}}(x;\bar{\bar{b}},\bar{\hat{b}}) = \begin{cases} 0, & x \leqslant \bar{\bar{b}} \\ \dfrac{x - \bar{\bar{b}}}{\bar{\hat{b}} - \bar{\bar{b}}}, & \bar{\bar{b}} \leqslant x \leqslant \bar{\hat{b}} \\ 1, & x \geqslant \bar{\hat{b}} \end{cases} \tag{30}$$

A graphical representation of (27) and (28) is shown in the Figure 1. A symmetrical reasoning can be done for visualizing (29) and (30).

Supervision Strategy of a Solar Volumetric Receiver Using NN and Rule Based Techniques

Ramón Ferreiro García[1], José Luis Calvo Rolle[2], and Francisco Javier Pérez Castelo[2]

[1] ETSNM, Dept. Industrial Eng. University of La Coruna, Spain
ferreiro@udc.es
[2] EUP, Dept. Industrial Eng. University of La Coruna, Spain
{jlcalvo,javierpc}@udc.es

Abstract. Most nonlinear processes suffer from lack of detectability when model based techniques are applied to IFDI (intelligent fault detection and isolation) tasks. Generally, all types of nonlinear processes will also suffer from lack of detectability due to the inherent ambiguity in discerning faults in the process, sensors and/or actuators. This work deals with a strategy to detect and isolate process and/or sensor faults by combining neural networks based on functional approximation procedures associated with recursive rule using techniques for a parity space approach. For this work, a case study dealing with the supervision of a solar volumetric receiver was performed using the proposed intelligent techniques, and produced reliable and acceptable IFDI results.

Keywords: Conjugate Gradient, Fault Detection, Fault Isolation, Neural Networks, Parity Space, Residual Generation.

1 Introduction

The detection and isolation of features indicating undesired changes in the sensors, actuators, behavior or performance of a process, is strongly associated with safety. When models describing a process function accurately under model-based approaches, the problem of fault detection may be solved by observer-type filters. These filters generate the so-called residuals computed from the inputs and outputs of the process. The generation of these residual signals is the first stage in the problem of fault detection and isolation (FDI). To perform reliable FDI tasks, the residuals must be insensitive to model errors and highly sensitive to the faults being considered. Thus, the residuals are designed so that the effects of possible faults are enhanced, which in turn increases their detectability. The residuals must also respond quickly. In order to detect the presence of faults, the residuals are tested. Several FDI methods have been previously published [1-5], among the classic bibliography [6-8].

1.1 Model Based on Fault Detection Methods

Fault detection methods based on process and signal models include actuators, processes and sensors for which input and output variables must be accurately

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 577–587, 2012.

measured. Such methods deal mainly with parameter estimation, state observers and parity equation methods. When measuring instrumentation fails, the fault detection techniques based on the use of input/output measurement yield ambiguous and/or erroneous and useless results.

Intensive research on models based on fault detection methods has been carried out in recent decades. The following is a brief list of process model based fault detection methods:

1. Fault detection with parameter estimation [4]
   - -Equation error methods
   - -Output error methods

2. Fault detection with state-estimation.
   (a) Dedicated observers for multi-output processes.
      - State Observe, excited by one output [9].
      - Kalman filter, excited by all outputs [10], [1].
      - Bank of state observers, excited by all outputs [1].
      - Bank of state observers, excited by single outputs [3].
      - Bank of state observers, excited by all outputs except one [3].

   (b) Fault detection filters for multi-output processes [11].

3. Fault detection with parity equations [2], [12], [13].
   (a) Output error methods.
   (b) Polynomial error methods.

4. Fault detection using analytical redundancy [14].
   (a) Static analytical redundancy.
   (b) Dynamic analytical redundancy.

No general method exists for the FDI problem. Successful FDI applications are based on strategies that combine several methods. Practical FDI systems apply analytical redundancy using the first-principles like action-reaction balances such as mass flow rate balance, energy flow rate balance, force/torque/power balances and more commonly, the mathematical balance of any cause-effect dynamic equilibrium condition.

When the previously mentioned diagnosing technique is applied to nonlinear systems, it suffers from lack of detectability. Residuals are the outcomes of consistency checks between the plant observations and a mathematical model. The three main ways to generate residuals are parameter estimation, observers, and parity relations. For parameter estimation, the residuals are the difference between the nominal model parameters and the estimated model parameters. Derivations in the model parameters serve as the basis for detecting and isolating faults.

In many practical applications of FDI, the process parameters can be either partially known or not known at all. Unknown parameters can be determined by parameter estimation methods measuring input and output signals, assuming the basic model structure exists. There are two conventional approaches based on the minimization of equation error and output error that are commonly used. The first one is least squares in non-recursive or recursive form. The second one requires numerical optimization methods and therefore iterative procedures, but may be more accurate under the influence of process disturbances. The symptoms are deviations from the process parameters. As the process parameters depend on physically defined process coefficients, determination of changes usually allows deeper insight and facilitates fault diagnosis [2].

These two conventional methods of parameter estimation usually require a process input excitation and are especially suitable for the detection of multiplicative faults. The parameter estimation requires an input/output correct measuring system. Some drawbacks of these methods are:

- The possibility of faulty measuring signals.
- Unknown model structure and/or unknown model changes.

Many complex problems that use intelligent supervision strategies require hybrid intelligent systems techniques that integrate several hybrid artificial intelligent techniques associated with expert or rule based systems, fuzzy logic, neural networks, neuro-fuzzy and learning algorithms, including genetic algorithms [15-19]. Some of these techniques are applied to achieve the proposed supervision and IFDI objectives.

1.2 Goals to Be Achieved

The basic principles of system diagnosis are based on residuals. Residual generation supposes a troublesome task in the case of sensor diagnostics. The most commonly applied techniques are all based on physical or analytical redundancy, though physical redundancy based on solutions require double or triple the number of sensors for each measure. Instead of applying the more expensive physical redundancy, the analytical or model-based redundancy approaches are more often integrated, thus reducing maintenance costs. Figure 1 shows a block diagram of the methods most commonly applied to the supervision of industrial processes for IFDI. There are many more methods for system diagnosis such as time frequency methods (Wavelet Signal Processing) or neuro-fuzzy classification techniques, which are among the most relevant.

The grey boxes in figure 1 deal with the concepts of applied diagnostic referred to in this work. As shown in figure 1, supervision by IFDI is performed using analytical redundancy, parity relations into a parity space approach, with the help of analytical models, NN based models, and a rule based system to solve a decision making task.

When the IFDI methods are applied to nonlinear processes, they suffer from lack of detectability. Thus, this work is oriented towards functional approximation techniques using well known models based on feedforward NN [20-25]. This research then focuses on the problem of fault detection, fault isolation, and fault estimation by

applying a parity space approach where plant modules are achieved by steady state functional approximation techniques on the basis of NNs, where residual generation is achieved by comparing data measured in real-time with data achieved from models of trained NN. The tasks to be carried out consist of achieving a consistent NN-based steady state model of the plant. Model output is then used as a residual generator. The residuals that are obtained are evaluated by a set of rules for detecting and isolating plant faults, which include process faults and sensor faults.

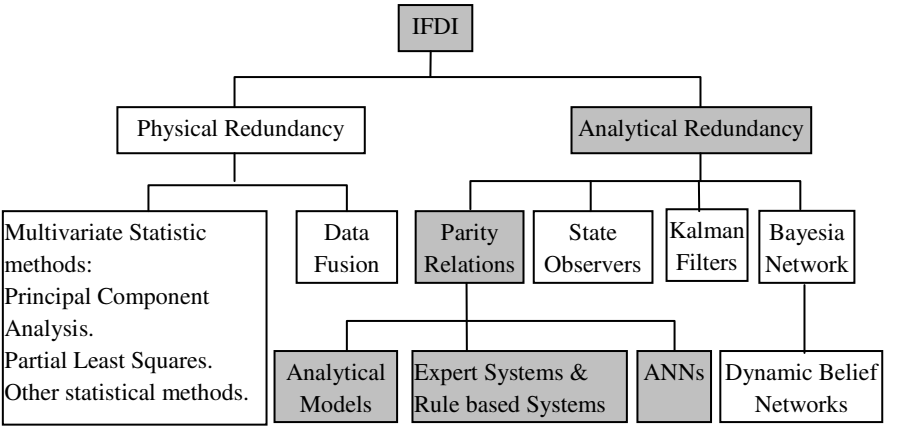


Fig. 1. Some generic methods for IFDI

The following sections will describe the proposed complementary technique, as an option to substitute or complement some of the mentioned conventional techniques.

2 Causality Based Modeling Approach

The relationship between cause and effect is known as the principle of causality. The causality of two events describes to what extent one event is caused by the other. When there is open loop causality, there is a measure of predictability of a second event caused by the first event.

Functional approximation is currently one of the most appropriate techniques for describing nonlinear models of plant modules such as process, actuators and/or sensors. In this task, feedforward neural networks have been effectively applied.

With regards to thermal efficiency, the analytical model of the volumetric receiver process is defined by equations (1) and (2) by applying energy balances.

$$Qu = Qu_1 = \dot{m} \cdot Cp \cdot (To - Ti) = Qu_2 = A_{Ab} \cdot U \cdot (T_A - T_F) \tag{1}$$

$$Qs = A_{Ap} \cdot C \cdot E_S \tag{2}$$

where

η	is the thermal efficiency of the volumetric receiver.
Qu	is the useful heat.
Qs	is the supplied heat.
$\dot{m}$	is the fluid mass flow rate.
Cp	is the fluid specific heat capacity.
To	is the fluid output temperature from the receiver.
Ti	is the fluid input temperature to the receiver.
A_{Ap}	is the volumetric receiver aperture area.
A_{Ab}	is the absorber area.
U	is the heat transfer coefficient.
T_A	is the absorber temperature.
T_F	is the average fluid temperature.
C	is the concentration factor.
E_S	is the direct solar normal radiation density.

Using this approach, some variables, such as temperatures, flow rates and energy flow density, can be directly measured. However, some process parameters must be indirectly obtained, such as the heat transfer coefficient, whose variation is a symptom of process (absorber) deterioration due to the solar temperature.

As shown in previous equations, several variables and parameters are involved. A complementary model is required to supervise the thermal efficiency of the volumetric receiver using analytical redundancy. This model is achieved from experimental data, which is used to train a NN. Thus, there are at least three definitions that exist for thermal efficiency, as shown in equations (3):

$$\begin{aligned} \eta &= \frac{Qu}{Qs} = f(C, T_F) \\ \eta_{A1} &= \frac{Qu_1}{A_{Ap} \cdot C \cdot E_S} = \frac{A_{Ab} \cdot U \cdot (T_A - T_F)}{A_{Ap} \cdot C \cdot E_S} \\ \eta_{A2} &= \frac{Qu_2}{A_{Ap} \cdot C \cdot E_S} = \frac{\dot{m} \cdot Cp \cdot (To - Ti)}{A_{Ap} \cdot C \cdot E_S} \end{aligned} \tag{3}$$

Under free faults condition, the defined modes of efficiency are identical, so that

$$\eta = \eta_{A1} = \eta_{A2} \tag{4}$$

Similarly, it follows that

$$\begin{aligned} &Qu \cong Qu_1 \cong Qu_2 \rightarrow \\ &\eta \cdot A_{Ap} \cdot C \cdot E_S \cong \dot{m} \cdot Cp \cdot (To - Ti) \cong A_{Ab} \cdot U \cdot (T_A - T_F) \end{aligned} \tag{5}$$

2.1 On Functional Approximation by Feedforward NNs

There are several steps that must be followed when building any empirical model, including neural models. They include: Data collection, Pre-processing, Design, Training/Testing and Verification.

One of the challenges of building neural models is the many inherent pseudo-correlations evident in process data. Auto-correlation, cross-correlation, and other statistical concerns will often confuse the system that generates models into believing that a true correlation exists, when in reality it is only temporary or coincidental. Training a neural model by using as much laboratory and process data from as many sources as possible is the best way to ensure that true correlations are correctly identified. Ideally, no step-testing or manual interruption of the process is necessary during the development and training of a network. A highly stable process can make it easier to correlate process data with lab analyses. Some process variability is required to observe a correlation. If the process does not vary sufficiently, or if the process wanders into a range where no training data has been collected, the model will not be reliable. Taking these possibilities into account, it follows that slow changes or steps that are held for relatively long periods are the best types of input for training a neural model. The NN model configuration makes it possible to discard any measured variables that have little or no effect on model outputs, thus effectively eliminating them from further input training sets. In fact we are interested in being able to identify the variables with the greatest significance. To this end, statistical techniques such as Partial Least Squares (PLS) and Principle Component Analysis (PCA) are used to identify not only the sensitivity of each input, but the sensitivity with regards to the time required to reach steady-state. This enables the identification of time delays between the moment when a process measurement is observed and the moment when the corresponding change in lab property is observed. Each input takes its own time delay into account, and the pre-processing analysis helps identify and eliminate inputs that don't contribute to the model. After performing data analysis, the thermal efficiency can be described by a simple nonlinear experimental function such as that provided in equation (6).

$$\eta \equiv f(T_F, C) \equiv \frac{Qu}{Qs} \cong \frac{\dot{m} \cdot Cp \cdot (To - Ti)}{A_{Ap} \cdot C \cdot E_S} \cong \frac{A_{Ab} \cdot U \cdot (T_A - T_F)}{A_{Ap} \cdot C \cdot E_S} \tag{6}$$

After variables that exhibit no effect on the NN output have been excluded from NN function in equation (6), the result is a simple experimental NN based function output, as shown in figure 2.

Such an experimental model depends on only two measured variables, and represents an advantage with regards to measuring data. Data acquired for the task of estimating efficiency must be reliable. Thus, the measuring system (sensors) is expected to be accurately adjusted and free of any faults. Under the assumption of such conditions, the pre-processed training data, where input variables that have not influenced the output (efficiency) have been eliminated from training sets, are the concentration factor and the average temperature.

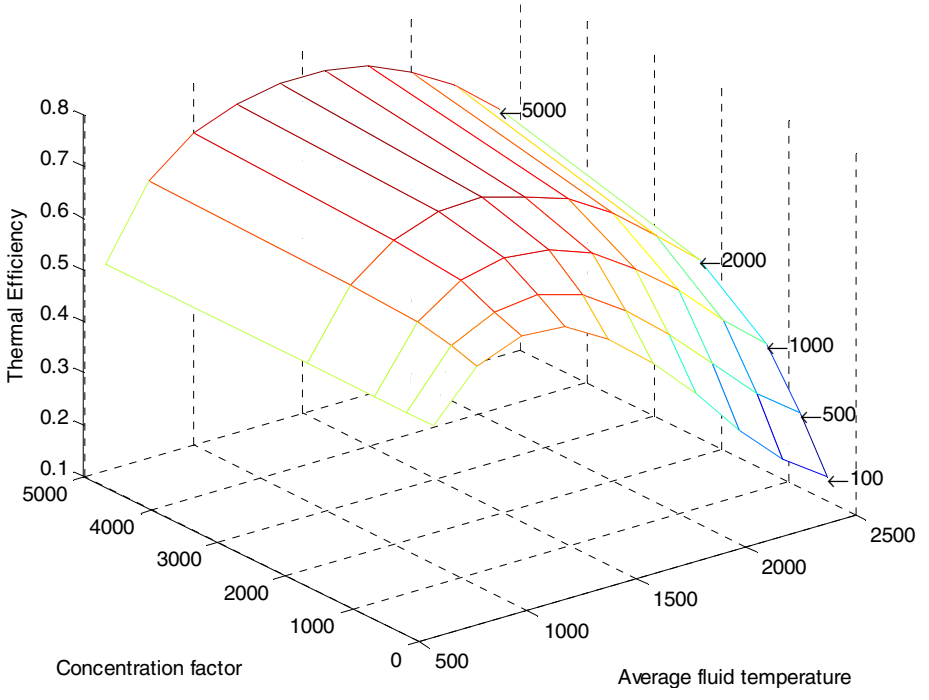


Fig. 2. Thermal efficiency as a function of the average absorber temperature T_{Ab} and solar energy concentration factor C

3 Supervision Strategy

The task of supervising is carried out by processing a set of rules which take into account the residuals achieved by comparing the computed results of efficiency between NN based models and analytical models. The proposed set of rules is described in table 1.

Table 1. The necessary set of rules to verify process and sensors correctness (rule 1), and to detect and isolate some faults (rules 2, 3 and 4) processed in the recursive supervision task.

Rule	Premises		Conclusion
1	IF $\eta = \eta_{A1}$ AND $\eta = \eta_{A2}$	THEN	Sensors (OK) and Process (OK)
2	IF Qu1 $\neq$ Qu2	THEN	At least one involved sensor fail
3	IF $\eta = \eta_{A1}$ AND $\eta \neq \eta_{A2}$	THEN	Sensors of η_{A2} fail
4	IF $\eta \neq \eta_{A1}$ AND $\eta = \eta_{A2}$	THEN	Sensors of η_{A1} fail

Rule 1 means that both the complete measuring system and the process parameters are correct.

At this point, assuming the measurements of C and T_F are correct, which means assuming that To and Ti are also correct, since $T_F = (To+Ti)/2$, then rules 2, 3 and 4 apply.

When processing rules 2, 3 and 4, potential sensor faults may be due to: T_A, To, Ti, $\dot{m}$, and E_S, while potential process faults are the parameter U due to the absorber deterioration under the same working fluid, which implies Cp constant.

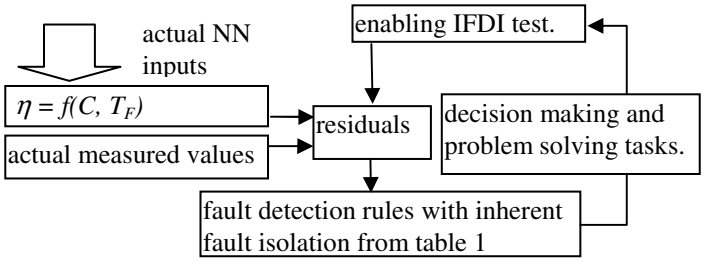


Fig. 3. A flowchart of the supervision strategy

A flowchart to schedule the supervision task is implemented according to the chart shown in figure 3. To generate residuals, the output of the neural network model $\eta = f(C, T_F)$ is compared to the current analytical and computed variable. The existence of any discrepancies, which are due to computation truncation and measuring imprecisions, must be taken into account. The conclusions given by the rules applied in table 1 are then carried out. When the diagnostic task is enabled, the flowchart depicted in figure 3 is performed sequentially and recursively.

Some redundant rules can be applied in order to verify reliability. Table 2 describes a set of such rules.

Table 2. Optional redundant rules to verify process and sensors

Rule	Premises		Conclusion	
1	IF Qu = Qu1 AND Qu= Qu2	THEN	Sensors (OK) and Process (OK)	
2	IF Qu = Qu1 AND Qu ≠ Qu2	THEN	Sensors of Qu2	FAIL
3	IF Qu ≠ Qu1 AND Qu = Qu2	THEN	Sensors of Qu1	FAIL
4	IF Qu1 = Qu2 AND (Qu1 OR Qu2) ≠ Qu	THEN	Sensors and/or Process	FAIL

3.1 Supervision Results

Extensive tests on the solar volumetric receiver model were carried out to verify the reliability of the supervision strategy by simulation, where the data processed belongs to the set of sensors that were taken into consideration for this study using analytical redundancy, and previously defined by equation 1. Since a NN depends on three inputs (two variables *To, Ti,* and a constant known parameter *C*), verifying the behaviour of the remaining sensors and parameters is a straightforward task that involves processing the set of rules previously related to the decision making task.

Table 3 lists the results of a supervision session. It consists of a series of tests carried out by invoking some previously known faults to verify the supervision strategy response.

As shown in table 3, decision making is an ambiguous task. Analytical redundancy is not completely reliable because available analytical models depend partially on the same sensors. Consequently, deterministic conclusions are not possible for involved sensor faults for tests 2 and 3. Nevertheless, proposed supervision strategy is thought to provide significant assistance in maintaining personnel at effective cost.

Table 3. Test results

Test	Invoked fault	Symptom	Isolation	Decision
1	Sensor T_{Ab}	Qu1 ≠ Qu2, Qu=Qu2	Fault in Qu1	Check Qu1 for T_{Ab} sensors and A_{Ab}.U parameter
2	Sensors To, Ti	Qu1=Qu2, Qu ≠Qu2 and Qu1	Fault in Qu1 and Qu2	Ambiguity: check To, Ti And sensors E_S
3	Sensor E_S	Qu1=Qu2, Qu ≠Qu2 and Qu1	Fault in Qu1,Qu2 and Qs	Ambiguity: check To, Ti And sensors E_S
4	Sensor m'	Qu1 ≠ Qu2, Qu=Qu1	Fault in Qu2	Check Qu2 for m' sensor and fluid properties

4 Conclusions

This study proposed and successfully applied a supervision strategy focused on detecting and isolating solar volumetric receiver under steady state conditions, based on causal NN modeling techniques.

Results show that the detection of a drift fault associated with the process variable (temperature) measuring sensor was successfully achieved with the presence of a correct measuring system.

The detection of a drift fault associated with the process variable (temperature) is not sensible to process disturbances. Such a characteristic is very interesting from the point of view of robustness and system reliability.

By discarding the transient state as a portion of time in which a supervision task cannot be applied, we were able to ensure that such a fault detection scheme is immune (not sensible) to the process disturbances.

This supervising system needs to be updated every time parameter changes are detected, by training the applied neural networks. This characteristic is a serious drawback since this forces the plant to go off-line, which affects productivity.

The most important disadvantage of the applied methodology is the impossibility of detecting faults when the process is in the transient state.

References

1. Willsky, A.S.: A survey of design methods for failure detection systems. Automatica 12, 601–611 (1976)
2. Isermann, R.: Process fault detection based on modeling and estimation methods - a survey. Automatica 20(4), 387–404 (1984)

3. Frank, P.M.: Fault Diagnosis in Dynamic systems via State Estimation - A Survey. In: Tzafestas, S., et al. (eds.) System Fault Diagnostics, Reliability and Related Knowledge-Based Approaches, vol. 1, pp. 35–98. D. Reidel Publíshing Company, Dordrecht (1987)
4. Gertler, J.J.: Survey of Model-Based Failure Detection and Isolation in Complex Plants. IEEE Control Systems Magazine 8(6), 3–11 (1988)
5. Patton, R.J., Chen, J.: A review of parity space approaches to fault diagnosis. In: IFAC Symposium SAFEPROCES 1991, Baden-Baden, Germany, vol. I, pp. 239–256 (1991) (preprints)
6. Himmelblau, D.M.: Fault detection and diagnosis in chemical and petrochemical processes. Elsevier, Amsterdam (1978)
7. Pau, L.F.: Failure Diagnosis and Performance Monitoring. Marcel Dekker, New York (1981)
8. Basseville, M.: Optimal Sensor Location for Detecting Changes in Dynamical Behaviour. Rapport de Recherche No. 498, INRIA (1986)
9. Clark, R.N.: A simplified instrument detection scheme. IEEE Trans. Aerospace Electron. Syst. 14, 558–563 (1978)
10. Mehra, R.K., Peschon, J.: An innovations approach to fault detection and diagnosis in dynamic systems. Automatica 7, 637–640 (1971)
11. Beard, R.V.: Failure accommodation in linear systems through self-reorganization. Rept. MVT-71-1. Man Vehicle Laboratory, Cambridge, Massachusetts (1971)
12. Gertler, J.J.: Analytical Redundancy Methods in Fault Detection and Isolation - Survey and Synthesis. In: Preprints of the IFAC/IMACS-Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS 1991, Baden-Baden, FRG, September 10-13, vol. 1, pp. 9–21 (1991)
13. Patton, R.J., Chen, J.: A review of parity space approaches to fault diagnosis for aerospace systems. J. of Guidance Control Dynamics 17(2), 278–285 (1994)
14. Ragot, J., Maquin, D., Kratz, F.: Observability and redundancy decomposition application to diagnosis. In: Patton, R.J., Frank, P.M., Clark, R.N. (eds.) Issues of Fault Diagnosis for Dynamic Systems, ch. 3, pp. 52–85. Springer, London (2000)
15. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
16. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
17. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
18. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
19. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
20. Hong, S.J., May, G.: Neural Network-Based Real-Time Malfunction Diagnosis of Reactive Ion Etching Using In Situ Metrology Data. IEEE Transactions on Semiconductor Manufacturing 17(3), 408–421 (2004)
21. Abe, Y., Konishi, M., Imai, J.: Neural network based diagnosis method for looper height controller of hot strip mills. In: Proceedings of the First International Conference on Innovative Computing, Information and Control, ICICIC 2006 (2006)

22. Patan, K., Witczak, M., Korbicz, J.: Towards robustness in neural network based fault diagnosis. Int. J. Appl. Math. Computers Sci. 18(4), 443–454 (2008)
23. Garcia, R.F., Rolle, J.L.C., Castelo, F.J.P.: Efficient Plant Supervision Strategy Using NN Based Techniques. In: Graña Romay, M., Corchado, E., Gacia-Sebastian, M.T. (eds.) HAIS 2010, Part I. LNCS(LNAI), vol. 6076, pp. 385–394. Springer, Heidelberg (2010)
24. Garcia, R.F., De Miguel Catoira, A., Sanz, B.F.: FDI and Accommodation Using NN Based Techniques. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010, Part I. LNCS(LNAI), vol. 6076, pp. 395–404. Springer, Heidelberg (2010)

Modeling an Operating System Based on Agents

Javier Palanca Cámara, Marti Navarro, Estefania Argente, Ana Garcia-Fornes, and Vicente Julián

DSIC - Universitat Politècnica de València
Camino de Vera s/n
Valencia, Spain
{jpalanca,mnavarro,eargente,agarcia,vinglada}@dsic.upv.es

Abstract. Operating Systems (OS) are the software parts that abstract the hardware to a higher level to be used by developers, users and other applications. It also ensures, with some limitations, a properly use of all these pieces and computing resources. In the last years, with the enormous success of the Internet, network management has been added to those tasks, allowing us the implementation and evolution of new technologies by providing basic services for them. However, all the recent advances in this area are constructed over the OS, not inside it, which implies that some levels of software abstractions are required to adapt them. As a result, it can be understood that the role of the OS has not reached enough control over new applications, technologies and paradigms creating a lack of security and efficiency. In this paper a new approach for the OS modeling is presented. It is taken into account the emerged necessity of improving it with modern paradigms such as interaction-based computing, cloud-computing and even multi-agent systems.

Keywords: Operating Systems, Multi-Agent Systems.

1 Introduction

Operating Systems (OS) are one of the most complex software products to design, develop and maintain. They must provide a critical functionality and performance and also good results for security, reliability, fault-tolerance and effectiveness. Moreover, their construction become really complicated due to their dependability[4] and efficiency constraints.

As an example we can focus on the *process* abstraction. The definition of *process* for the abstraction of a piece of code compiled and running has resulted quite static and not too flexible. When the implementation of threads was introduced, a lot of changes have had to be performed in the OS kernels. Additionally, its behavior is far away from the current trends of the new developed paradigms, such as the *Computing as Interaction* [10], peer-to-peer[12], autonomous systems[3] or distributed computing like cloud computing[14] or grid systems[6].

In the last years, different middleware for the OS have been developed. These middleware provide a higher level of abstractions in order to enable designers to build more complex software in an easier and feasible way. In this sense, not

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 588–599, 2012.

only the middleware has been created to extend the capabilities of the basic abstractions of the system, but also virtual machines have been implemented for managing specific entities of the system in an independent way, specially those that the OS cannot deal with. Multi-agent system platforms are a clear example of middleware which provide a powerful and high-level functionality that the OS is unable to give support to. These agent platforms are required for agent execution, but they imply a loss of efficiency, not only by including different layers between the agent paradigm and the OS, but also by lacking in support to several abstractions of the OS.

In this work, a new model of OS is presented. To face these problematic situations exposed before this work proposes the use of an agent-based methodology, structured in organizations and oriented to services by moving towards Operating Systems based on Agent Organizations. Therefore, it is possible to centralize the management of the OS and middleware. Using the agent technology to improve the OS design and solving these inconsistent and replication conflicts. In this proposal system applications are modeled as *agent organizations*, that form a modular system in which each agent can dynamically enter inside, collaborating to the achievement of the global goals of its organizations. Similarly, the OS itself is modeled as an agent organization that offers functionality to the other organizations that are executed in the system by means of *services*. Some of its offered services are: organization life-cycle management, access to system services, registration of new services, resource access, etc.

In Section 2 our position about integrating multi-agent system abstractions in the OS design is presented. In this section some traditional abstractions are mentioned as possible candidates to be improved or at least reconsidered, taking into account modern technologies and methodologies. Next, in Section 3, a description of the analysis of an Operating System based on Agent Organizations is detailed. In this analysis, the global goals of the system are defined, its functionality (services) and its stakeholders, employing the GORMAS methodology, which offers a set of meta-models for the analysis and design of an open multi-agent system, from an organizational point-of-view. In section 4, an abstract architecture of the OS is detailed. Finally, conclusions are presented in section 5.

2 Towards the Integration of OS and Agents

The agent model is a complex computational entity, usually driven by a set of goals and supported by a set of believes. In order to execute these computational models, a specific support for evaluating goals, managing believes and selecting the different tasks that lead to goal achievement is needed. This support can be included inside the OS, implementing the execution support based on goals (instead of other older programming paradigms) and improving the efficiency in the context switches, memory management and other parameters of the OS performance.

The *user* abstraction also presents a completely different perspective in a Multi-agent System or in an Operating System. The owner of an agent that is

running has no relationship with the user abstraction of an OS. As a consequence it is not clear which privileges should be conceded to the agent. This is due to the fact that the OS has not knowledge about the particular user management of the agent platform.

When trying to implement the security policies of agents, many conflicts inside the agent platform implementation and the OS have to be faced with. The agent platform is executed in the OS with user permissions. Hence, all the agents have the same permissions that the user and this is not necessary correct. Each agent could have a different OS owner. However, in some situations, the security policies that are developed in the middleware complement the OS ones.

Integrating concepts of the multi-agent system technology as new abstractions of the OS can provide many interesting advances in the research field. Agent features, like reactivity, proactivity, autonomy and sociability, will provide a great flexibility and dynamicity to the computational entities of the system. These features are not developed in the current abstractions of an OS, but they can be considered when including the agent technology as a computational entity of the system. In this way, creating autonomous computational entities allows the system to have applications that can achieve their goals on their own, without external interaction. This type of computational entity, unlike processes, is not a program counter that executes instructions in a sequential way, but a program that searches the best way for achieving its goals based on the tasks that it is capable of running. Moreover, as a social entity, it is able to communicate with other agents of its environment, demanding or providing services, in order to find the best solution to a problem.

Finally, employing the agent concept as a computational entity of the OS implies making profit of reactive and proactive entities. In this way, the computational entities of the system not only can execute code in a sequential way, but can also be aware of their environment and take proactive actions without external interaction.

With all these features in mind, we have envision an OS in which the computational entity is the agent, a high-level abstraction that gives great power to the applications development by means of the agent paradigm. Moreover, the multi-agent system technology focuses on groups of agents that collaborate to achieve a specific goal. In the last years, the concept of agent organizations has been highly employed, which implies a global goal, a defined topology, and a set of roles and norms that have to be accomplished by the members of the organization. These organizations, based on the Human Organization Theory [1], allow us to implement open systems where heterogeneous agents can be organized to achieve the organizational global goal.

An added value that can be obtained when modeling an OS based on agent organizations is the development of an open distributed system. Thereby, making use of the OS as a virtual organization manager and defining applications as agent organizations, user applications can be executed in a distributed way, selecting dynamically the best components that are more suitable to reach the goals of the organization.

With this executing model based on agent organizations it is possible to build applications that contract their calculation operations to external organizations, from a set of offered options. More specifically, the organization that represents the application is capable of selecting the best service among the offered options based on the service cost, its precision or other interesting parameters. Thus, the OS computational model is extended from the *cloud-computing* one, since the system is able to select, in an autonomous way, the elements to be included inside its organization to reach its goals.

3 Analysis of an Organizations-Based Operating System

The major obstacle of designing an OS based on agent organizations is that this OS is, concurrently, both an agent organization and a platform for other agent organizations. More specifically, the agent platform is completely integrated in the kernel of the OS, so it can provide all the advantages related in the previous section.

The analysis of the proposed OS is undertaken using a modern organizational-oriented Multi-Agent System (MAS) methodology called GORMAS (Guidelines for ORganizational-based Multi-Agent Systems)[2], which includes different meta-models and views to model systems based on the agent organizations technology.

In the analysis stage, a description of the motivation (mission) of the system is detailed. This includes defining the main goals of the system and the expected results, the services offered by the organization and its stakeholders (actors), and also the required roles that these stakeholder must adopt to provide or to employ the registered services of the organization.

Following, a brief description of the goals, services, actors and roles of the system and also their relationships are detailed.

3.1 Goals

The main goals pursued by the OS organization, which is also an agent platform that gives support to other organizations, include all those ones of a classical OS (i.e. protected hardware access, maximum resource usage) and also the goals of an agent platform based on organizations. The *Mission* of the proposed system is defined as the following set of goals:

1. Maximizing the utilization of all the system resources (hardware and software).
2. Protecting access to hardware resources.
3. Providing a simple layer of abstraction for hardware resources.
4. Providing an execution support and a life-cycle management of all organizations and agents that are executed in the system.
5. Giving support to service registration, search and invocation of open services.

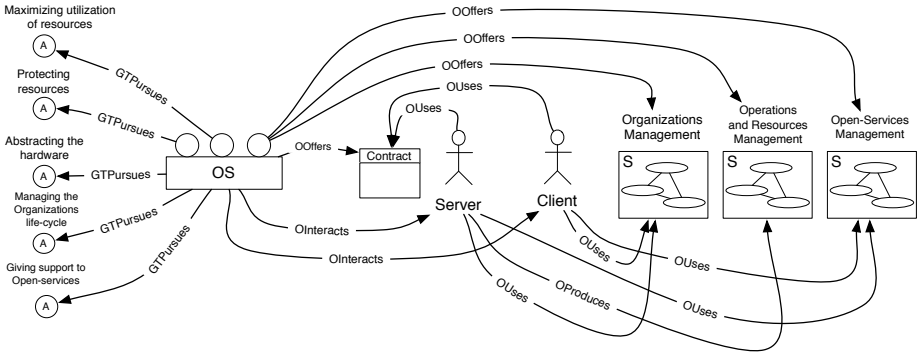


Fig. 1. Organization Model. Functional View. Mission

In order to achieve this set of global goals (Figure 1), the system must provide several services that represent the internal functionality of the system. These services are explained in the next subsection.

3.2 Services

As an open system, not only the services provided by the OS are offered and consumed, but also other computational entities (those that are allowed to participate inside the system according to the security policies) can register and provide their services to the rest of the system entities.

Under the paradigm we are working on, we have modeled a service-oriented system[11] based on the Service-Oriented Computing paradigm (SOC)[13]. In this system every organization consumes services in order to achieve its goals correctly or provides its services to the rest of the system organizations (or even to other systems, as long as it is an open and distributed system). We have called these open services *operations*.

System services are divided into three different types of meta-services:

Organizations Management: this type of services provides a basic functionality to create organizations in the system, to manage their life-cycle and topological structure, to manage their norms and rules and also to enable dynamic entry/exit of agents into organizations.

Open-Services Management: represents a set of basic services to register, deregister and search *operations* (open services). Thus, operations can be registered and employed by other computational entities that interact with the system.

Operations and Resources Management: includes a set of basic services to invoke operations and to access hardware resources, in a protected way.

These are the services offered by the OS to its agents or organizations. These meta-services consist of a set of services that agents are able to invoke in order to interact with the OS.

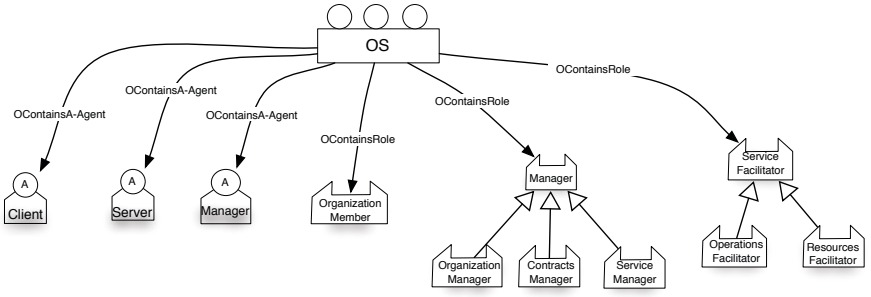


Fig. 2. Organization Model. Structural View.

3.3 Actors and Roles

The Agent Organizations-based Operating System is considered as an open service-oriented system, in which entities can provide its own implementation to registered services for operations and resources. There are two types of actors or stakeholders that interact with the system: *clients* and *servers*. In Figure 1 the relationships between these actors and the system are shown.

The *Client* actors interact with the OS to consume system services, both resources and operations. These actors are represented as abstract agents, that might be modeled as single agents or agent organizations in further steps of the analysis and design process. Their main goal is to achieve their own goals, making use of the resources and operations provided by the system.

The *Server* actors implement the specific functionality of the open services: resources and operations. Resource services include hardware access and other related functions, so they require some special privileges. *Operations* represent the software services that any agent might be capable to offer, without needing so many privileges from the OS.

To manage the security polices and privileges needed to perform these open services this OS defines a *role system*. Using this defined role system, system agents can dynamically adopt several roles, which establish all permitted and prohibited actions. An agent can adopt different roles inside the system simultaneously, such as being both a resources *Server* and an operations *Client* at the same time, while it is also a member of the system organization.

Following, the roles defined in the system are described. They are also depicted in Figure 2.

Organization Member: this is the most basic role that any agent adopts when entering inside the system. An agent belongs at least to one organization (for example, a plain organization in which it is the only member). When adopting this member role, he also acquires all privileges and obligations related to this role, such as permissions for invoking some specific management services of the organization (e.g. organization entry/exit, information about organizational goals, information about the services of the organization). It can also invoke system services.

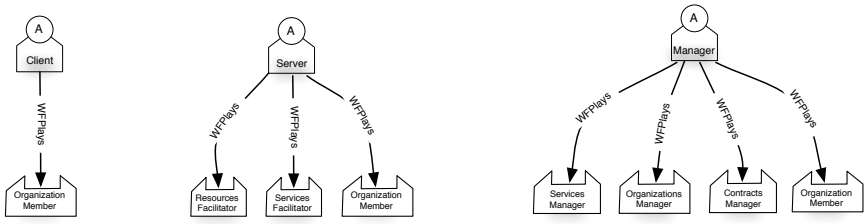


Fig. 3. Organization Model. Internal Functional View

Service Facilitator: this role is adopted by any agent that wants to provide a service inside the system. As explained before, open services are divided into *resources* and *operations*. As each type of service implies different privileges (i.e. it is not the same to provide a software service than a hardware device access), this role is specialized into: **Resources Facilitator** and **Operations Facilitator** for the management of both the resources and the operations.

Manager: this role is in charge of managing the different critical tasks of the system, so it can only be adopted by internal agents of the OS. It is specialized into:

Organization Manager: it controls the life-cycle process of the organizations of the system. It is in charge of providing the services related with the management of organizations, such as creation, access, information, etc., being a key element of the OS.

Service Manager: it provides all functionality related with service registration, deregistration and service searching. Thus, agents playing this role are in charge of providing the basic management of services for the other agents of the system, so they are enabled to publish and offer they own services and invoke those ones of other agents.

Contracts Manager: it controls the creation and management of *contracts*, which are an additional product of the system to improve the reliability and stability of the system. By using *contracts*, agents can negotiate some specific parameters and features needed for an interaction inside the system. For example, a *contract* is generated when an agent enters inside an organization. A contract is also generated before a service provision is done. It can even be required to consume a specific resource of the system. *Contracts* define which are the allowed interactions, which are the resources that can be accessed by the agent and they can also indicate time restrictions to achieve goals or to use services. Therefore, *contracts* define a way to control agent actions.

Figure 3 shows the Organizational Model (Internal Functional View), that connects system roles with the different abstract agents of the system: clients, servers and managers. Finally, Figure 4 shows the Organizational Model (External Functional View) that connects system roles with services provided by the OS. This Organizational Model is defined according to the GORMAS methodology.

As stated before, an external entity of the system can participate inside both as a client or a server, but not as a manager, which is implemented by internal agents of the system (similarly to the OS kernel). *Client* agents can only adopt the *Organization Member* role, which, as shown in Figure 4, is enabled to invoke service systems: Organization Management, Services Management and Resources Management. *Server* agents can adopt the *Organization Member* role, and also the *Resources Facilitator* or the *Services Facilitator* roles.

In order to achieve a global view of the modeled system, we can head to the External Functionality diagram of the Organization Model from the GORMAS methodology. In such diagram (see Figure 4) it is possible to identify the relationships between the services provided by the system (by means of the *OProvides* relationship) and the roles that agents take on in order to use or provide such services.

As can be seen in the related figure, the *Organization Member* role is the most basic role that an agent can assume. Its only purpose is to provide the ability to invoke the system self services: Organizations Management, Services Management and Resources Management.

Notice that this diagram does not show the *operations* service. This is because *operations* are features provided by the agents executing inside the system. It is not an OS goal to provide such *operations*. Its goal is to provide the support to register, search and use these *operations*, and this goal is pursued by the Services Manager role. The ability to provide both *resources* and *operations* is assumed by the Service Facilitator role. This role is specialized in the Resources Facilitator and the Operations Facilitator. Agents that assume the Resources Facilitator role are able to provide the *resource access service*, which is part of the OS. It is necessary to execute those services under some minimum protection parameters

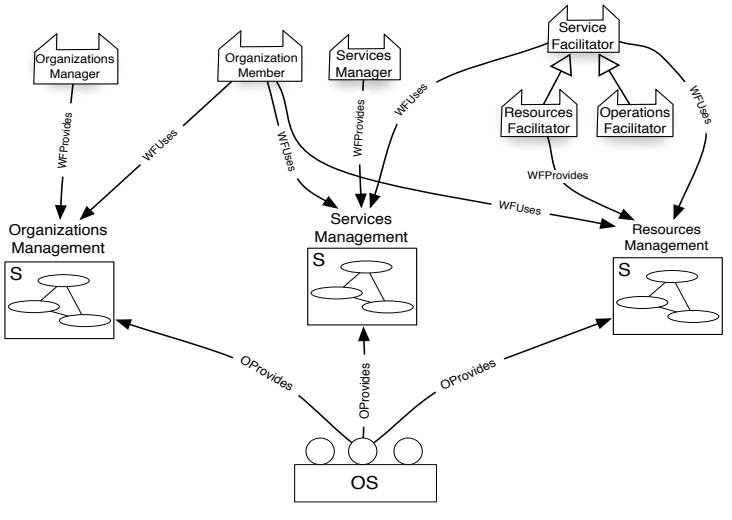


Fig. 4. Organization Model. External Functional View

in order to avoid wrong manipulations of the hardware. Those minimum resource access services can also be inherited by agents that assume the Resources Facilitator role (not only internal agents). Also notice that the Operations Facilitator role does not provide any initial service, since these services are open services (not part of the OS) and will be registered dynamically in the system.

4 Abstract Architecture of an Agent-Based Operating System

An initial abstract architecture for an Operating System based on Virtual Agent Organizations has been proposed (Figure 5). This type of architecture is challenging, mainly because we are building the system making use of a specific technology as design philosophy (i.e. agent technology), but also the main goal of the system is to provide this technology to its users. In short, we are building a multi-agent system platform that is also a multi-agent system at the same time. Moreover, this platform is directly working on the system hardware, so it must have extra goals that a classical agent platform does not need to take into account, such as resource protection, users management or platform execution of its own agents. In the same way, it must consider the specific tasks of an agent platform, but with a more strict way, such as agents' life-cycle management, agent organizations and communication layer.

In this work, system efficiency is not under scope, since our main goal is to propose an architecture based on an advanced technology that enables building very dynamic, autonomous and social systems with great expressivity. However, a small kernel or microkernel[5,7] that provides the minimal functionality required is needed, in order to build our OS on top of this kernel and the hardware resources. These microkernels have been highly developed in the recent years in the OS field.

As it is shown in Figure 5, all components of the OSs are built on top of this microkernel. These components offer the basic services of the system, i.e. Organizations Management, Services Management and Service Facilitator meta-services. It should be noticed that the Service Facilitator offers two kind of services: operations and hardware resources. As resources have direct access to hardware (controlled by the Service Facilitator component), they do not always need to make use of the microkernel.

It is necessary to provide these agent-related services from the OS layer. This way the OS can manage correctly the execution of the running agents in the system. It can control the high level abstractions that an agent owns from the management layer. This abstractions could be: the mailboxes where agents receive messages, the set of believes and desires (in a BDI[15] environment) and the services offered by agents. By upgrading these abstractions to the OS management layer, where processes are scheduled and interruptions are managed, we can make a more efficient, secure and reliable use of agents in the system as long as they are treated as computational entities.

Finally, applications executed in the system include the virtual agent organization abstraction in a goal-driven, service-oriented system. Each application

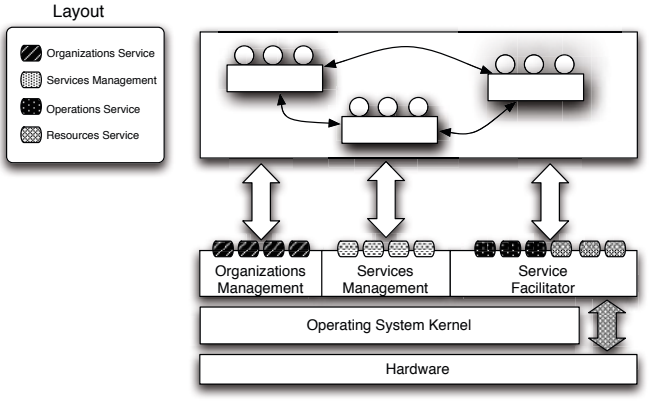


Fig. 5. Abstract Architecture of the OS

is represented by an agent organization that pursues a global goal or mission. Agents belonging to the organization must help on achieving this mission and can be punished or rewarded by the Manager agents.

As a remark, applications and classical dynamic libraries do not present any difference in this system, since it is a service-oriented system and any application can register a new service inside it, in a controlled way, and consume any service with a previous negotiation with this service provider and the contract manager. As stated before, the security of the system is controlled by the internal agents playing the Manager role. These agents are in charge of controlling the role-enactment process of the other agents, monitoring the contract establishment and its accomplishment and applying the specific sanctions or rewards according to the organizational norms.

5 Conclusions

This work presents the application of an Organizational Multi-Agent System methodology to the modeling of a complex piece of software: an Operating System. The objective of this work is to use the technology that has been developed to create intelligent systems during the last years as a solution for OS design. This technology has proved that it is enough robust and flexible to create complex systems where the communication between entities and the deliberation about their environment is present.

By providing more advanced abstractions from the OS we enable the possibility of creating complex applications using the support provided by the OS itself, without the necessity of middleware that introduces overload in the system. Thereby, the OS provides a set of powerful abstractions that are managed *directly* by the OS itself, making a more efficient and reliable system, while the built applications are also high-leveled thanks to the abstractions provided by the OS core.

As a result, some higher level abstractions for the OS have been identified (like agent organizations, roles or contracts). The objective of these abstractions is to offer a better functionality from the base. All this without the imposed limitations by a middleware created over the OS. These limitations are mainly system security policies, resources management and user management.

The elements identified in this work provide an initial study of the abstractions that can be provided by an OS. These elements introduce a more powerful technology that improves the system security and reliability. By introducing agreement technologies in the communication between OS entities, we improve the security provided by the system. OS entities (agents, services,...) are able to negotiate the interactions that are going to occur inside the system. It is possible to introduce *contracts*, which are managed by the OS, that control the rights and obligations that a computational entity must apply.

The aim of this work is to analyze the foundational elements of an OS from the point of view of an agent-based methodology. As *future work* we are designing an *agent model* based on the conclusions of this work, where we have identified the OS as a service-oriented system with complex computational entities (more advanced that a current *process abstraction*). This kind of service-oriented system will provide us an inherent distributed system where computational entities will be able to interact and to produce/consume services by means of the abstractions of the OS.

An appropriate base where the functionality proposed in this work could be introduced is the design of modern Operating Systems (like Microsoft Singularity[9,8]). These architectures are designed with technologies that were not usual in previous OS designs (like the use of an intermediate language, the virtual machine embedded in the kernel or the use of contracts and manifests for the security policies) and they are a suitable framework to develop the proposal of this OS architecture in the future.

Acknowledgments. This work is supported by TIN2008-04446 and TIN2009-13839-C03-01 projects of the Spanish government, PROMETEO/2008/051 project, FEDER funds and CONSOLIDER-INGENIO 2010 under grant CSD2007-00022.

References

1. Argente, E., Julian, V., Botti, V.: From Human to Agent Organizations. In: First International Workshop on Coordination and Organization (CoOrg 2005), pp. 1–11 (2005)
2. Argente, E., Julian, V., Botti, V.: Multi-agent system development based on organizations. Electronic Notes in Theoretical Computer Science 150(3), 55–71 (2006)
3. Covrigaru, A., Lindsay, R.: Deterministic autonomous systems. AI Magazine 12(3), 110–117 (1991)
4. David, F., Carlyle, J., Chan, E., Reames, P.: Improving dependability by revisiting operating system design. In: Proceedings of the 3rd Workshop on Hot Topics in System Dependability (HotDep 2007), p. 1 (January 2007)

5. Escoi, F.D.M., Bernabéu-Aubán, J.M.: The NanOS Object Oriented Microkernel: An Overview. Technical Report ITI-ITE-98/3 (March 2007)
6. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: Grid services for distributed system integration. Computer (January 2002)
7. Heiser, G., Elphinstone, K., Kuz, I., Klein, G., et al.: Towards trustworthy computing systems: Taking microkernels to the next level (2007), ertos.nicta.com.au
8. Hunt, G.C., Larus, J.R.: Singularity: rethinking the software stack. Techreport (January 2007)
9. Hunt, G.C., Larus, J.R., Tarditi, D., Wobber, T.: Broad New OS Research: Challenges and Opportunities. In: Proceedings of the 10th Workshop on Hot Topics in Operating Systems (January 2005)
10. Luck, M., McBurney, P.: Computing as interaction: Agent and agreement technologies (2008), csc.liv.ac.uk
11. Milanovic, N., Malek, M.: Service-Oriented Operating System: A Key Element in Improving Service Availability. In: Malek, M., Reitenspieß, M., van Moorsel, A. (eds.) ISAS 2007. LNCS, vol. 4526, pp. 31–42. Springer, Heidelberg (2007)
12. Milojicic, D., Kalogeraki, V., Lukose, R., Nagaraja, K.: Peer-to-peer computing. Tech. Rep. HPL-2002-57, HP Laboratories, Palo Alto (January 2002)
13. Papazoglou, M., Georgakopoulos, D.: Service-Oriented Computing. Communications of the ACM 46(10), 25–28 (2003)
14. Ramakrishnan, R.: Cloud computing was thomas watson right after all? In: IEEE 24th International Conference on Data Engineering (January 2008)
15. Rao, A., Georgeff, M.: BDI agents: From theory to practice. In: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS 1995), pp. 312–319 (January 1995)

An Empirical Comparison of Some Approximate Methods for Graph Coloring

Israel Rebollo-Ruiz and Manuel Graña

Computational Intelligence Group - University of the Basque Country
beca98@gmail.com, ccpgrrom@si.ehu.es

Abstract. The Graph Coloring Problem (GCP) is a classical NP-complete problem for which several approximate solution algorithms have been proposed: Brelaz algorithm, simulated annealing (SA), ant colony optimization (ACO). This paper reports empirical results on the GCP over a collection of graphs of some approximate solution algorithms. Among them, we test a recently proposed Gravitational Swarm Intelligence (GSI). Results in this benchmarking experiment show that GSI performance compares well to other methods.

Keywords: Graph Coloring, Gravitational Swarm.

1 Introduction

The Graph Coloring Problem (GCP) is a classical combinatorial optimization problem which is of NP-complete complexity [10,14,15,20,21]. The GCP consists in assigning a color to the nodes of a graph with the restriction that any pair of nodes that are linked can't have the same color. The chromatic number K is the minimum number of colors needed to color the graph. Classical algorithms to solve GCP are deterministic search algorithms [8,7,2]. Heuristics and random search allow to obtain approximations to the optimal solutions in bounded time. Recent approaches have applied Ant Colony Optimization (ACO) [11], Particle Swarm Optimization (PSO) [13], and Swarm Intelligence (SI) [27,31].

The bee hives [1], ant colonies [12] and flocking birds [9,29,30] are examples of swarms, whose global spatial configuration and dynamics can be interpreted as working in a cooperative way towards solving a problem. In SI models, the emergent collective behavior is the outcome of a process of self-organization, where the agents evolve autonomously following a set of internal rules for its motion and interaction with the environment and the other agents. Intelligent complex behavior appears from simple individual behaviors. An important feature of SI is that there is no leader agent or central control.

Diverse methods have diverse representations of the problem. ACO approaches make a correspondence between colors traveling over the graph and ants. The space for the motion of the agents is the topology defined by the graph, without any physical correspondence. In PSO, agents contain a full solution of the problem and exploration is made by generating perturbations around known solutions. In SI graph nodes correspond to agents traveling in a space towards a color coded goal.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 600–609, 2012.

The remaining of the paper is organized as follows: section 2 reviews the methods used in the comparison. Section 3 describes the generation of test graph instances. Section 4 give experimental results showing the accuracy finding the solution, computational cost measured in algorithmic iteration steps and time in seconds . Finally, section 5 gives some conclusions and our lines for future work.

2 Graph Coloring Problem Methods

We have implemented 5 GCP solving methods as described in the literature: Backtracking, DSATUR, Tabu Search, Simulated Annealing and Ant Colony Optimization. These methods have been proved individually to solve the GCP, but we have not find a direct comparison between all of them. We have developed a new algorithm called Gravitational Swarm Intelligence [26] that is included in this comparison, after proving that our algorithm works with the GCP. This algorithm used new methods of optimization in the artificial intelligence field [6,5]. A brief description of each algorithm follows:

1. Backtracking is a greedy but exhaustive algorithm that explores all the search space and always return the optimal solution if it exists. As the GCP is a NP-complete problem we can use backtracking only in small size problems or especial graphs like the mycielsky graphs [22]. This algorithm is deterministic, so always return the same solution for the same graph instance. Backtracking is no useful with medium size or big graphs, because it needs a huge computational time.
2. DSATUR (Degree of Saturation): this algorithm developed by Brlaz [2] is a greedy backtracking algorithm but does not explore exhaustively all the search space. It looks for the biggest clique in the graph and fix the initial number of colors needed to color it. Then starts the search to determine the color of the remaining nodes of the graph. The clique of a graph [3] is a subset of its vertexes such that every two vertexes in the subset are connected by an edge. It will be necessary at least the same number of colors k as the clique degree to color the graph, that is the reason of the algorithm's name "degree of saturation".
3. Tabu Search (TS): it is a random local search with some memory of the previous steps, so the best solution is always retained while exploring the environment [24]. TS needs a great amount of memory to keep the solutions visited, and if the tabu list is big, it will need so much time to search in the tabu list indeed. A full solution of a big problem can imply a lot of data to keep so could be a limitation in the GCP.
4. Simulated Annealing [28]: inspired in the annealing performed in metallurgy, this probabilistic algorithm finds solutions randomly. If a solution is worse than the previous solution it can nevertheless be accepted as the new solution with a certain probability that decreases with a global parameter called temperature. At the beginning the temperature is big and almost all the solutions are accepted, but when the temperature cools down, only the best

solutions are selected. This process allows the algorithm avoid local maximum. This algorithm has a big handicap when applied to solve the GCP, because there are a lot of neighboring states that have the same energy value. Despite this handicap, Simulated Annealing algorithm provides state-of-the-art results for this problem[23].

5. Ant Colony Optimization (ACO): we have build an implementation following [11] where we have $n * n$ ants making clusters around the colors. We have n ants in each of the n vertexes. Each ant is labeled with a randomly selected color, and the color of a vertex is equal to the color of the maximum size group of ants of the same color in this vertex. In each step, the ants that have a different color of the vertexes color moves through the edges to the neighbors. With the exiting ants and the new coming ants, the color of each vertex is again evaluated until the problem is solved.
6. Gravitational Swarm Intelligence (GSI): this algorithm is inspired in the Gravitation physic law of Newton, and the Boids swarm of Reynolds [29]. The gravitation law has been previously used in Swarm Intelligence [25], different from the GSI formulated for GCP in [27] which does not try to mimic exactly a physical system obeying Newton's law. An intuitive description of the algorithm follows. The GSI for GCP consists in a group of agents representing the vertexes moving in a world where the colors are represented as goal locations that exert an attraction to the agents. When an agent arrives at a goal, it can get that goal color and stop moving if there are no other agents than can't have the same color for the GCP definition, called enemies. The flowchart of figure 1 shows the internal logic works of each GSI agent . Initially a random position is selected for each agent. Depending on the position of the agent and the color goals, it moves toward the nearest goal until reaches a position inside the circle around the color goal defined a given radius.This circle is the region of the space where the agents stay still after getting a color. If two agents can't have the same color, we call them enemies. If there are enemies in that goal, the agent try to expel the enemies outside the goal to a random position. An enemy can be expelled if its internal parameter $Confort = 0$. The $Confort$ on an agent inside a goal grows with time. If the $Confort$ of the enemies is greater than zero then the enemy $Confort$ decreases one unit and the agent is expelled to a random position and start again. Otherwise the agent holds the goal color position and stops moving. If all the agents are stopped then the algorithm has solve the problem.

3 Instances of the Problem

We have implemented a graph generator to have our own graph families with specific features, that will help to tune the algorithms. Using Kuratowski's theorem [17] [18] we have create five families of planar graphs, increasing the number of nodes and vertexes regularly, stating with 50 vertexes and 100 edges and finishing with 250 vertexes and 500 edges. The planar graphs upper bound for

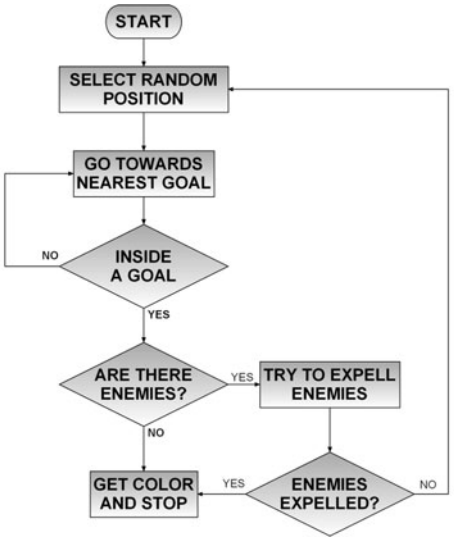


Fig. 1. SI Agent behavior flowchart for GCP

Table 1. Experimental graph tested features

Graph name	#nodes	#Edges	Density	K
kuratowski 50x100 (10)	50	100	0.5	4
kuratowski 100x200 (10)	100	200	0.5	4
kuratowski 150x300 (10)	150	300	0.5	4
kuratowski 1200x400 (10)	200	400	0.5	4
kuratowski 250x500 (10)	250	500	0.5	4

the chromatic number is 4 [19]. Kuratowski's theorem is useful to built regular planar graphs, but we have the limitation that we only know the upped bound of the chromatic number, but no the chromatic number.

For validation, it's a good idea to use well-known benchmarking graphs, whose chromatic number is known. For graphs whose chromatic number is unknown the algorithm validation comes from the comparison to other graph coloring algorithms [32,4]. We have test our algorithm in previous works [27] with instances of Mycielsky graphs [22], and the DIMACS graphs [15,16], but in this paper we go further with more explicit problems, and bigger families. In Table 1 we show the features of the graphs used in our test

We have a total of 50 graph grouped in 5 families. The deterministic algorithms Backtracking and DSATUR has been tested once for each graph (because are deterministic) and letting them 10^6 steps. The non deterministic algorithms have been tested 30 times for each graph, and letting them 5.000 steps before stopping them, except for the Simulated Annealing algorithm that is faster and lest complex so we let it 50.000 steps.

4 Experimental results

We have made experiments with randomly generated graphs generated. We have implemented all the algorithms using Visual Basic .Net, thus building a GCP suite that will be made public for independent validation of our claims. The graph generator is also been included in this suite to allow other researchers to generate theirs own graphs and solve then with one of the sixth methods. The implementation can be found in http://www.ehu.es/ccwintco/uploads/b/be/Swarm.rar.

We have implemented all these algorithm because we desire to perform comparison of the GCP solving methods on new graph instances, instead of using result published in the literature, and also because is difficult to find a working implementation of this algorithms. The programing language, the computer used or even the structures used in the implementation can made a big difference between different works. All the experiments have been run in the same computer.

4.1 GSI Implementation

Even though, our algorithm is about SI agents moving around the search space, we haven't use any parallel implementation, even though we claim that our algorithm is scalable. At each time step all the SI agents motion is evaluated. After each time step, the cost function must be evaluated to see if the problem is solved or not. We have two time reference units, the standard hours, minutes and seconds to compare with other algorithms and the iteration steps to compare experiments over the same graph. The real computing time can change from one computer to another, but the steps will be always the same. When we are evaluating the next position of a SI agent in the step t, we take into account the position of the other SI agents in the step $t-1$. All the implementations have been made using the same developing language and trying to use the most homogeneous data structures.

We have arbitrarily defined a 100 x 100 toric world. The goal radius and Comfort parameters have been adjust in order to have better results.. The goal radius has been set to 30 points and the goals have been deployed equidistant in a imaginary circle. The comfort has been set to 4. With this value the algorithm is dynamic enough to get good results. These parameter have been selected empirically. The speed is normalized between [0,1] so is no need to change the agents speed. We have seen that if the goal radius is small and the number of agents is big the convergence is slow, as we expected. The same happens with the comfort. We have demonstrate empirically that our algorithm is scalable because have almost the same result for the five families. We haven't have exactly the same results for round errors.

4.2 Results

In table 2, 3, 4 and 5 we show a cloud of points corresponding the percentage of success for each method in each graph. Each table there are results of ten graph

Table 2. Graph coloring results for the kuratowski graph instances of size 50×100

50x100	BT	DSATUR	SA	TABU	ACO	GSI
K1	100	100	100	80	73	97
K2	100	100	100	90	77	97
K3	100	0	100	47	47	100
K4	100	100	100	80	77	93
K5	100	100	100	60	70	100
K6	100	100	100	87	70	97
K7	0	0	100	67	73	93
K8	0	100	100	50	60	100
K9	100	100	100	43	57	100
K10	100	100	100	77	73	100

Table 3. Graph coloring results for the kuratowski graph instances of size 100×200

100x200	BT	DSATUR	SA	TABU	ACO	GSI
K1	100	0	100	47	73	97
K2	0	0	100	43	80	97
K3	100	100	100	40	33	97
K4	100	100	100	47	43	90
K5	100	100	100	40	70	100
K6	0	100	100	3	63	97
K7	0	0	100	30	37	90
K8	0	0	100	43	57	100
K9	100	100	100	17	57	97
K10	0	0	100	20	80	93

instances with the same number of vertexes and edges. The deterministic BT and DSATUR algorithms with only 1 execution have a success of 100% or 0 %. The Backtracking and DSATUR algorithm achieved the same result for almost all the instances.

We can see that GSI algorithm is not the best one in some cases, but is always between the best ones. And when the problem becomes more difficult is the only method that still obtains good results. The GSI algorithm has a high level of scalability and for this reason the size of the problem doesn't affect too much to the results. The SA algorithm is the best for small instances but it's accuracy decreases with the size of the problem. The ACO like our GSI algorithm should had a stable behavior, but get poor results, and the size of the problem affects it behavior. Other problem with ACO is the computation time requirements, when the graph size grows the time consumed increases very fast, making this ACO implementation useless for big graphs. Finally, the TS gets very bad results in almost all the situations and it also very expensive in computation time.

Figure 2 shows the average evolution time in steps of the experiment for each graph family and method. We have normalized the number of steps to

Table 4. Graph coloring results for the kuratowski graph instances of size 150 × 300

150x300	BT	DSATUR	SA	TABU	ACO	GSI
K1	0	0	90	7	23	87
K2	0	0	57	7	30	80
K3	100	100	100	37	57	93
K4	0	0	83	3	30	90
K5	0	0	100	13	30	93
K6	0	0	100	17	23	93
K7	0	0	93	7	23	90
K8	0	0	100	0	57	87
K9	0	0	100	0	30	97
K10	0	0	100	10	57	93

Table 5. Graph coloring results for the kuratowski graph instances of size 200 × 400

200x400	BT	DSATUR	SA	TABU	ACO	GSI
K1	0	0	20	3	30	97
K2	0	0	37	3	17	90
K3	0	0	23	0	0	93
K4	0	0	100	3	7	83
K5	0	0	87	0	23	93
K6	0	0	27	0	30	83
K7	0	0	100	0	17	90
K8	100	100	77	3	0	87
K9	0	0	57	3	7	87
K10	0	0	100	0	7	93

5.000 and show them graphically. We can see that SA is the fasted in small instances, but GSI is the fastest with a big difference when the problem start to grow. The Backtracking, DSatur, Tabu Search and ACO algorithms needs about the same number of steps for each graph family. The steps need to solve the problem increases very fast when the problem went difficult. It appears from this experiment that GSI provides good approximate solutions in linear time.

Even though we always are speaking about steps instead of time in seconds,the ACO method is too slow. Among the other five, SA and GSI are the fastest. In figure 3 we can see the evolution of average time in seconds of the experiments grouped by graph instance size. For a clear vision of the results, we have set a maximum value of the ordinate axis of 40 seconds, saturating the plot when the algorithm time goes over this number. The ACO needed more than 600 seconds for the two biggest families. The GSI needs little time, but more than the SA. This is because, even though our algorithm is scalable, the implementation haven't take into account this important feature that would make the GSI the fastest algorithm.

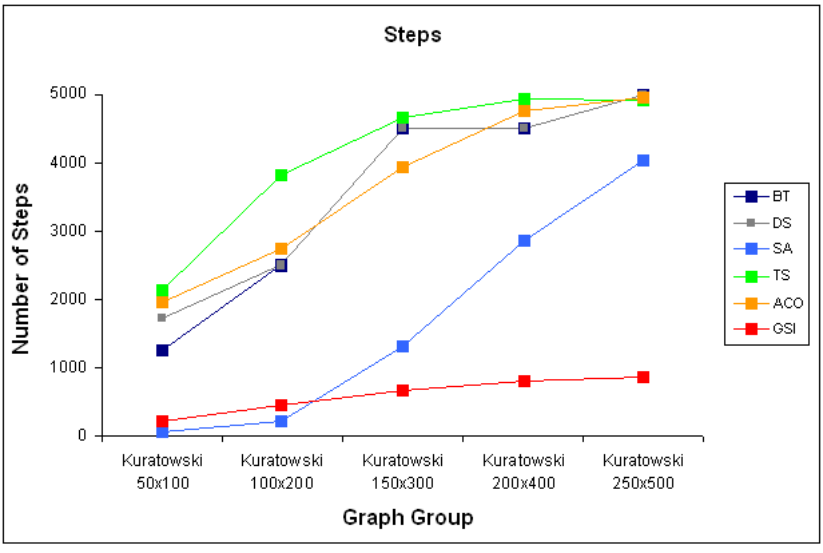


Fig. 2. Average time evolution between families in steps

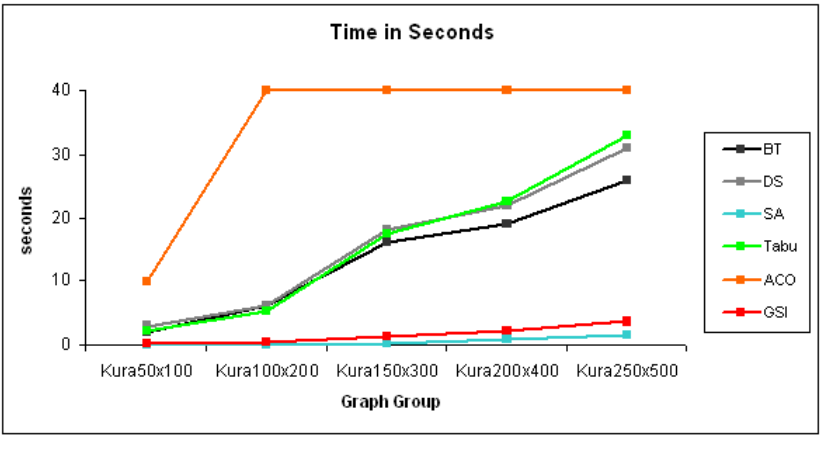


Fig. 3. Average time evolution between families in seconds

5 Conclusions

We have build a GCP suite for testing six different GCP solution methods. We have added to this suite two graph generators, one for regular planar graphs and other for hard 3-color-able graphs. We have tested all the methods with groups of graphs of increasing size.

We have seen that the stochastic Simulated Annealing is the fastest and the most successful method for small graphs. When the size of the graphs grows, SA starts to have problems to find a solution, but is remains the fastest method. If we haven't been comparing graph with a strict restriction of time and stooped, we will have achieved better results. The GSI approach is among the best methods for small graph and is the best for big graphs, because of its scalability features. The ACO algorithm is very slow and has obtained quite poor results. The Tabu Search it the worst algorithm for this problem and the most time consuming.

Future work will be directed to test the approaches on the Mizuno's graphs [21] and also a bigger group of graph families to continue comparing. We also want to seek for other methods of coloring to add to our suite.

References

1. Akay, B., Karaboga, D.: A modified artificial bee colony algorithm for real-parameter optimization. Information Sciences (2010) (in press, corrected proof)
2. Brelaz, D.: New methods to color the vertices of a graph. Commun. ACM 22, 251–256 (1979)
3. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Commun. ACM 16, 575–577 (1973)
4. Chvatal, V.: Coloring the queen graphs, Web repository (2004) (last visited July 2005)
5. Carvalho, A., Corchado, E., Abraham, A.: Hybrid intelligent algorithms and applications. Information Sciences, 2633–2634 (2010)
6. Wozniak, M., Corchado, E., Graña, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75, 61–63 (2012)
7. Corneil, D.G., Graham, B.: An algorithm for determining the chromatic number of a graph. SIAM J. Comput. 2(4), 311–318 (1973)
8. Dutton, R.D., Brigham, R.C.: A new graph colouring algorithm. The Computer Journal 24(1), 85–86 (1981)
9. Folino, G., Forestiero, A., Spezzano, G.: An adaptive flocking algorithm for performing approximate clustering. Information Sciences 179(18), 3059–3078 (2009)
10. Galinier, P., Hertz, A.: A survey of local search methods for graph coloring. Comput. Oper. Res. 33(9), 2547–2562 (2006)
11. Ge, F., Wei, Z., Tian, Y., Huang, Z.: Chaotic ant swarm for graph coloring. In: 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), vol. 1, pp. 512–516 (2010)
12. Handl, J., Meyer, B.: Ant-based and swarm-based clustering. Swarm Intelligence 1, 95–113 (2007)
13. Hsu, L., Horng, S., Fan, P.: Mtpso algorithm for solving planar graph coloring problem. Expert. Syst. Appl. 38, 5525–5531 (2011)
14. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning. Operations Research 39(3), 378–406 (1991)
15. Johnson, D.S., Trick, M.A.: Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, vol. 26. American Mathematical Society (1993)
16. Johnson, D.S., Trick, M.A.: Proceedings of the 2nd DIMACS Implementation Challenge. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26. American Mathematical Society (1996)

17. Kuratowski, K.: Sur le problème des courbes gauches en topologie. Fund. Math. 15, 271–283 (1930)
18. Kuratowski, K.: A half century of polish mathematics: Remembrances and reflections. Pergamon Press, Oxford (1980)
19. Luzar, B., Skrekovski, R., Tancer, M.: Injective colorings of planar graphs with few colors. Discrete Mathematics 309(18), 5636–5649 (2009)
20. Mehrotra, A., Trick, M.: A column generation approach for graph coloring. INFORMS Journal On Computing 8(4), 344–354 (1996)
21. Mizuno, K., Nishihara, S.: Constructive generation of very hard 3-colorability instances. Discrete Appl. Math. 156(2), 218–229 (2008)
22. Mycielski, J.: Sur le coloureage des graphes. Colloquium Mathematicum 3, 161–162 (1955)
23. Nolte, A., Schrader, R.: Simulated annealing and graph colouring. Comb. Probab. Comput. 10, 29–40 (2001)
24. Porumbel, D.C., Hao, J., Kuntz, P.: A search space cartography for guiding graph coloring heuristics. Computers & Operations Research 37(4), 769–778 (2010)
25. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: Gsa: A gravitational search algorithm. Information Sciences 179(13), 2232–2248 (2009)
26. Rebollo, I., Graña, M.: Further results of gravitational swarm intelligence for graph coloring. In: Nature and Biologically Inspired Computing (2011)
27. Ruiz, I.R., Romay, M.G.: Gravitational Swarm Approach for Graph Coloring. In: Pelta, D.A., Krasnogor, N., Dumitrescu, D., Chira, C., Lung, R. (eds.) NICSO 2011. SCI, vol. 387, pp. 159–168. Springer, Heidelberg (2011)
28. Rebollo, I., Grana, M., Hernandez, C.: Aplicacion de algoritmos estocosticos de optimizacion al problema de la disposicion de objetos no-convexos. Revista Investigacion Operacional 22(2), 184–191 (2001)
29. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. In: Computer Graphics, pp. 25–34 (1987)
30. Reynolds, C.W.: Steering behaviors for autonomous characters (1999)
31. Sundar, S., Singh, A.: A swarm intelligence approach to the quadratic minimum spanning tree problem. Information Sciences 180(17), 3182–3191 (2010)
32. Turner, J.S.: Almost all k-colorable graphs are easy to color. Journal of Algorithms 9(1), 63–82 (1988)

A Predictive Evolutionary Algorithm for Dynamic Constrained Inverse Kinematics Problems

Patryk Filipiak[1], Krzysztof Michalak[2], and Piotr Lipinski[1,⋆]

[1] Institute of Computer Science,
University of Wroclaw, Wroclaw, Poland
lipinski@ii.uni.wroc.pl
[2] Institute of Business Informatics
Wroclaw University of Economics, Wroclaw, Poland

Abstract. This paper presents an evolutionary approach to the Inverse Kinematics problem. The Inverse Kinematics problem concerns finding the placement of a manipulator that satisfies certain conditions. In this paper apart from reaching the target point the manipulator is required to avoid a number of obstacles. The problem which we tackle is dynamic: the obstacles and the target point may be moving which necessitates the continuous update of the solution. The evolutionary algorithm used for this task is a modification of the Infeasibility Driven Evolutionary Algorithm (IDEA) augmented with a prediction mechanism based on the ARIMA model.

Keywords: evolutionary algorithm, dynamic function optimization, dynamic environment, inverse kinematics, time series prediction.

1 Introduction

The recent developments in robotics, cybernetics or even computer graphics have led to a growing interest in the Inverse Kinematics problem, which concerns finding values of parameters describing some object of skeletal structure (e.g. a robot or virtual model of human body) which allow an intended pose of this object to be achieved. The Inverse Kinematics approach is usually employed when it is more important to find an acceptable placement of some manipulator (e.g. robotic arm) rather than provide an actual control for it. Thus, this kind of problem arise mainly in the area of visualization (computer graphics, animation, etc.) and as a sub-problem called *arm configuration selection* in the wider context of the reaching subtask in manipulation planning problems [3,18].

One of the simplest formulations of the Inverse Kinematics problem is as follows: find such a pose of a manipulator that, given a starting position, a desired position of its ending is achieved. This simple problem formulation may be modified in several ways. The manipulator can have a more complicated geometry than a simple chain of linear segments. For example a tree–like structure can be used to represent a hand with the movement of each finger simulated individually. It is also possible, that segments of the manipulator are described by a different set of parameters, for example segments may bend or stretch. Another modification is the introduction of obstacles. Shape of the obstacles may vary from simple blocks to sophisticated labyrinths.

⋆ Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 610–621, 2012.

Although the classic Inverse Kinematics problem can be solved by some classic analytical methods, more advanced version of the problem, e.g. with additional constraints or obstacles, require more efficient algorithms. There are a number of evolutionary algorithms applied to various versions of the classic Inverse Kinematics problem.

In this paper we use the evolutionary approach in order to solve an Inverse Kinematics problem with an articulated arm and dynamic obstacles and target point. A modification of the Infeasibility Driven Evolutionary Algorithm (IDEA) [15,16] seems to be particularly useful for this problem, because many solutions which collide with obstacles while infeasible in physical world may constitute useful approximations of a feasible solution. Because of the assumption that obstacles and the goal are dynamic we employ a prediction mechanism based on the ARIMA model [4] to improve algorithm performance.

This paper is structured as follows: in Section 2 a formal problem definition is given. Section 3 presents some classical approaches to various Inverse Kinematics problems. Section 4 introduces a detailed description of the proposed evolutionary algorithm with a prediction mechanism. In section 5 the experimental setup is described and the results of experiments are presented. Section 6 concludes the paper.

2 Problem Definition

There is a broad range of Inverse Kinematics problems with their own slightly different formal definitions.

In this paper we focus on a problem with an articulated manipulator (like robotic arm) in the shape of a chain of any number M of fixed–length linear segments. The problem is to find angles at all joints of such a manipulator given a starting position and a desired position of its ending. The manipulator has one fixed starting point S and its shape is determined by a set of line segment lengths $l_1, \ldots, l_M$. The objective is to reach a target point O. Given the fixed coordinates of the starting point S the position of the manipulator can be described by a set of numbers $\alpha_1, \ldots, \alpha_M$ which represent angles at which manipulator segments are positioned. There are two possible interpretations of these numbers: as absolute or as relative angles. In this paper we treat the numbers as relative angles, so the actual angle of $k-th$ manipulator segment is $\sum_0^k \alpha_k$, where α_0 is an arbitrarily chosen initial angle (we chose $\alpha_0 = \pi/2$ - the direction straight along Y axis). In the environment in which the manipulator operates a number of obstacles are introduced.

The problem that we approach in this paper is dynamic which means that the obstacles move and the target point may also change its location. The motion of the obstacles may be linear or circular with varying values of velocity and period.

3 Classical Approach

There exists a broad range of solutions that address many variants of the Inverse Kinematics problem upon which the Cyclic Coordinate Descent (CCD) [7] and the Jacobian–based methods [2,19] are considered nearly canonical. Performance of CCD, though, is

limited by the fact that it optimizes each joint separately. What is more, neither of the two methods allows for incorporating constraints.

Some numerical methods were introduced in [9,10] however they tend to fall into local optima in more complex tasks. Universal and rapid solutions that can address various static constrained Inverse Kinematics problems using Genetic Algorithms and Niching Techniques were also presented in [12,13,17].

One ubiquitous disadvantage of many evolutionary methods lies in their low reactivity to changes in dynamic environment. On the other hand, analytical methods can typically be applied only to special cases of Inverse Kinematics and strictly depend on precise analytical model that is usually difficult to construct in real–life problems.

Many recent developments in artificial intelligence systems reveal that applying hybrid techniques gives promising results in difficult computational problems [5,6,8] by employing various additional tools, e.g. learning machines [1], neural networks [14], and/or statistical methods (as it is proposed in this paper).

4 Evolutionary Algorithm with Prediction

The algorithm used in this paper is based on the Infeasibility Driven Evolutionary Algorithm (IDEA) [15,16] that is well–fitted to some dynamic problems. IDEA increases the reactivity to changes by maintaining in population a fraction of infeasible individuals with the highest fitness value. Let P_{Size} be the size of a population and $0 < \alpha \ll 1$ be the rate of infeasible solutions which means there are $\lfloor \alpha P_{Size} \rfloor$ infeasible individuals kept in population.

An evaluation function and constraint functions (i.e. functions counting the number of crossings with obstacles) are tested for changes at each t-th time step where $t = 2, \ldots, N_G$. If any such change is detected, a whole population P_{t-1} is re-evaluated.

Evolutionary operators are enclosed in the Sub-evolve function which comprises of $N_{G'}$ iterations of crossover and mutation. Being invoked with an argument P_{t-1} as an initial population Sub-evolve function returns the resulting population C_{t-1} of sub-evolution unioned with P_{t-1} then reduced to the fractions of $\lceil (1-\alpha) P_{Size} \rceil$ feasible and $\lfloor \alpha P_{Size} \rfloor$ infeasible individuals both with the highest fitness value.

Algorithm 1 presents the evolutionary framework using IDEA for dynamic optimization problems proposed in [16].

4.1 Prediction Mechanism

The modification suggested in this paper integrates original IDEA with a prediction model in the following manner. In each iteration $t \geq H_{Start}$ an additional hidden population H_{t+1} of the size P_{Size} is initialized randomly then evaluated, sub-evolved and reduced as described above. However, instead of proper values of an evaluation function at the current time step t predictions of corresponding values at $t+1$ are used. Later on, at the beginning of the next iteration, an additional hidden population H_t is re-evaluated according to the actual evaluation function that is known at the moment. After that, the population P_{t-1} is unioned with H_t and reduced back to P_{Size} individuals in the same manner as described above, i.e. including $\lfloor \alpha P_{Size} \rfloor$ best infeasible solutions.

Algorithm 1. Evolutionary framework for Inverse Kinematics problem where P_{Size} = population size, N_{Gen} = number of generations

$P_1 = \text{InitPopulation}()$
Evaluate(P_1)
for $t = 2 \rightarrow N_{Gen}$ **do**
  if the function has changed **then**
    Evaluate(P_{t-1})
  end if
  $C_{t-1} = \text{Sub-evolve}(P_{t-1})$
  Evaluate(C_{t-1})
  $P_t = \text{Reduce}(P_{t-1} + C_{t-1})$
end for

As a prediction model ARIMA(p, d, q) [4] is used in order to forecast changes in the location of obstacles and/or target point because this is the factor that implicitly influence the values of evalution function. Previous positions of such dynamic objects of environment are collected in time series. Their new current coordinates are stored each time a change of location is detected. As a consequence, a predicted state of environment for time step $t + 1$ can be estimated at time step t then used for the evaluation of a hidden population.

Algorithm 2. Evolutionary framework for Inverse Kinematics problem with a prediction model where P_{Size} = population size, N_{Gen} = number of generations and H_{Start} is the number of iteration at which the prediction begins.

$P_1 = \text{InitPopulation}()$
Evaluate(P_1)
for $t = 2 \rightarrow N_{Gen}$ **do**
  if the function has changed **then**
    Evaluate(P_{t-1})
    if $t - 1 \geq H_{Start}$ **then**
      Evaluate(H_t)
      $P_{t-1} = \text{Reduce}(P_{t-1} + H_t)$
    end if
  end if
  $C_{t-1} = \text{Sub-evolve}(P_{t-1})$
  Evaluate(C_{t-1})
  $P_t = \text{Reduce}(P_{t-1} + C_{t-1})$
  if $t \geq H_{Start}$ **then**
    $H_t = \text{InitPopulation}()$
    PredictionEvaluate(H_t)
    $D_t = \text{PredictionSub-evolve}(H_t)$
    $H_{t+1} = \text{Reduce}(H_t + D_t)$
  end if
end for

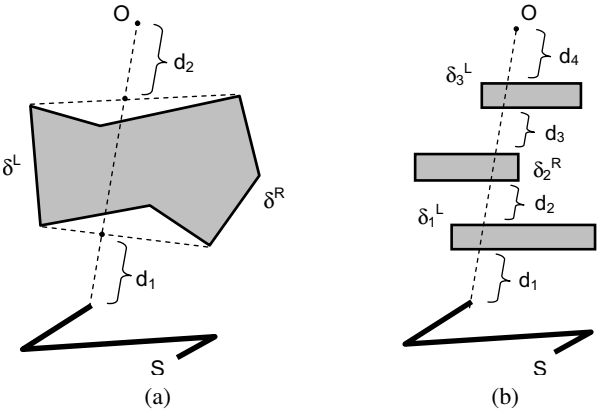


Fig. 1. Evaluation of an individual in the case of (a) one obstacle ahead, (b) three obstacles ahead

Algorithm 2 presents the framework of the described process in a form of pseudocode. For simplicity, the proposed algorithm with ARIMA prediction model will be referred to as IDEA-ARIMA for the remainder of this paper.

4.2 Evaluation Function

Each individual in populations P_t or H_t is represented by the following sequence of angles

$$(\alpha_1, \ldots, \alpha_M) \in [A_1, B_1] \times [A_2, B_2] \times \ldots \times [A_M, B_M], \tag{1}$$

where $0 \leq A_i < B_i \leq 2\pi$ determines the lower and the upper bounds (respectively) of the angle range of i-th manipulator's segment for $i = 1, \ldots, M$.

The evaluation of an individual is equal to Euclidean distance between the end of the last (i.e. M-th) segment of a manipulator and the target point O providing there are no obstacles crossed by the line between them.

If there is one obstacle crossed by this line, the convex hull of the obstacle is computed. In this case, two ways to pass by the obstacle emerge, i.e. by following the clockwise (δ^R) or the counterclockwise direction (δ^L). As a result, the shorter path of these two is selected. In the example depicted in Figure 1(a) the distance between the end–effector and the target point is split into two segments of the lengths d_1, d_2 and two possible paths of the lengths δ^L, δ^R. The evaluation equals $d_1 + \delta^L + d_2$, because clearly $\delta^L < \delta^R$.

In the case when more than one obstacle is crossed by the mentioned line, all obstacles are sorted according to distances between them and the end of the last segment of a manipulator. Then, consecutive obstacles are treated separately as described above. Figure 1(b) presents such case. Here, the distance is split into segments of the lengths d_1, d_2, d_3, d_4 and paths of the lengths $\delta_1^L, \delta_1^R, \delta_2^L, \delta_2^R, \delta_3^L, \delta_3^R$ hence the evaluation is $d_1 + \delta_1^L + d_2 + \delta_2^R + d_3 + \delta_3^L + d_4$. Note that the evaluation function defined in this manner rewards passing by the obstacles rather than greedily searching for the shortest Euclidean distance to the target as it would completely ignore the presence of obstacles.

Individuals coding manipulators that cross an obstacle in at least one point are considered infeasible. A number of such crossings defines the order among infeasible solutions, i.e. the greater number of crossings, the more infeasible the solution.

5 Experiments

The proposed algorithm was tested on three types of benchmark problems described below. In each problem, the aim was to find a pose of manipulator that minimizes the distance (measured in the manner that was presented in Section 4) between its last segment and the target point $O \in \mathbb{R}^2$ without crossing any obstacles. For simplicity, all feasible ranges of angles α_i $(i = 1, \ldots, M)$ were set to $[0, 2\pi]$.

Obstacles and/or target point were able to change their locations in time. For any given point $(x, y) \in \mathbb{R}^2$ of such dynamic elements the position of it at time step t, namely (x_t, y_t), was calculated using the following predefined motion function

$$\mathbb{R}^2 \times \{1, \ldots, \tau\} \ni t \longmapsto \phi(x, y, t) = (x_t, y_t). \tag{2}$$

Crossover and mutation probability were set to 0.9 and 0.1 respectively in both IDEA and IDEA-ARIMA. The fraction of infeasible yet promising individuals formed 20% of the population at all time steps.

Parameters of ARIMA prediction model were set to (1, 1, 1) as it resulted in optimal forecasting accuracy for examined problems. First H_{Start} = 16 iterations of any run of IDEA-ARIMA were dedicated only for collecting information about locations of obstacles for the sake of further predictions. These initial iterations were not taken into account in either of statistics presented in this section since observable results of both algorithms do not differ within this time period.

5.1 Benchmark I

Shape and alignment of obstacles including the way their locations change in time is depicted in Figure 3. Point $S \in \mathbb{R}^2$ represents the fixed location of a manipulator. The distance d between S and the target point O was set to 4. The manipulator itself was built out of $M = 4$ joints of the length $l = 2$. The test procedure lasted for $\tau = 16$ iterations (excluding H_{Start}) during which the three obstacles (marked as light-gray rectangles in Figure 3) where moving from left to right and vice versa in one of the two following manners:

(a) *uniform* – with the motion function $\phi(x, y, t) = (x + \omega(t), y)$ where $(x, y) \in \mathbb{R}^2$ and $\omega(t)$ defined as follows

$$\omega(t) = \begin{cases} -4\eta t/\tau & t \in [0, \tau/4) \cap \mathbb{Z}, \\ 4\eta t/\tau - 2\eta & t \in [\tau/4, 3\tau/4) \cap \mathbb{Z}, \\ -4\eta t/\tau + 4\eta & t \in [3\tau/4, \tau] \cap \mathbb{Z} \end{cases} \tag{3}$$

for a given amplitude $\eta > 0$. It is worth mentioning that such $\omega(.)$ is a discretization of the function depicted in Figure 2.

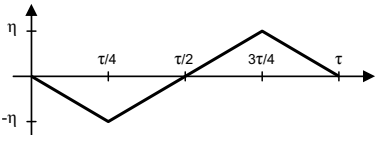


Fig. 2. Uniform motion function used in Benchmark I

(b) *sinusoidal* – with the motion function defined as

$$\phi(x, y, t) = \left(x + \eta \sin\left(\frac{2\pi t}{\tau}\right), y \right), \tag{4}$$

where $(x, y) \in \mathbb{R}^2,\ t = 1, \ldots, \tau,\ \eta > 0$.

Table 1 presents the summary of results for 10 runs of IDEA and IDEA-ARIMA with population sizes (*pop*) 20, 50, 100 and 10, 20 sub-EA steps (*sub*) and motion amplitude $\eta = 2.5$. For each combination of *pop* and *sub* the three following values are presented:

1. *mean* – mean value of best feasible solutions found by given algorithm at each time step $t = 1, \ldots, 10 \cdot \tau$,
2. *wins* – percent of time steps t when the best feasible solution found by given algorithm was better than the corresponding one found by the other algorithm,
3. $\Delta < 0.5$ – percent of time steps t when the best feasible solution found by given algorithm was worse than the corresponding one found by the other algorithm but not worse than 0.5 (i.e. solution that is still satisfying one though not the best found).

Note that IDEA *wins* and IDEA-ARIMA *wins* do not always sum up to 100% (especially in cases where population size was relatively low). It is because none of the algorithms was able to find a feasible solution in some iterations.

As it is clearly seen, IDEA-ARIMA outperforms IDEA in all presented cases. The Student's t-test performed on each combination of population size and sub-EA steps used within the experiment revealed the superiority of IDEA-ARIMA at the significance level $\alpha < 0.01$. Additionally, values in column $\Delta < 0.5$ (fraction of still satisfying but not the best solutions found) show that the number of remaining time steps when IDEA-ARIMA performs rather poorly is relatively small.

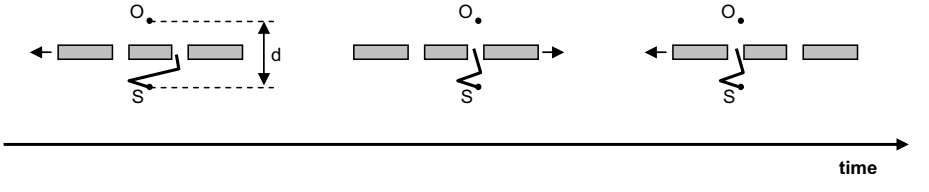


Fig. 3. Illustration of the movement of obstacles in Benchmark I. On each frame point S represents the location of a manipulator while O is the target point. The distance d between these points was set to 4.

Table 1. Summary of results for Benchmark I after 10 runs of IDEA and IDEA-ARIMA with population sizes 20, 50, 100 and 10, 20 sub-EA steps

uniform motion function

		IDEA			IDEA-ARIMA		
pop	sub	mean	wins	$\Delta < 0.5$	mean	wins	$\Delta < 0.5$
20	10	7.0954	28%	13%	1.4139	71%	17%
50	10	0.7419	29%	23%	0.3128	71%	21%
100	10	0.4841	29%	51%	0.1745	71%	24%
20	20	1.6240	38%	30%	0.4937	61%	23%
50	20	1.1384	30%	44%	0.1729	68%	23%
100	20	0.2537	24%	60%	0.0727	76%	21%

sinusoidal motion function

		IDEA			IDEA-ARIMA		
pop	sub	mean	wins	$\Delta < 0.5$	mean	wins	$\Delta < 0.5$
20	10	1.6738	34%	13%	0.5580	64%	12%
50	10	1.1252	31%	27%	0.1586	66%	18%
100	10	0.9452	30%	32%	0.0565	70%	26%
20	20	1.4262	34%	25%	0.3199	65%	16%
50	20	1.0245	28%	40%	0.0649	73%	20%
100	20	1.3396	15%	51%	0.0156	85%	14%

Table 2. Summary of results for Benchmark II after 10 runs of IDEA and IDEA-ARIMA with population sizes 20, 50, 100 and 10, 20 sub-EA steps

		IDEA			IDEA-ARIMA		
pop	sub	mean	wins	$\Delta < 0.5$	mean	wins	$\Delta < 0.5$
20	10	3.1932	23%	26%	2.2538	77%	12%
50	10	2.9021	23%	20%	1.3467	77%	21%
100	10	2.9346	17%	16%	1.1059	83%	10%
20	20	2.4268	49%	21%	2.3896	51%	21%
50	20	2.1339	34%	34%	1.6165	66%	20%
100	20	2.1130	23%	25%	0.7715	77%	18%

It is visible in Table 1 (as it was expected) that effectiveness of both evolutionary algorithms usually grows with the increase of population size and number of iterations. On the other hand, the dynamic character of the presented benchmark problem reveals that there are time steps when the target point is quite easy to reach by the manipulator and other ones when obstacles are extremely hard to avoid. This is why it was impossible to cross the certain level of *wins* despite increasing parameters of algorithm.

5.2 Benchmark II

Figure 4 depicts the dynamics of Benchmark II. Two vertices of obstacle change their locations by moving outwards for the first $\tau/2$ time steps and inwards for another $\tau/2$ time steps as the arrows indicate.

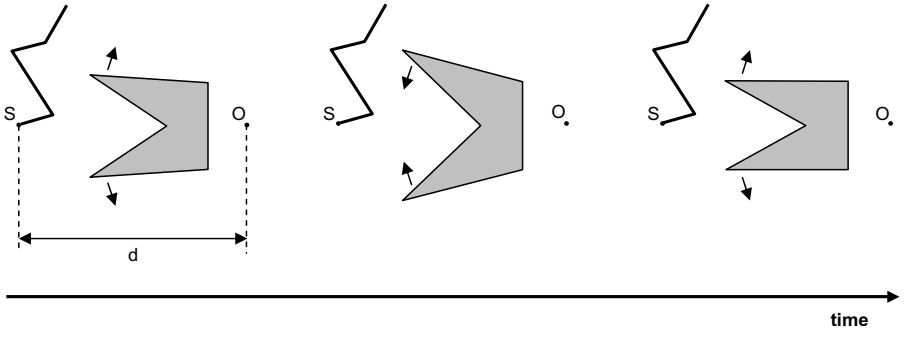


Fig. 4. Illustration of the movement of obstacles in Benchmark II. On each frame point S represents the location of a manipulator while O is the target point. The distance d between these points was set to 9.

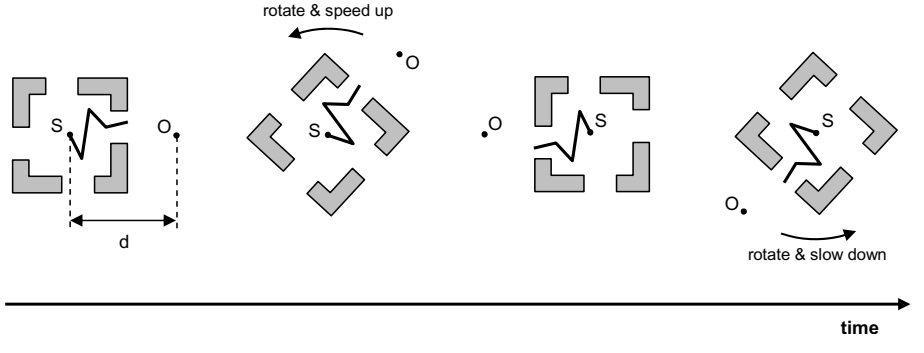


Fig. 5. Illustration of sigmoid-circular movement of obstacles in Benchmark III. On each frame point S represents the location of a manipulator while O is the target point. The distance d between these points was set to 5.

The parameters were set as follows: number of joints $M = 7$, length of joints $l = 2$, number of time steps $\tau = 16$, distance to target point $d = 9$ and sinusoidal motion amplitude $\eta = 1$.

The use of evaluation function described in Section 4 plays an important role in this benchmark problem because the obstacle is a concave figure. The shape of the obstacle is designed to have a local optimum very near point O in the terms of Euclidean distance.

Table 2 summarizes the results of 10 runs of both algorithms. IDEA-ARIMA clearly outperforms IDEA. The Student's t-test revealed the statistical significance (at the level $\alpha < 0.01$) in all cases except for $pop = 20, sub = 20$. Apparently, it is caused by the higher level of difficulty of this problem.

5.3 Benchmark III

Unlike Benchmarks I and II where obstacles were moving and target point O was static, in this case both obstacles and target point are circulating around point $S = (x_S, y_S) \in \mathbb{R}^2$ where manipulator is located. Let for all $(x, y) \in \mathbb{R}^2$ and $t = 1, \ldots, \tau$ motion function be defined as $\phi(x, y, t) = (x_t, y_t)$ where

$$x_t = (x - x_S)\cos\beta(t) + (y - y_S)\sin\beta(t) + x_S, \tag{5}$$

$$y_t = (x - x_S)\sin\beta(t) + (y - y_S)\cos\beta(t) + y_S. \tag{6}$$

Two variants of $\beta(t)$ defined in time domain determine two motion functions considered in this benchmark problem:

(a) for *uniform-circular* motion function defined as

$$\beta(t) = \frac{2\pi t}{\tau}, \qquad t = 1, \ldots, \tau, \tag{7}$$

(b) for *sigmoid-circular* motion function defined as

$$\beta(t) = \frac{2\pi}{1 + \exp(-6 + \frac{12t}{\tau})}, \qquad t = 1, \ldots, \tau. \tag{8}$$

Figure 5 presents the environment (i.e. shape and alignment of obstacles, location of manipulator and target point) and illustrates sigmoid-circular motion of objects.

Table 3 summarizes the results of 10 runs of IDEA and IDEA-ARIMA with parameters: number of joints $M = 4$, length of joints $l = 2$, number of time steps $\tau = 32$, distance to target point $d = 5$. The use of IDEA gave satisfactory results in less then 60% of time steps. IDEA-ARIMA noted *wins* around twice as often than IDEA and produced satisfactory results in almost all time steps in this benchmark. The Student's t-test revealed the statistical significance (at the level $\alpha < 0.01$) in all examined cases.

6 Conclusions

In this paper an extension of the Infeasibility Driven Evolutionary Algorithm (IDEA) with a prediction mechanism based on ARIMA model is proposed for solving dynamic constrained Inverse Kinematics problem of reaching target point with M-segmented manipulator.

Experiments performed on a few dynamic benchmark problems revealed that IDEA-ARIMA outperforms IDEA in all examined cases by producing satisfactory results in almost all time steps, which proves that the prediction of the dynamic objective function may lead to better results. However, further studies are necessary to examine the relevance of prediction in more random environments.

This paper shows that ARIMA–based prediction delivers promising results when applied to some dynamic constrained Inverse Kinematics problems that can occur in real–life robotics applications.

Table 3. Summary of results for Benchmark III after 10 runs of IDEA and IDEA-ARIMA with population sizes 20, 50, 100 and 10, 20 sub-EA steps

uniform-circular motion function

		IDEA			IDEA-ARIMA		
pop	sub	mean	wins	$\Delta < 0.5$	mean	wins	$\Delta < 0.5$
20	10	1.9126	26%	9%	0.9376	67%	8%
50	10	1.0658	29%	11%	0.4296	68%	11%
100	10	0.6066	28%	20%	0.1836	71%	8%
20	20	1.0224	31%	31%	0.5842	67%	15%
50	20	0.6549	25%	28%	0.1201	69%	19%
100	20	0.8457	23%	48%	0.0591	71%	20%

sigmoid-circular motion function

		IDEA			IDEA-ARIMA		
pop	sub	mean	wins	$\Delta < 0.5$	mean	wins	$\Delta < 0.5$
20	10	1.6738	18%	18%	0.5580	67%	12%
50	10	1.1252	23%	30%	0.1586	73%	20%
100	10	0.9452	25%	36%	0.0565	70%	24%
20	20	1.4262	26%	22%	0.3199	57%	20%
50	20	1.0245	20%	37%	0.0649	71%	18%
100	20	1.3396	24%	32%	0.0156	70%	23%

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Balestrino, A., De Maria, G., Sciavicco, L.: Robust control of robotic manipulators. In: Proceedings of the 9th IFAC World Congress, vol. 5, pp. 2435–2440 (1984)
3. Bertram, D., Kuffner, J., Dillmann, R., Asfour, T.: An Integrated Approach to Inverse Kinematics and Path Planning for Redundant Manipulators. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1874–1879 (2006)
4. Box, G.E.P., Jenkins, G.M.: Time series analysis: Forecasting and control, revised edition. Holden-Day, San Francisco (1976)
5. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
6. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
7. Fêdor, M.: Application of inverse kinematics for skeleton manipulation in real-time. In: Proceedings of the 19th Spring Conference on Computer Graphics, pp. 203–212. ACM (2003)
8. Garca, S., Fernndez, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
9. Goldenberg, A.A., Benhabib, B., Fenton, G.: A complete generalized solution to the inverse kinematics of robots. IEEE Journal of Robotics and Automation RA-1(1) (1985)
10. Goldenberg, A.A., Lawrence, D.L.: A generalized solution to the inverse kinematics of robot manipulators. ASME Journal of Dynamic Systems, Measurement, and Control 107, 103–106 (1985)

11. Hatzakis, I., Wallace, D.: Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: Proceedings of the GECCO 2006, pp. 1201–1208. ACM (2006)
12. Karla, P., Mahapatra, P.B., Aggarwal, D.K.: On the solution of multimodal robot inverse kinematics function using real-coded genetic algorithms. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1840–1845 (2003)
13. Karla, P., Mahapatra, P.B., Aggarwal, D.K.: On the comparison of niching strategies for finding the solution of multimodal robot inverse kinematics. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, vol. 6, pp. 5356–5361 (2004)
14. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
15. Singh, H.K., Isaacs, A., Tapabrata, R.: Infeasibility Driven Evolutionary Algorithm (IDEA) for engineering design optimization. In: Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence, pp. 104–115 (2008)
16. Singh, H.K., Isaacs, A., Nguyen, T.T., Ray, T., Yao, X.: Performance of infeasibility driven evolutionary algorithm (IDEA) on constrained dynamic single objective optimization problems. In: Proceedings of IEEE Congress on Evolutionary Computation, CEC 2009, pp. 3127–3134 (2009)
17. Tabandeh, S., Clark, C., Melek, W.: A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering. In: Proceedings of IEEE Congress on Evolutionary Computation, CEC 2006, pp. 1815–1822 (2006)
18. Wilfong, G.: Motion planning in the presence of movable obstacles. In: Proceedings of ACM Symposium on Computational Geometry, pp. 279–288 (1988)
19. Wolovich, W.A., Elliot, H.: A computational technique for inverse kinematics. In: Proceedings of 23rd IEEE Conference on Decision and Control, pp. 1359–1363 (1984)
20. Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E.: Prediction-Based Population Reinitialization for Evolutionary Dynamic Multi-objective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 832–846. Springer, Heidelberg (2007)

Non-linear Data Stream Compression: Foundations and Theoretical Results

Alfredo Cuzzocrea and Hendrik Decker

[1] ICAR-CNR and University of Calabria, I-87036 Cosenza, Italy
[2] Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, E-46071 Valencia, Spain
cuzzocrea@si.deis.unical.it, hendrik@iti.upv.es

Abstract. In this paper, we provide foundations and theoretical results of a novel paradigm for supporting *data stream miming algorithms* effectively and efficiently, the so-called *non-linear data stream compression model*. Particularly, the proposed model falls in that class of data stream mining applications where interesting knowledge is extracted via *suitable collections of OLAP queries from data streams*, being latter ones *baseline operations* of complex knowledge discovery tasks over data streams implemented by ad-hoc data stream mining algorithms. Here, a fortunate line of research consists in *admitting approximate, i.e. compressed, representation models and query/mining results at the benefit of a more efficient and faster computation*. On top of this main assumption, the proposed non-linear data stream compression model pursues the idea of maintaining a *lower degree of approximation* (thus, as a consequence, a higher *query error*) for *aggregate information* on those data stream readings related to *interesting events*, and, by contrast, a *higher degree of approximation* (thus, as a consequence, a lower query error) for aggregate information on other data stream readings, i.e. readings not related to any particular event, or related to low-interesting events.

Keywords: Theoretical foundations of data stream processing; non-linear data stream compression.

1 Introduction

Data stream mining algorithms aim at discovery interesting knowledge from multi-rate rapidly-evolving streams by means of *clustering* [21], *classification* (e.g., [1]), *decision trees* (e.g., [15]), OLAM (e.g., [10]), and so forth. Contrary to traditional knowledge discovery methodologies on static data, mining algorithms on data streams cannot take advantages from multi-scan routines allowed to run on the *entire* data set, and, as a consequence, applying conventional mining algorithms over evolving data like streams results to be of limited efficacy and efficiency in real-life application scenarios. Therefore, it turns out more and more the crucial relevance of *efficiently representing and querying data streams*, being latter ones *baseline operations* of complex knowledge discovery tasks over data streams implemented by ad-hoc data stream mining algorithms.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 622–634, 2012.

A well-recognized solution to issues above consists in *admitting approximate representation models and query/mining results at the benefit of a more efficient and faster computation* [9]. This main intuition, which has been inherited from similar research experiences in the context of *approximate query answering on large databases and data cubes* [3], has caused the proliferation of a number of proposals focusing on *synopsis data structures* for data streams [9]. The idea beyond these research initiatives is quite simple, yet effective. Since the stream is potentially unbounded, devise methodologies (and related data structures) able to *summarize* the incoming stream within the actual time window in order to finally make the capabilities of advanced analysis/mining tools over data streams more effective and efficient. Obviously, retrieved analysis/mining results are approximate in nature, due to an unavoidable information loss. Despite this, it is clearly understood that for many data stream application scenarios (e.g., analysis of sensor network data [4], mining of frequent patterns over data streams [23], RFID processing tools [6] etc) precision is not a tight requirement, and *trend analysis* naturally suffices to the desired (analysis) goals. Among successful synopsis data structures for data streams proposals of the literature, we recall: *sampling* (e.g., [28,5]), *histograms* (e.g., [20,17]), *wavelets* [18], *sketches* (e.g., [16,14,26]), *quantile* (e.g., [19]) and *frequency moments* (e.g., [25]).

In line with this research, an innovative *approximate query answering technique* is proposed in [12]. This technique introduces a particular representation model for collected data stream readings in terms of *a list of* OLAP*-like compressed two-dimensional arrays*. According to this representation model, for each array one dimension is defined on the *stream domain* and the other dimension on the *temporal domain*, respectively. Each two-dimensional array is computed over a *fixed time window*, and stores aggregate information useful to efficiently answer *range-SUM queries* [22], a noticeable class of OLAP queries, over the stream-source/time range it refers. Particularly, not only the previous class of OLAP queries, also called *Window Queries* (WQ) [12], can be answered, but the so-called *Continuous Queries* (CQ) [2], which take as input a data stream and, fixed the desired range of interest, return as output a stream of SUM-based answers, can even be answered efficiently. Then, upon each array, a *quad-tree* [27] is hierarchically built by means of progressively aggregating summarized readings, in a bottom-up manner. This leads to the definition of the so-called *Quad-Tree Window* (QTW), whose data domain at the lowest aggregation level corresponds to the underlying two-dimensional array. Also, to gain efficiency during data management and query evaluation, the collection of QTW is indexed by means of a high-efficient B-tree indexing structure. The resulting whole data structure is called *Multi-Resolution Data stream Summary* ($MRDS$), and it is particularly suitable to answer range-SUM queries over compressed data streams in an OLAP-like multi-resolution manner at a provable query error [12]. Efficient algorithms for progressively compressing the $MRDS$ under a given storage space bound B as time passes and new readings arrive are presented in [12], along with models and algorithms for efficiently evaluating approximate range-SUM queries over (compressed) data streams.

The basic assumption of data stream compression algorithms presented in [12] relies on the idea of progressively compressing the data structure via erasing nodes of the "oldest" QTW in order to obtain free space to be exploited for representing incoming arrivals within nodes of "new" QTW. This is a quite reasonable compression strategy, as *for several data stream processing application scenarios the most recent information is more important than old information*. Overall, the compression model presented in [12] introduces what we call as the *linear data stream compression model*. [13] is a significant extension of [12] to the context of *Grid Computing*; here, algorithms of [12] are specialized to this computational environment in order to gain performance. Despite this, there exist several classes of data stream application scenarios for which this model fails. Specifically, the idea of removing old data in favor of new data is not suitable for all those application scenarios in which *sporadic interesting events, although expired, can still be relevant for knowledge discovery tasks over data streams*. Therefore, in [11] we assert that the linear compression model must be adequately improved via embedding the amenity of achieving the so-called *non-linear data stream compression model*. This model pursues the idea of maintaining a *lower degree of approximation* (thus, as a consequence, a higher *query error*) for aggregate information on those data stream readings related to interesting events, and, by contrast, a *higher degree of approximation* (thus, as a consequence, a lower query error) for aggregate information on other data stream readings, i.e. readings not related to any particular event, or related to low-interesting events. At least, if some events are particularly significant for the specific application scenario considered, one can also decide to maintain those portions of aggregate information related to the corresponding readings *uncompressed*. Overall, the above-described strategy defines an *adaptive data stream compression approach* [11].

While in [11] we proposed the general non-linear data stream compression framework, in this paper we propose theoretical foundations and results coming from the framework [11], also based on *hybrid artificial intelligent paradigms* (e.g., [7]). In particular, the general framework [11] and the results presented in this paper fail in the hybrid artificial intelligent research area as they combine the data stream compression paradigm with the event-processing paradigm.

2 Foundations of Non-linear Data Stream Compression

This Section focuses on the foundations of the non-linear data stream compression paradigm. Under the non-linear compression scheme, the $MRDS$ is compressed in dependence on a given degree of approximation δ and the storage space B' to be released in order to represent new arrivals , such that $B' < B$, B denoting the input storage space bound available to house the whole $MRDS$. With respect to the linear compression model, here the novelty relies in the fact that the degree of approximation δ must be now considered along with B' *simultaneously*. In more detail, the non-linear compression process aims at finally

achieving a compressed $MRDS$ such that *aggregate information expose a degree of approximation which is at least equal to δ, having released the required storage space B'.*

In order to retrieve the degree of approximation δ, readings are elaborated within semantic windows with the aim of adequately capturing the degree of interestingness of events and, in consequence of this, producing in output appropriate values of the parameter δ. *The way of defining and handling the relation between interesting events and δ strongly depends on the particular data stream application scenario*, and hence it must be determined and characterized accordingly during the start-up phase of the target data stream application/system by the system administrator on the basis of his/her knowledge about the specific (data stream) application domain. As a consequence, defining and handling the relation between interesting events and δ plays a *critical role* for modern data stream applications and systems, and must be considered carefully. This critical task is related to the definition of semantic windows [24], and, as a consequence, it should be accomplished separately in a transparent-for-the-data-stream-processing-layer (implemented by the $MRDS$) manner. For this reason, δ must be considered as a *free parameter* , and must be contextualized to the particular (data stream application) instance. It should be noted that, with respect to the specific issue of handling the parameter δ, this amenity introduces a meaningful *separation abstraction* between the data stream processing layer (i.e., $MRDS$) and the external event processing layer.

Events occurring in the target data stream are continuously collected and recorded, by also automatically *annotating* their degree of interestingness. As a consequence, the whole temporal dimension of the $MRDS$ is annotated by means of tuples of the following kind:

$$\varrho_k = \langle E_k, t_{E_k,start}, t_{E_k,end}, \delta_k \rangle \tag{1}$$

such that: (i) E_k denotes the actual event; (ii) $t_{E_k,start}$ denotes the starting timestamp in which the event E_k occurred; (iii) $t_{E_k,end}$ denotes the ending timestamp in which the event E_k expired; (iv) δ_k is the degree of approximation required for the compression of aggregate values related to readings E_k is associated to. Furthermore, without any loss of generality, we denote as $\langle [S_0 : S_{N-1}], [t_{E_k,start} : t_{E_k,end}] \rangle$ the portion of aggregate values related to readings E_k is associated to, and as QTW_{E_k} the QTW built on top of $\langle [S_0 : S_{N-1}], [t_{E_k,start} : t_{E_k,end}] \rangle$.

δ_k can be reasonably expressed as a percentage value, whose semantics is as follows:

- $\delta_k = 100\%$: this value of δ means that the portion of aggregate values $\langle [S_0 : S_{N-1}], [t_{E_k,start} : t_{E_k,end}] \rangle$ must be maintained uncompressed (in other words, E_k is an event of particular relevance for the actual data stream application scenario) – this value of δ does not originate any compression in QTW_{E_k};

- $\delta_k = 0\%$: this value of δ means that the portion of aggregate values $\langle [S_0 : S_{N-1}], [t_{E_k,start} : t_{E_k,end}] \rangle$ is not critical for the analysis goals of the specific knowledge-discovery/mining tasks over the target data stream considered, and hence it can be completely removed by saving the whole aggregate value it represents in a singleton node (in other words, E_k is an event of no relevance for the actual data stream application scenario) – this value of δ originates a *full compression* of QTW_{E_k};
- $\delta_k \in]0, 100[\%$: this value of δ means that the portion of aggregate values $\langle [S_0 : S_{N-1}], [t_{E_k,start} : t_{E_k,end}] \rangle$ have to be compressed depending on δ_k (in other words, E_k is an event of intermediate relevance for the actual data stream application scenario) – this value of δ originates a *partial compression* of QTW_{E_k}.

The scheme above and its semantics finally determine an event-based compression of the $MRDS$, i.e. a compression that is driven by the degree of interestingness and the relevance of events occurring in the target data stream. As highlighted in Section 2, recall that compression tasks above (triggered by different δ values) involve QTW_{E_k} *locally*, but they must be performed under the *global* goal of accomplishing the main requirement of the compression process, i.e. releasing the storage space B' needed to represent new arrivals. Just to give an insight, given a portion of aggregate values $\langle [S_0 : S_{N-1}], [t_{E_k,start} : t_{E_k,end}] \rangle$ for which the required degree of approximation is $\delta_k = 0\%$, it could be the case that the deriving compression task still does not cause a full compression of QTW_{E_k}, as the desired storage space B' could be released *before* the limit condition represented by $\delta_k = 0\%$ is reached on QTW_{E_k}, due to (partial) amounts of storage space released by the compression of *other* portions of aggregate values of the $MRDS$ different from $\langle [S_0 : S_{N-1}], [t_{E_k,start} : t_{E_k,end}] \rangle$. This finally originates in QTW_{E_k} a degree of approximation $\delta_k > 0\%$ that is still compliant with the main requirement of the compression process , i.e. releasing the storage space B'. Overall, the paradigm above delineates a *best-effort strategy*.

3 Formal Models and Theoretical Results

In this Section, we focus the attention on specific properties and theoretical results of non-linear data stream compression. Properties and theoretical results discussed here derive from studying approximate query answering techniques over (event-based) compressed data streams in great detail. This stage will constitute the conceptual basis of the non-linear QTW compression process [11].

From [12], note that, for an uncompressed QTW, $\delta_k = 100\%$, whereas for a full-compressed QTW, $\delta_k = 0\%$. In the first case (i.e., $\delta_k = 100\%$), retrieved answers to range-SUM queries issued against such a QTW expose the higher degree of approximation supported by the *current* representation of the $MRDS$ (and, by the contrary, the lower query error due to the approximate evaluation of range-SUM queries). In the second case (i.e., $\delta_k = 0\%$), the QTW is reduced to its root node solely, and retrieved answers to range-SUM queries issued against

such a QTW expose the lower degree of approximation supported by the *current* representation of the $MRDS$ (and, by the contrary, the higher query error due to the approximate evaluation of range-SUM queries). Recall that the query error is due to the application of classical linear interpolation techniques to the two-dimensional ranges of stream sources and time corresponding to QTW nodes involved by input range-SUM queries (see [12]).

A critical question related to the non-linear compression of QTW (and, in turn, to the non-linear compression of the $MRDS$) is: *How to model the relation between the degree of approximation δ of the event-based data stream compression model and the effective compression of a QTW?*

In order to answer this question, consider that, given a range-SUM query Q against a QTW, *the accuracy of the approximate answer to Q, $\widetilde{A}(Q)$, is mostly due to the contribution of leaf nodes of the QTW* [12]. In addition to this, *the less is the aggregation level of summarized stream readings represented by leaf nodes the more is the accuracy of the approximate answer* (see Section 1). This claim is obviously valid for arbitrary queries, which, in the most general case, overlap the two-dimensional ranges associated to (involved) nodes of the QTW (internal as well as leaf nodes – from [12], recall that if a query Q is perfectly corresponding to the two-dimensional range of a singleton node or a collection of nodes of the QTW, then the answer to Q, $A(Q)$, is exact). In this case (i.e., overlapping arbitrary queries), it is completely obvious that QTW leaf nodes contribute to the approximate answer $\widetilde{A}(Q)$ by means of *exact partial results* (those retrieved from the ranges of stream sources and time they refer to) hence they do not introduce query error and, instead, tend to make $\widetilde{A}(Q)$ accurate, by keeping a *direct proportionality* with the depth of the QTW level they belong to (and, by the contrary, an *inverse proportionality* with the aggregation level of summarized stream readings they represent). It should be noted that the property above is valid under the common assumption stating that, in hierarchical/multi-resolution approximate query answering synopsis data structures (like histograms), the selectivity of input OLAP queries is higher than the granularity of the lowest aggregation level represented in the structure [8], which is a proper artifact of synopsis data structures meant to enhance the quality and the accuracy of approximate answers as more as possible.

To become convinced of the property illustrated above, consider the following running example. Figure 1 (a) shows a 13×26 $(\Delta t = 2)$ two-dimensional range of stream sources and time, whereas Figure 1 (b) shows the data-oriented representation of a QTW built on top of such a range. In addition to this, Figure 2 shows the logical representation of the QTW of Figure 1 (b).

Now, consider Figure 3, where two range-SUM queries, namely Q_0 and Q_1, against the two-dimensional range of stream sources and time (Figure 3 (a)) and the QTW (Figure 3 (b)) of Figure 1, respectively, are depicted.

From Figure 3 and Figure 2, note that Q_1 overlaps QTW leaf nodes representing summarized stream readings at an aggregation level that is lower than the aggregation level of summarized stream readings represented by QTW leaf nodes overlapped by Q_0. Also, note that the answer to Q_0 evaluated against

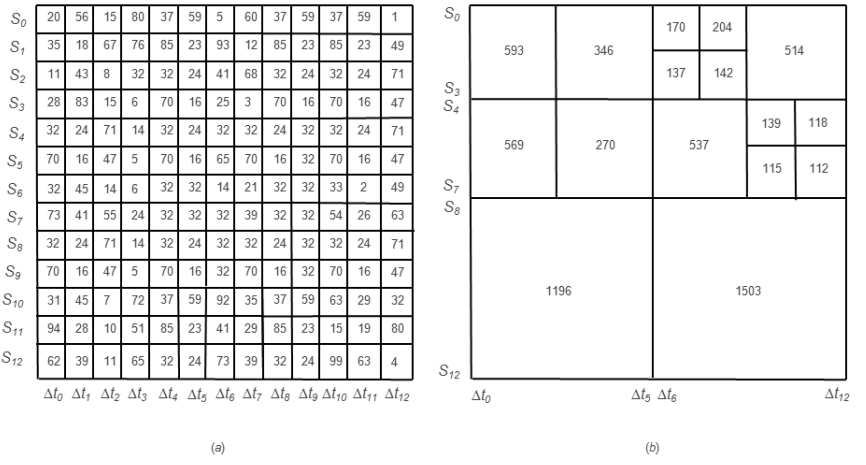


Fig. 1. A range-SUM query example over a 13×26 ($\Delta t = 2$) two-dimensional range of stream sources and time (a) and the data-oriented representation of a QTW built on top of such a range (b)

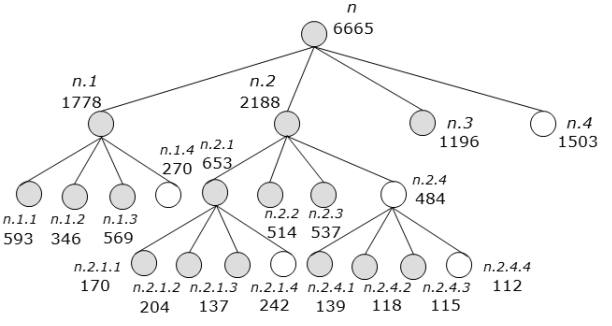


Fig. 2. Logical representation of the QTW of Figure 1 (b)

the range of Figure 1 (a) is exact (i.e., $A(Q) = 1054$), as Q_0 is perfectly corresponding to the grid defined on the range, whereas the answer to Q_1 evaluated against the same range is approximate (i.e., $\widetilde{A}(Q_1) = 655$), as Q_1 overlaps the grid defined on the range (see [12]). When Q_0 and Q_1 are evaluated the QTW of Figure 1 (b) (which, in turn, is built on the range of Figure 1 (a)), the following approximate values are retrieved: $\widetilde{A}(Q_0) = 822.12$ and $\widetilde{A}(Q_1) = 550.20$. In our formal framework, given a range-SUM query Q against a two-dimensional range of stream sources and time $\mathcal{A}$, we introduce the *Percentage Relative Error* (PRE) $\varepsilon(Q)$, defined as follows ([12]):

$$\varepsilon(Q) = \frac{|A(Q) - \widetilde{A}(Q)|}{A(Q)} \tag{2}$$

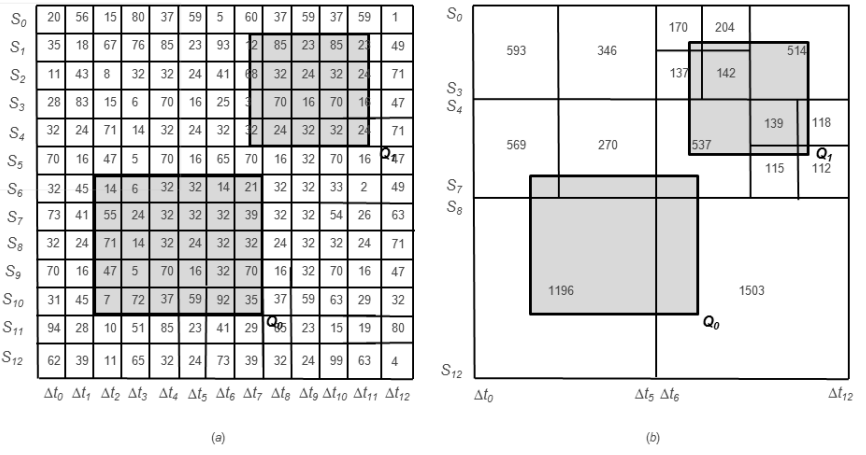


Fig. 3. Range-SUM queries Q_0 and Q_1 against the two-dimensional range of stream sources and time (a) and the QTW (b) of Figure 1, respectively

such that $A(Q)$ is the exact answer to Q, i.e. the answer to Q evaluated against $\mathcal{A}$, and $\widetilde{A}(Q)$ is the approximate answer to Q, i.e. the answer to Q evaluated against the QTW built on top of $\mathcal{A}$.

For $\widetilde{A}(Q_0)$, $\varepsilon(Q_0) = 22\%$, whereas for $\widetilde{A}(Q_1)$, $\varepsilon(Q_0) = 16\%$, respectively. It follows that $\varepsilon(Q_1) < \varepsilon(Q_0)$. This is due to the fact that Q_1 overlaps the leaf nodes $n.2.1.1$, $n.2.1.2$, $n.2.1.3$, $n.2.1.4$, $n.2.4.1$, $n.2.4.2$, $n.2.4.3$ and $n.2.4.4$, beyond to the leaf nodes $n.2.2$ and $n.2.2$ (see Figure 2), whereas Q_0 overlaps the leaf nodes $n.1.3$, $n.1.4$, $n.2.3$, $n.3$ and $n.4$ (see Figure 2). Overall, the aggregation level of leaf nodes overlapped by Q_1 is lower than the aggregation level of leaf nodes overlapped by Q_0, and this, in turn, causes that evaluating Q_1 introduces a lower query error than evaluating Q_0. This confirms us our main intuition on the influence of QTW leaf nodes on the accuracy of approximate answers to range-SUM queries, according to the considerations discussed above.

Let us focus the attention on some properties and theoretical results deriving from the main intuition above. Given a QTW, it follows that *a partial compression* (see Section 2) *of the QTW produces a partially-compressed QTW, denoted by* $\widetilde{QTW}_p$, *having a number of nodes* $N_{\widetilde{QTW}_p}$ *between the two extreme bounds represented by the number of nodes of the full-compressed QTW* (see Section 2), *denoted by* $\widetilde{QTW}_f$, *namely* $N_{\widetilde{QTW}_f}$, *and the number of nodes of the uncompressed QTW, denoted by* $\widetilde{QTW}_u$, *namely* $N_{\widetilde{QTW}_u}$, *respectively.* From this, the following property holds:

$$N_{\widetilde{QTW}_f} < N_{\widetilde{QTW}_p} < N_{\widetilde{QTW}_u} \tag{3}$$

i.e. (from Equation (2)):

$$1 < N_{\widetilde{QTW}_p} < \sum_{k=0}^{P_{\widetilde{QTW}_u}} 4^k \tag{4}$$

such that $P_{\widetilde{QTW}_u}$ models the depth of $\widetilde{QTW}_u$.

From this theorem, another interesting theorem can be derived. Some concepts and definitions are necessary before enunciating this new theorem. Given a range-SUM query Q and a QTW, we introduce the concept of *degree of approximation of Q over the QTW*, denoted by $\delta_{QTW}(Q)$, which is defined as follows:

$$\delta_{QTW}(Q) \approx \Psi \cdot \frac{1}{\varepsilon(Q)} \tag{5}$$

such that Ψ is a function that captures the (inversely proportional) dependency between $\delta_{QTW}(Q)$ and $\varepsilon(Q)$, and $\varepsilon(Q)$ is the PRE due to evaluating Q against the QTW (Equation (2)). Without going into detail, it should be noted that the definition and the shape of Ψ depend on the nature of queries considered, and more or less accurate models for Ψ can be devised to this end. In this research, rather than studying and detailing the nature and characteristics of Ψ, we are interested in capturing the relation between $\delta_{QTW}(Q)$ and $\varepsilon(Q)$, and we left that study as future work.

Furthermore, given a QTW and a population of range-SUM queries $\mathcal{Q}$, we introduce the concept of *degree of approximation of $\mathcal{Q}$ over the QTW*, denoted by $\delta_{QTW}(\mathcal{Q})$, which is defined as follows:

$$\delta_{QTW}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \cdot \sum_{k=0}^{|\mathcal{Q}|-1} \delta_{QTW}(Q_k) = \frac{1}{|\mathcal{Q}|} \cdot \sum_{k=0}^{|\mathcal{Q}|-1} \frac{\Psi_k}{\varepsilon(Q_k)} \tag{6}$$

such that: Ψ_k models a function capturing the dependency between $\delta_{QTW}(Q_k)$ and $\varepsilon(Q_k)$ for $Q_k \in \mathcal{Q}$ over the QTW (note that, in the most general case, a *different* Ψ_k for each query Q_k exists), and $|\mathcal{Q}|$ models the cardinality of $\mathcal{Q}$.

Now, let us focus on the last theorem mentioned above. Given a *partially-compressed QTW $\widetilde{QTW}_p$* and a population of range-SUM queries $\mathcal{Q}$, it follows that *the degree of approximation of $\mathcal{Q}$ over $\widetilde{QTW}_p$, $\delta_{\widetilde{QTW}_p}(\mathcal{Q})$, keeps a linear dependency on the number of nodes of $\widetilde{QTW}_p$, $N_{\widetilde{QTW}_p}$*. Formally:

$$\delta_{\widetilde{QTW}_p}(\mathcal{Q}) \approx \Phi \cdot N_{\widetilde{QTW}_p} \tag{7}$$

such that Φ models a function capturing the linear dependency between $\delta_{\widetilde{QTW}_p}(\mathcal{Q})$ and $N_{\widetilde{QTW}_p}$.

Theorem (7) plays a critical role in our research, as it can be used as theoretical basis for devising simple-yet-effective query optimization schemes for the evaluation of range-SUM queries against the $MRDS$, in single-server environments as well as distributed environments. From theorem (7), two nice corollaries

complete our theoretical analysis on the influence of QTW nodes on the accuracy of approximate answers to range-SUM queries.

First corollary states what follows. Given a QTW having N_{QTW} nodes, for every arbitrary population of range-SUM queries $\mathcal{Q}$, the following property holds:

$$\lim_{N_{QTW} \to 1} \delta_{QTW}(\mathcal{Q}) = \delta_{\widetilde{QTW}_f}(\mathcal{Q}) \tag{8}$$

such that $\widetilde{QTW}_f$ denotes the full-compressed QTW.

Second corollary states what follows. Given a QTW having N_{QTW} nodes, for every arbitrary population of range-SUM queries $\mathcal{Q}$, the following property holds:

$$\lim_{N_{QTW} \to \sum_{k=0}^{P_{QTW}} 4^k} \delta_{QTW}(\mathcal{Q}) = \delta_{\widetilde{QTW}_u}(\mathcal{Q}) \tag{9}$$

such that $\widetilde{QTW}_u$ denotes the uncompressed QTW and P_{QTW} denotes the depth of the QTW.

Like theorem (7), both corollaries (8) and (9) are important for query optimization purposes in both single-server and distributed environments, as they can be used as theoretical basis for finding lower bounds (corollary (8)) and upper bound (corollary (9)) to the complexity due to evaluating range-SUM queries against the $MRDS$.

4 Refinement of Theoretical Contributions

In this Section, we provide theorems that formally summarize the main theoretical contributions of this research. Theorem 1 refers to the number of nodes of the partially-compressed QTW produced by the non-linear data stream compression framework [11].

Theorem 1. *For each partially-compressed QTW* $\widetilde{QTW}_p$ *produced by the non-linear data stream compression framework [11], the following property holds:* $1 < N_{\widetilde{QTW}_p} < \sum_{k=0}^{P_{\widetilde{QTW}_u}} 4^k$, *such that* $N_{\widetilde{QTW}_p}$ *denotes the number of nodes of* $\widetilde{QTW}_p$, $\widetilde{QTW}_u$ *denotes the uncompressed QTW, and* $P_{\widetilde{QTW}_u}$ *denotes the depth of* $\widetilde{QTW}_u$.

Theorem 2 instead focuses on the linear dependency between the degree of approximation of populations of range-SUM queries over partially-compressed QTW produced by the non-linear data stream compression framework [11] and the number of nodes of the partially-compressed QTW themselves.

Theorem 2. *Given a partially-compressed QTW* $\widetilde{QTW}_p$ *produced by the non-linear data stream compression framework [11] and a population of range-SUM queries* $\mathcal{Q}$, *the degree of approximation of* $\mathcal{Q}$ *over* $\widetilde{QTW}_p$, $\delta_{\widetilde{QTW}_p}(\mathcal{Q})$, *is linearly*

dependent on the number of nodes of $\widetilde{QTW}_p$, $N_{\widetilde{QTW}_p}$ *via the function* Φ, *as follows:* $\delta_{\widetilde{QTW}_p}(\mathcal{Q}) \approx \Phi \cdot N_{\widetilde{QTW}_p}$.

Finally, Corollary 1 and Corollary 2 directly derive from Theorem 2, and provide further properties on the degree of approximation of populations of range-SUM queries $\mathcal{Q}$ over partially-compressed QTW produced by the non-linear data stream compression framework [11].

Corollary 1. *Given a partially-compressed QTW* $\widetilde{QTW}_p$ *produced by the non-linear data stream compression framework [11], having* $N_{\widetilde{QTW}_p}$ *nodes, and a population of range-SUM queries* $\mathcal{Q}$, *the following property holds:* $\lim_{N_{\widetilde{QTW}_p} \to 1} \delta_{\widetilde{QTW}_p}(\mathcal{Q}) = \delta_{\widetilde{QTW}_f}(\mathcal{Q})$, *such that* $\delta_{\widetilde{QTW}_p}(\mathcal{Q})$ *denotes the degree of approximation of* $\mathcal{Q}$ *over* $\widetilde{QTW}_p$, *and* $\widetilde{QTW}_f$ *denotes the full-compressed QTW.*

Corollary 2. *Given a partially-compressed QTW* $\widetilde{QTW}_p$ *produced by the non-linear data stream compression framework [11], having* $N_{\widetilde{QTW}_p}$ *nodes, and a population of range-SUM queries* $\mathcal{Q}$, *the following property holds:* $\lim_{N_{\widetilde{QTW}_p} \to \sum_{k=0}^{P_{\widetilde{QTW}_u}} 4^k} \delta_{\widetilde{QTW}_p}(\mathcal{Q}) = \delta_{\widetilde{QTW}_u}(\mathcal{Q})$, *such that* $\delta_{\widetilde{QTW}_p}(\mathcal{Q})$ *denotes the degree of approximation of* $\mathcal{Q}$ *over* $\widetilde{QTW}_p$, *and* $\widetilde{QTW}_u$ *denotes the uncompressed QTW, and* $P_{\widetilde{QTW}_u}$ *denotes the depth of* $\widetilde{QTW}_u$.

5 Conclusions and Future Work

This paper has provided foundations and theoretical results of the so-called non-linear data stream compression model, a novel paradigm for supporting data stream miming algorithms effectively and efficiently. Particularly, the proposed model falls in that class of data stream mining applications where interesting knowledge is extracted via suitable collections of OLAP queries from data streams, via admitting approximate, i.e. compressed, representation models and query/mining results at the benefit of a more efficient and faster computation. Current and future work is mainly going to be oriented towards novel framework modules which will enable us to deal with updates that may occur in the target data stream. In that case, there is an important question which needs to be answered: *How can non-linearly compressed summarized data be managed effectively and efficiently?* Moreover, we intend to look further into the problem of optimization schemes for the evaluation of range-SUM queries against the MRDS in distributed environments, possibly including the use of integrity constraints.

Acknowledgements. The second author, Dr. Hendrik Decker, is supported by FEDER and the Spanish grants TIN2009-14460-C03 and TIN2010-17139.

References

1. Aggarwal, C., Han, J., Wang, J., Yu, P.S.: On Demand Classification of Data Streams. In: ACM SIGKDD (2004)
2. Babu, S., Widom, J.: Continuous Queries over Data Streams. In: ACM SIGMOD RECORD (September 2001)
3. Barbara, D., DuMouchel, W., Faloutsos, C., Haas, P.J., Hellerstein, J.M., Ioannidis, Y.E., Jagadish, H.V., Johnson, T., Ng, R.T., Poosala, V., Ross, K.A., Sevcik, K.C.: The New Jersey Data Reduction Report. IEEE Data Engineering Bulletin 20(4) (1997)
4. Cai, Y.D., Clutterx, D., Papex, G., Han, J., Welgex, M., Auvilx, L.: MAIDS: Mining Alarming Incidents from Data Streams. In: ACM SIGMOD (2004)
5. Chaudhuri, S., Motwani, R., Narasayya, V.: On Random Sampling over Joins. In: ACM SIGMOD (1999)
6. Chen, Q., Li, Z., Liu, H.: Optimizing Complex Event Processing over RFID Data Streams. In: IEEE ICDE (2008)
7. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1) (2012)
8. Cuzzocrea, A.: Overcoming Limitations of Approximate Query Answering in OLAP. In: IDEAS (2005)
9. Cuzzocrea, A.: Synopsis Data Structures for Representing, Querying, and Mining Data Streams. In: Ferragine, V.E., Doorn, J.H., Rivero, L.C. (eds.) Encyclopedia of Database Technologies and Applications (2008)
10. Cuzzocrea, A.: Intelligent Techniques for Warehousing and Mining Sensor Network Data. IGI Global (2009)
11. Cuzzocrea, A., Chakravarthy, S.: Event-based lossy compression for effective and efficient olap over data streams. Data Knowl. Eng. 69(7) (2010)
12. Cuzzocrea, A., Furfaro, F., Masciari, E., Saccà, D., Sirangelo, C.: Approximate Query Answering on Sensor Network Data Streams. In: Stefanidis, A., Nittel, S. (eds.) GeoSensor Networks (2004)
13. Cuzzocrea, A., Furfaro, F., Mazzeo, G.M., Saccá, D.: A Grid Framework for Approximate Aggregate Query Answering on Summarized Sensor Network Readings. In: Meersman, R., Tari, Z., Corsaro, A. (eds.) OTM-WS 2004. LNCS, vol. 3292, pp. 144–153. Springer, Heidelberg (2004)
14. Dobra, A., Gehrke, J., Garofalakis, M., Rastogi, R.: Processing Complex Aggregate Queries over Data Streams. In: ACM SIGMOD (2002)
15. Domingos, P., Hulten, G.: Mining High-Speed Data Streams. In: ACM SIGKDD (2000)
16. Gehrke, J., Korn, F., Srivastava, D.: On Computing Correlated Aggregates over Continual Data Streams. In: ACM SIGMOD (2001)
17. Gilbert, A., Guha, S., Indyk, P., Kotidis, Y., Muthukrishnan, S., Strauss, M.: Fast, Small-Space Algorithms for Approximate Histogram Maintenance. In: ACM STOC (2002)
18. Gilbert, A., Kotidis, Y., Muthukrishnan, S., Strauss, M.: One-Pass Wavelet Decompositions of Data Streams. IEEE Trans. on Knowledge and Data Engineering 15(3) (2003)
19. Greenwald, M., Khanna, S.: Space-Efficient Online Computation of Quantile Summaries. In: ACM SIGMOD (2001)
20. Guha, S., Koudas, N., Shim, K.: Data Streams and Histograms. In: ACM STOC (2001)

21. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering Data Streams: Theory and Practice. IEEE Trans. on Knowledge and Data Engineering 15(3) (2003)
22. Ho, C.-T., Agrawal, R., Megiddo, N., Srikant, R.: Range Queries in OLAP Data Cubes. In: ACM SIGMOD (1997)
23. Jiang, N., Gruenwald, L.: Research Issues in Data Stream Association Rule Mining. ACM SIGMOD Record 35(1) (2006)
24. Jiang, Q., Adaikkalavan, R., Chakravarthy, S.: MavEStream: Synergistic Integration of Stream and Event Processing. In: IEEE ICDT (2007)
25. Manku, G., Motwani, R.: Approximate Frequency Counts over Data Streams. In: VLDB (2002)
26. Muthukrishnan, S.: Data Streams: Algorithms and Applications. In: ACM-SIAM SODA (2003)
27. Samet, H.: The Quad-Tree and Related Hierarchical Data Structures. ACM Computing Surveys 16(2) (1984)
28. Vitter, J.: Random Sampling with a Reservoir. CM Trans. on Mathematical Software 11(1) (1985)

Reasoning with Qualitative Velocity: Towards a Hybrid Approach*

Golińska-Pilarek J.[1] and Muñoz-Velasco E.[2]

[1] Institute of Philosophy, University of Warsaw and
National Institute of Telecommunications, Poland
j.golinska@uw.edu.pl
[2] Dept. Applied Mathematics, University of Málaga, Spain
emilio@ctima.uma.es

Abstract. Qualitative description of the movement of objects can be very important when there are large quantity of data or incomplete information, such as in positioning technologies and movement of robots. We present a first step in the combination of fuzzy qualitative reasoning and quantitative data obtained by human interaction and external devices as GPS, in order to update and correct the qualitative information. We consider a Propositional Dynamic Logic which deals with qualitative velocity and enables us to represent some reasoning tasks about qualitative properties. The use of logic provides a general framework which improves the capacity of reasoning. This way, we can infer additional information by using axioms and the logic apparatus. In this paper we present sound and complete relational dual tableau that can be used for verification of validity of formulas of the logic in question.

Keywords: qualitative reasoning, propositional dynamic logic, relational logics, hybrid qualitative reasoning.

1 Introduction

Qualitative reasoning, QR, is an area of AI which tries to simulate the way of humans think in almost all situations. For example, we do not need to know the exact value of velocity and position of a car in order to drive it. When raising or answering questions about moving objects, both qualitative and quantitative responses are possible, as stated in [9]. However, human beings usually prefer to communicate using qualitative information according to their intuition rather than using quantitative values. Moreover, representing and reasoning with quantitative information can lead to information overload, that is, there is more information to be handled than the one that can be processed. A form of QR is order of magnitude reasoning, where the values are represented by different qualitative classes. For example, talking about velocity we may consider *slow*, *normal*, and *quick* as qualitative classes.

As said in [27], qualitative models can be seen as discrete abstractions of continuous and hybrid systems and can be fully explored by a verification tool providing conservative analysis of hybrid systems.

* Partially supported by projects TIN2006-15455-C03-01 and P6-FQM-02049.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 635–646, 2012.

The use of logic in QR, as in other areas of AI, provides a general framework which allows us to improve the capacity of solving problems and, as we will see in this paper, to deal with the reasoning problem. This way, we can infer additional information by using axioms and the logic apparatus. There are several applications of logics for QR (see e.g., [2,10]) and many of them concern spatial reasoning. As an example of logic for order of magnitude reasoning, see [4]; a theorem prover for one of these logics can be seen in [16]; and some implementations in [15,5].

Qualitative description of the movement of objects can be very important when there are large quantity of data, such as in positioning technologies (GPS, wireless communication) and movement of robots. In this direction, the concept of qualitative velocity [11, 28], together with qualitative distance and orientation, could help in order to represent spatial reasoning.

In a hybrid intelligence system, multiple techniques are used in order to obtain an efficient solution for a particular problem [7, 8, 14, 23, 1]. In our case, this combination is made by using qualitative reasoning, fuzzy aspects as in [3], and quantitative data obtained by human interaction and external devices as GPS, in order to update and correct the qualitative information. There are recent papers where similar combinations have been presented. For instance, in [26] a quantitative method is used for analyzing and comparing trajectories of robots using point distribution models; in [17] a simulator that combines qualitative reasoning, a geographic information system and targeted probabilistic calculations is presented; and in [31] a mix of qualitative and quantitative data is used for a hybrid modeling approach is studied.

Some papers [6, 18, 19] are developing the qualitative kinematics models studied in [13, 21, 12]. The relative movement of one object with respect to another has been studied by the Region Connection Calculus [24] and the Qualitative Trajectory Calculus [30,9]. However, as far as we know, the first paper which proposes a logic framework for qualitative velocity is [3], where a Propositional Dynamic Logic for order of magnitude reasoning to deal with the concept of qualitative velocity is proposed. The main advantages of this approach are: the possibility of constructing complex relations from simpler ones; the flexibility for using different levels of granularity; its possible extension by adding other spatial components, such as position, distance, cardinal directions, etc.; the use of a language close to programming languages; and, above all, the strong support of logic in spatial reasoning. Following [11], velocity of an object B with respect to another object A is represented by two components: module and orientation, each one given by a qualitative class. If we consider a velocity of B with respect to A, and another velocity of C with respect to B, the composition of these two velocities consists of obtaining the velocity of C with respect to A. For example, if (Q,l) represents a *quick* velocity towards the *left* orientation of B with respect to A, and (N,r) is a *normal* velocity towards the *right* of C with respect to B, the composition is a velocity of C with respect to A, that could be either (Q,l) or (N,l), that is, a *quick* or *normal* velocity towards the left orientation. The results of these compositions could depend on the specific problem we are dealing with. In the following section, we consider the logic QV where some assumptions about these compositions are posed in its models.

In this paper we present a first step in the construction of an hybrid approach to deal with qualitative velocity. We show a sound and complete relational dual tableau

for the Propositional Dynamic Logic of qualitative velocity introduced in [3], which can be used for verification of validity of its formulas. The system is based on Rasiowa-Sikorski diagrams for first-order logic [25]. The common language of most of relational dual tableaux is the logic of binary relations, which is a logical counterpart to the class RRA of (representable) relation algebras introduced by Tarski in [29]. The formulas of the classical logic of binary relations are intended to represent statements saying that two objects are related. Relations are specified in the form of relational terms. Terms are built from relational variables and/or relational constants with relational operations of union, intersection, complement, composition, and converse.

Relational dual tableaux are powerful tools for verification of validity as well as for proving entailment, finite model checking (i.e., verification of truth of a statement in a particular fixed finite model) and finite satisfaction (i.e., verification that a statement is satisfied by some fixed objects of a finite model). A comprehensive survey on applications of dual tableaux methodology to various theories and logics can be found in [22]. The main advantage of relational methodology is the possibility of representation within a uniform formalism the three basic components of formal systems: syntax, semantics, and deduction apparatus. Hence, the relational approach provides a general framework for representation, investigation and implementation of theories with different languages and/or semantics.

The paper is organized as follows. In Section 2 we present the Propositional Dynamic Logic of qualitative velocity, QV, its syntax and semantics. Relational formalization of the logic is presented in Section 3. In Section 4 we present the relational dual tableau for this logic, and we study its soundness and completeness; moreover, we show an example of the relational proof of validity of a QV-formula. Conclusions and final remarks are discussed in Section 5.

2 Logic QV for Reasoning with Qualitative Velocity

In this section we present the syntax and semantics of the logic QV for order of magnitude qualitative reasoning to deal with the concept of qualitative velocity. We consider the set of qualitative velocities $L_1 = \{\mathsf{z}, \mathsf{v}_1, \mathsf{v}_2, \mathsf{v}_3\}$, where $\mathsf{z}, \mathsf{v}_1, \mathsf{v}_2, \mathsf{v}_3$ represent zero, slow, normal, and quick velocity, respectively; and the set of qualitative orientations $L_2 = \{\mathsf{n}, \mathsf{o}_1, \mathsf{o}_2, \mathsf{o}_3, \mathsf{o}_4\}$ representing none, front, right, back, and left orientations, respectively. Thus, we consider four qualitative classes for the module of the velocities, and five qualitative classes for the orientation of the velocity. Orientations o_j and $\mathsf{o}_{\mathsf{j}+2}$, for $\mathsf{j} \in \{1, 2\}$, are interpreted as *opposite*. Furthermore, orientations o_j and $\mathsf{o}_{\mathsf{j}+1}$, for $\mathsf{j} \in \{1, 2, 3\}$, are interpreted as *perpendicular*.

The logic QV is an extension of Propositional Dynamic Logic PDL which is a framework for specification and verification of dynamic properties of systems. It is a multimodal logic with the modal operations of necessity and possibility determined by binary relations understood as state transition relations or input-output relations associated with computer programs. The vocabulary of the language of QV consists of symbols from the following pairwise disjoint sets: $\mathbb{V}$ - a countably infinite set of propositional variables; $\mathbb{C} = L_1 \times L_2$ - the set of constants representing labels from the set $L_1 \times L_2$; $\mathbb{SP} = \{\otimes_\star | \star \in \mathbb{C}\}$ - the set of relational constants representing specific

programs; $\{\cup, ;, ?, ^*\}$ - the set of relational operations, where $\cup$ is interpreted as a nondeterministic choice, ; is interpreted as a sequential composition of programs, ? is the test operation, and * is interpreted as a nondeterministic iteration; $\{\neg, \vee, \wedge, \rightarrow, [\,], \langle\rangle\}$ - the set of propositional operations of negation, disjunction, conjunction, implication, necessity, and possibility, respectively.

The set of QV-relational terms interpreted as compound programs and the set of QV-formulas are the smallest sets containing $\mathbb{SP}$ and $\mathbb{V} \cup \mathbb{C}$, respectively, and satisfying the following conditions:

- If S and T are QV-relational terms, then so are $S \cup T$, $S\,;T$, and T^*.
- If φ is a QV-formula, then $\varphi?$ is a QV-relational term.
- If φ and ψ are QV-formulas, then so are $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi$, and $\varphi \rightarrow \psi$.
- If φ is a QV-formula and T is a QV-relational term, then $[T]\varphi$ and $\langle T\rangle\varphi$ are QV-formulas.

Given a binary relation R on a set W and $X \subseteq W$, we define:

$$R(X) \stackrel{\text{df}}{=} \{w \in W \mid \exists x \in X,\ (x, w) \in R\}.$$

Fact 1. *For every binary relation R on a set W and for all $X, Y \subseteq W$:*

$$R(X) \subseteq Y \textit{ iff } (R^{-1}\,; (X \times W)) \subseteq (Y \times W).$$

A QV-*model* is a structure $\mathcal{M} = (W, m)$, where W is a non-empty set of states and m is a meaning function satisfying the following conditions:

- $W = \bigcup_{\star \in \mathbb{C}} \star$ where all $\star$'s are pairwise disjoint subsets of states understood as states of objects affected by a qualitative velocity
- $m(p) \subseteq W$ for every $p \in \mathbb{V}$
- $m(\star) = \star$, for every $\star \in \mathbb{C}$
- $m(\otimes_\star) \subseteq W \times W$, for every $\otimes_\star \in \mathbb{SP}$, and, in addition, for all $\mathsf{v}, \mathsf{v_r}, \mathsf{v_s} \in L_1$ and for all $\mathsf{o}, \mathsf{o_j}, \mathsf{o_{j+1}}, \mathsf{o_{j+2}} \in L_2$, the following hold:
  - (S1) $m(\otimes_{(\mathsf{v},\mathsf{o})})\,; m(\otimes_{(\mathsf{z},\mathsf{n})}) = m(\otimes_{(\mathsf{v},\mathsf{o})})$
  - (S2) $m(\otimes_{(\mathsf{v},\mathsf{o_j})})\,; m(\otimes_{(\mathsf{v},\mathsf{o_{j+2}})}) = m(\otimes_{(\mathsf{z},\mathsf{n})})$, for $\mathsf{j} \in \{1, 2\}$
  - (S3) $m(\otimes_{(\mathsf{v},\mathsf{o_{j+1}})})(m(\mathsf{v}, \mathsf{o_j})) \subseteq m(\mathsf{v}, \mathsf{o_j}) \cup m(\mathsf{v}, \mathsf{o_{j+1}})$, for $\mathsf{j} \in \{1, 2, 3\}$
  - (S4) $m(\otimes_{(\mathsf{v_s},\mathsf{o_{j+1}})})(m(\mathsf{v_r}, \mathsf{o_j})) \subseteq m(\mathsf{v_s}, \mathsf{o_{j+1}})$, for $\mathsf{j} \in \{1, 2, 3\}$ and $\mathsf{r} < \mathsf{s}$
  - (S5) $m(\otimes_{(\mathsf{v_s},\mathsf{o})})(m(\mathsf{v_r}, \mathsf{o})) \subseteq m(\mathsf{v_s}, \mathsf{o}) \cup m(\mathsf{v_3}, \mathsf{o})$, for $\mathsf{s} \in \{2, 3\}$ and $\mathsf{r} < \mathsf{s}$
  - (S6) $m(\otimes_{(\mathsf{v_s},\mathsf{o_{j+2}})})(m(\mathsf{v_r}, \mathsf{o_j})) \subseteq m(\mathsf{v_s}, \mathsf{o_{j+2}}) \cup m(\mathsf{v_{s-1}}, \mathsf{o_{j+2}})$, for $\mathsf{j} \in \{1, 2\}$, $\mathsf{s} \in \{2, 3\}$, and $\mathsf{r} < \mathsf{s}$

m extends to all the compound QV-relational terms and formulas:

- $m(T^*) = m(T)^* = \bigcup_{i \geq 0} m(T^i)$, where T^0 is the identity relation on W and $T^{i+1} \stackrel{\text{df}}{=} (T^i\,; T)$
- $m(S \cup T) = m(S) \cup m(T)$
- $m(S\,; T) = m(S)\,; m(T)$
- $m(\varphi?) = \{(s, s) \in W \times W : s \in m(\varphi)\}$

- $m(\neg\varphi) = W \setminus m(\varphi)$
- $m(\varphi \vee \psi) = m(\varphi) \cup m(\psi))$
- $m(\varphi \wedge \psi) = m(\varphi) \cap m(\psi))$
- $m(\varphi \rightarrow \psi) = m(\neg\varphi) \cup m(\psi))$
- $m([T]\varphi) = \{s \in W |$ for all $t \in W$, if $(s,t) \in m(T)$, then $t \in m(\varphi)\}$
- $m(\langle T\rangle\varphi) = \{s \in W |$ exists $t \in W$ such that $(s,t) \in m(T)$ and $t \in m(\varphi)\}$

Given a QV-model $\mathcal{M} = (W, m)$, a QV-formula φ is said to be *satisfied in* $\mathcal{M}$ *by* $s \in W$, $\mathcal{M}, s \models \varphi$ for short, whenever $s \in m(\varphi)$. As usual, a formula is true in a model whenever it is satisfied in all states of the model and it is QV-valid iff it is true in all QV-models. Intuitively, $(s, s') \in m(T)$ means that there exists a computation of program T starting in the state s and terminating in the state s'. Program $S \cup T$ performs S or T nondeterministically; program $S\,;T$ performs first S and then T. Expression $\varphi?$ is a command to continue if φ is true, and fail otherwise. Program T^* performs T zero or more times sequentially. For example, the formula $\langle(\mathsf{v}_1, \mathsf{o}_4)?\rangle\varphi$ is satisfied in s whenever s is a slow velocity towards the left orientation and φ is satisfied in s; the formula $[\otimes^*_{(\mathsf{v}_3,\mathsf{o}_2)}]\varphi$ is satisfied in s iff for every velocity s' obtained by the repetition of the composition of s with a quick velocity towards the right orientation a nondeterministically chosen finite number of times, φ is satisfied in s'; the formula $[\otimes_{(\mathsf{v}_1,\mathsf{o}_4)}\,;\otimes_{(\mathsf{v}_2,\mathsf{o}_2)}]\varphi$ is satisfied in s iff for every velocity s' obtained by composing s with a slow velocity towards the left followed by a normal velocity towards the right orientation, φ is satisfied in s'.

Example 1. Let us consider the case study of ball interception of simulated soccer agents, presented in [3]. Suppose that the ball is located at a point B and is moving with a velocity $(\mathsf{v}_\mathsf{b}, \mathsf{o}_\mathsf{b})$ and the robot is at point R and it is not moving at this instant (see Figure 1). Suppose also that the robot can calculate the qualitative velocity needed to catch the ball at the current instant and position and this velocity is $(\mathsf{v}_\mathsf{c}, \mathsf{o}_\mathsf{c})$. A simple vectorial argument leads us to the fact that the composition of both velocities has to be the velocity needed to catch the ball. This condition can be expressed in our language by the formula $(\mathsf{z}, \mathsf{n}) \rightarrow [\otimes_{(\mathsf{v}_\mathsf{c},\mathsf{o}_\mathsf{c})}\,;\otimes_{(\mathsf{v}_\mathsf{b},\mathsf{o}_\mathsf{b})}]\mathsf{OK}$, where (z, n) means that the robot is not moving at this instant, and OK means that the velocity of the robot is the correct one in order to catch the ball. the validity of this formula has to be checked as the robot is moving towards the ball and has to be corrected if it is not OK. The correction of this movement could require the human intervention and, on the other hand, the position of the robot may need some external device as a GPS.

3 Relational Representation of Logic QV

In this section we present the relational formalization of logic QV providing a framework for deduction in logic QV. First, we define the relational logic $\mathsf{RL}_{\mathsf{QV}}$ appropriate for expressing QV-terms and QV-formulas. Then, we translate all QV-terms and QV-formulas into relational terms and we show the equivalence of validity between a modal formula and its corresponding relational formula. The vocabulary of

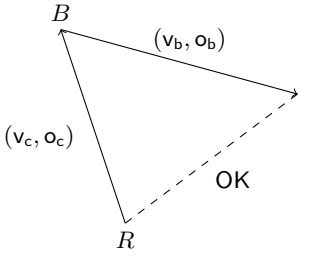


Fig. 1. Catching a ball when the robot is not moving

the language of the relational logic $\mathsf{RL_{QV}}$ consists of of symbols from the following pairwise disjoint sets: $\mathbb{OV} = \{x, y, z, \ldots\}$ - a countably infinite set of object variables; $\mathbb{RV} = \{P, Q, \ldots\}$ - a countably infinite set of binary relational variables; $\mathbb{RC} = \{1, 1'\} \cup \{R_\star, \Psi_\star | \star \in \mathbb{C}\}$ - the set of relational constants, where $\mathbb{C}$ is defined as in QV-models; $\mathbb{OP} = \{-, \cup, \cap, ;, ^{-1}, ^*\}$ - the set of relational operation symbols. The intuitive meaning of the relational representation of the symbols of logic QV is as follows: propositional variables are represented by relational variables; constants from $\mathbb{C}$ are represented by relational constants $\Psi_\star$ interpreted as right ideal binary relations; relational constants $R_\star$ correspond to specific programs $\otimes_\star$; the relational constants 1 (the universal relation), $1'$ (the identity relation), and relational operations are used in the representation of compound QV-formulas.

The set of $\mathsf{RL_{QV}}$-terms is the smallest set containing relational variables and relational constants and closed on all the relational operations. $\mathsf{RL_{QV}}$-formulas are of the form xTy, where T is an $\mathsf{RL_{QV}}$-relational term and x, y are object variables. An $\mathsf{RL_{QV}}$-model is a structure $\mathcal{M} = (W, m)$, where W is defined as in QV-models and m is the meaning function that satisfies:

- $m(P) \subseteq W \times W$, for every $P \in \mathbb{RV} \cup \{R_\star | \star \in \mathbb{C}\}$
- $m(\Psi_\star) = \star \times W$, for every $\star \in \mathbb{C}$
- $m(1')$ is an equivalence relation on W
- $m(1'); m(P) = m(P); m(1') = m(P)$, for every $P \in \mathbb{RV} \cup \mathbb{RC}$ (the extensionality property)
- $m(1) = W \times W$
- For all $\mathsf{v}, \mathsf{v_r}, \mathsf{v_s} \in L_1$ and for all $\mathsf{o}, \mathsf{o_j}, \mathsf{o_{j+1}}, \mathsf{o_{j+2}} \in L_2$, the following hold:

(RS1) $m(R_{(\mathsf{v},\mathsf{o})}); m(R_{(\mathsf{z},\mathsf{n})}) = m(R_{(\mathsf{v},\mathsf{o})})$
(RS2) $m(R_{(\mathsf{v},\mathsf{o_j})}); m(R_{(\mathsf{v},\mathsf{o_{j+2}})}) = m(R_{(\mathsf{z},\mathsf{n})})$, for $\mathsf{j} \in \{1, 2\}$
(RS3) $m(R_{(\mathsf{v},\mathsf{o_{j+1}})})^{-1}; m(\Psi_{(\mathsf{v},\mathsf{o_j})}) \subseteq m(\Psi_{(\mathsf{v},\mathsf{o_j})}) \cup m(\Psi_{(\mathsf{v},\mathsf{o_{j+1}})})$, for $\mathsf{j} \in \{1, 2, 3\}$
(RS4) $m(R_{(\mathsf{v_s},\mathsf{o_{j+1}})})^{-1}; m(\Psi_{(\mathsf{v_r},\mathsf{o_j})}) \subseteq m(\Psi_{(\mathsf{v_s},\mathsf{o_{j+1}})})$, for $\mathsf{j} \in \{1, 2, 3\}$ and $\mathsf{r} < \mathsf{s}$
(RS5) $m(R_{(\mathsf{v_s},\mathsf{o})})^{-1}; m(\Psi_{(\mathsf{v_r},\mathsf{o})}) \subseteq m(\Psi_{(\mathsf{v_s},\mathsf{o})}) \cup m(\Psi_{(\mathsf{v_3},\mathsf{o})})$, for $\mathsf{s} \in \{2, 3\}$ and $\mathsf{r} < \mathsf{s}$
(RS6) $m(R_{(\mathsf{v_s},\mathsf{o_{j+2}})})^{-1}; m(\Psi_{(\mathsf{v_r},\mathsf{o_j})}) \subseteq m(\Psi_{(\mathsf{v_s},\mathsf{o_{j+2}})}) \cup m(\Psi_{(\mathsf{v_{s-1}},\mathsf{o_{j+2}})})$, for $\mathsf{j} \in \{1, 2\}$, $\mathsf{s} \in \{2, 3\}$, and $\mathsf{r} < \mathsf{s}$

- m extends to all the compound relational terms as follows:

$$m(-T) = m(1) \cap -m(T), \qquad m(T^{-1}) = m(T)^{-1},$$
$$m(S \cup T) = m(S) \cup m(T), \qquad m(S\,;T) = m(S)\,;m(T),$$
$$m(S \cap T) = m(S) \cap m(T), \qquad m(T^*) = m(T)^*.$$

Observe that the conditions (RS1), ..., (RS6) are relational counterparts of the conditions (S1), ..., (S6) assumed in QV-models. An $\mathsf{RL_{QV}}$-model $\mathcal{M}$ in which $1'$ is interpreted as the identity is said to be *standard*. Let $v\colon \mathbb{OV} \to W$ be a valuation in an $\mathsf{RL_{QV}}$-model $\mathcal{M}$. An $\mathsf{RL_{QV}}$-formula xTy is said to be satisfied in $\mathcal{M}$ by v whenever $(v(x), v(y)) \in m(T)$. A formula φ is true in $\mathcal{M}$ iff it is satisfied in $\mathcal{M}$ by all the valuations and it is $\mathsf{RL_{QV}}$-valid whenever it is true in all $\mathsf{RL_{QV}}$-models.

Now, we define the translation τ of QV-terms and QV-formulas into $\mathsf{RL_{QV}}$-relational terms. Let τ' be a one-to-one mapping that assigns relational variables to propositional variables. The translation τ is defined as follows:

- $\tau(p) = (\tau'(p)\,;1)$, for every $p \in \mathbb{V}$
- $\tau(\star) = \Psi_\star$, for every $\star \in \mathbb{C}$
- $\tau(\otimes_\star) = R_\star$, for every $\otimes_\star \in \mathbb{SP}$

For all relational terms T and S:

- $\tau(T^*) = \tau(T)^*$
- $\tau(S \cup T) = \tau(S) \cup \tau(T)$
- $\tau(S\,;T) = \tau(S)\,;\tau(T)$
- $\tau(\varphi?) = 1' \cap \tau(\varphi)$
- $\tau(\varphi \vee \psi) = \tau(\varphi) \cup \tau(\psi)$
- $\tau(\varphi \wedge \psi) = \tau(\varphi) \cap \tau(\psi)$
- $\tau(\varphi \to \psi) = \tau(\neg\varphi \vee \psi)$
- $\tau(\langle T\rangle\varphi) = \tau(T)\,;\tau(\varphi)$
- $\tau([T]\varphi) = -(\tau(T)\,;-\tau(\varphi))$.

Relational terms obtained from formulas of logic QV include both declarative information and procedural information provided by these formulas. The declarative part which represents static facts about a domain is represented by means of a Boolean reduct of algebras of relations, and the procedural part, which is intended to model dynamics of the domain, requires the relational operations. In the relational terms which represent the formulas after the translation, the two types of information receive a uniform representation and the process of reasoning about both statics and dynamics, and about relationships between them can be performed within the framework of a single uniform formalism.

Theorem 1. *For every* QV*-formula* φ *and for all object variables* x *and* y, φ *is* QV*-valid iff* $x\tau(\varphi)y$ *is true in all standard* $\mathsf{RL_{QV}}$*-models.*

4 Relational Dual Tableau for QV

In this section we present a dual tableau for the logic $\mathsf{RL_{QV}}$ that can be used for verification of validity of QV-formulas. Relational dual tableaux are determined by the axiomatic sets of formulas and rules which apply to finite sets of relational formulas. The axiomatic sets take the place of axioms. The rules are intended to reflect properties of relational operations and constants. There are two groups of rules: decomposition

rules and specific rules. Although most often the rules of dual tableaux are finitary, the dual tableau system for logic QV includes an infinitary rule reflecting the behaviour of an iteration operation. Given a formula, the decomposition rules of the system enable us to transform it into simpler formulas, or the specific rules enable us to replace a formula by some other formulas. The rules have the following general form:

$$\text{(rule)} \qquad \frac{\Phi(\overline{x})}{\Phi_1(\overline{x}_1,\overline{u}_1,\overline{w}_1) \mid \ldots \mid \Phi_n(\overline{x}_j,\overline{u}_j,\overline{w}_j) \mid \ldots}$$

where $j \in J$, for some (possibly infinite) set J, $\Phi(\overline{x})$ is a finite (possibly empty) set of formulas whose object variables are among the elements of $\mathrm{set}(\overline{x})$, where $\overline{x}$ is a finite sequence of object variables and $\mathrm{set}(\overline{x})$ is a set of elements of sequence $\overline{x}$; every $\Phi_j(\overline{x}_j,\overline{u}_j,\overline{w}_j)$, $j \in J$, is a finite non-empty set of formulas, whose object variables are among the elements of $\mathrm{set}(\overline{x}_j) \cup \mathrm{set}(\overline{u}_j) \cup \mathrm{set}(\overline{w}_j)$, where $\overline{x}_j, \overline{u}_j, \overline{w}_j$ are finite sequences of object variables such that $\mathrm{set}(\overline{x}_j) \subseteq \mathrm{set}(\overline{x})$, $\mathrm{set}(\overline{u}_j)$ consists of the object variables that may be instantiated to arbitrary object variables when the rule is applied (usually to the object variables that appear in the set to which the rule is being applied), $\mathrm{set}(\overline{w}_j)$ consists of the object variables that must be instantiated to pairwise distinct new variables (not appearing in the set to which the rule is being applied) and distinct from any variable of sequence $\overline{u}_j$. A rule of the form (rule) is *applicable* to a finite set X of formulas whenever $\Phi(\overline{x}) \subseteq X$. As a result of an application of a rule of the form (rule) to set X, we obtain the sets $(X \setminus \Phi(\overline{x})) \cup \Phi_j(\overline{x}_j,\overline{u}_j,\overline{w}_j)$, for every $j \in J$. A set to which a rule is applied is called the *premise* of the rule, and the sets obtained by the application of the rule are called its *conclusions*. If the set J is finite, then a rule of the form (rule) is said to be *finitary*, otherwise it is referred to as *infinitary*. Thus, if J has n elements, then the rule of the form (rule) has n conclusions.

A finite set $\{\varphi_1,\ldots,\varphi_n\}$ of $\mathsf{RL_{QV}}$-formulas is said to be an $\mathsf{RL_{QV}}$-*set* whenever for every $\mathsf{RL_{QV}}$-model $\mathcal{M}$ and for every valuation v in $\mathcal{M}$ there exists $i \in \{1,\ldots,n\}$ such that φ_i is satisfied by v in $\mathcal{M}$. It follows that the first-order disjunction of all the formulas from an $\mathsf{RL_{QV}}$-set is valid in the first-order logic. A rule of the form (rule) is $\mathsf{RL_{QV}}$-*correct* whenever for every finite set X of $\mathsf{RL_{QV}}$-formulas, $X \cup \Phi(\overline{x})$ is an $\mathsf{RL_{QV}}$-set if and only if $X \cup \Phi_j(\overline{x}_j,\overline{u}_j,\overline{w}_j)$ is an $\mathsf{RL_{QV}}$-set, for every $j \in J$, i.e., the rule preserves and reflects validity. It follows that ',' (comma) in the rules is interpreted as disjunction and '|' (branching) is interpreted as conjunction.

$\mathsf{RL_{QV}}$-dual tableau includes decomposition rules of the following forms, for any object variables x and y and for any relational terms S and T:

$$(\cup)\ \frac{x(S\cup T)y}{xSy, xTy} \qquad (-\cup)\ \frac{x-(S\cup T)y}{x-Sy \mid x-Ty} \qquad (\cap)\ \frac{x(S\cap T)y}{xSy \mid xTy} \qquad (-\cap)\ \frac{x-(S\cap T)y}{x-Sy, x-Ty}$$

$$(-)\ \frac{x--Ty}{xTy} \qquad (^{-1})\ \frac{xT^{-1}y}{yTx} \qquad (-^{-1})\ \frac{x-T^{-1}y}{y-Tx}$$

$$(;)\ \frac{x(S;T)y}{xSz, x(S;T)y \mid zTy, x(S;T)y} \qquad (-;)\ \frac{x-(S;T)y}{x-Sz, z-Ty}$$

for any object variable z — for a new object variable z

$$(^*)\ \frac{xT^*y}{xT^iy, xT^*y} \qquad (-^*)\ \frac{x{-}(T^*)y}{x{-}(T^0)y \mid \ldots \mid x{-}(T^i)y \mid \ldots}$$
for any $i \geq 0$ where $T^0 = 1'$, $T^{i+1} = T\,;T^i$

Below we list the specific rules of $\mathsf{RL_{QV}}$-dual tableau.

For all object variables x, y, z and for every relational term $T \in \mathbb{RC}$:

$$(1'1)\ \frac{xTy}{xTz, xTy \mid y1'z, xTy} \qquad (1'2)\ \frac{xTy}{x1'z, xTy \mid zTy, xTy}$$

For every $\star \in \mathbb{C}$ and for all object variables x and y:

$$\text{(right)}\ \frac{x\Psi_\star y}{x\Psi_\star z, x\Psi_\star y}$$ for any object variable z

For every $T \in \{R_{(\mathsf{z},\mathsf{n})}\} \cup \{R_{(\mathsf{v_i},\mathsf{o_j})} \mid 1 \leq \mathsf{i} \leq 3, 1 \leq \mathsf{j} \leq 4\} \cup \{\Psi_\star \mid \star \in \mathbb{C}\}$ and for all object variables x and y:

$$\text{(cut)}\ \frac{}{xTy \mid x{-}Ty}$$

For all $\mathsf{v}, \mathsf{v_r}, \mathsf{v_s} \in L_1$, $\mathsf{o}, \mathsf{o_j}, \mathsf{o_{j+1}}, \mathsf{o_{j+2}} \in L_2$, and for all object variables x and y:

$$(r1 \subseteq)\ \frac{xR_{(\mathsf{v},\mathsf{o})}y}{xR_{(\mathsf{v},\mathsf{o})}z, xR_{(\mathsf{v},\mathsf{o})}y \mid zR_{(\mathsf{z},\mathsf{n})}y, xR_{(\mathsf{v},\mathsf{o})}y}$$
for any object variable z

$$(r1 \supseteq)\ \frac{x{-}R_{(\mathsf{v},\mathsf{o})}y}{x{-}R_{(\mathsf{v},\mathsf{o})}z, z{-}R_{(\mathsf{z},\mathsf{n})}y, x{-}R_{(\mathsf{v},\mathsf{o})}y}$$
for a new object variable z

$$(r2 \subseteq)\ \frac{xR_{(\mathsf{z},\mathsf{n})}y}{xR_{(\mathsf{v},\mathsf{o_j})}z, xR_{(\mathsf{z},\mathsf{n})}y \mid zR_{(\mathsf{v},\mathsf{o_{j+2}})}y, xR_{(\mathsf{z},\mathsf{n})}y}$$
for any object variable z and $\mathsf{j} \in \{1,2\}$

$$(r2 \supseteq)\ \frac{x{-}R_{(\mathsf{z},\mathsf{n})}y}{x{-}R_{(\mathsf{v},\mathsf{o_j})}z, z{-}R_{(\mathsf{v},\mathsf{o_{j+2}})}y, x{-}R_{(\mathsf{z},\mathsf{n})}y}$$
for a new object variable z and $\mathsf{j} \in \{1,2\}$

$$(r3)\ \frac{x\Psi_{(\mathsf{v},\mathsf{o_j})}y, x\Psi_{(\mathsf{v},\mathsf{o_{j+1}})}y}{zR_{(\mathsf{v},\mathsf{o_{j+1}})}x, K \mid z\Psi_{(\mathsf{v},\mathsf{o_j})}y, K}$$
for any object variable z, $\mathsf{j} \in \{1,2,3\}$
$K = x\Psi_{(\mathsf{v},\mathsf{o_j})}y, x\Psi_{(\mathsf{v},\mathsf{o_{j+1}})}y$

$$(r4)\ \frac{x\Psi_{(\mathsf{v_s},\mathsf{o_{j+1}})}y}{zR_{(\mathsf{v_s},\mathsf{o_{j+1}})}x, x\Psi_{(\mathsf{v_s},\mathsf{o_{j+1}})}y \mid z\Psi_{(\mathsf{v_r},\mathsf{o_j})}y, x\Psi_{(\mathsf{v_s},\mathsf{o_{j+1}})}y}$$
for any object variable z and $\mathsf{j} \in \{1,2,3\}$ and $\mathsf{r} < \mathsf{s}$

$$(r5)\ \frac{x\Psi_{(\mathsf{v_s},\mathsf{o})}y, x\Psi_{(\mathsf{v_3},\mathsf{o})}y}{zR_{(\mathsf{v_s},\mathsf{o})}x, K \mid z\Psi_{(\mathsf{v_r},\mathsf{o})}y, K}$$
for any object variable z and $\mathsf{s} \in \{2,3\}$
$\mathsf{r} < \mathsf{s}$ and $K = x\Psi_{(\mathsf{v_s},\mathsf{o})}y, x\Psi_{(\mathsf{v_3},\mathsf{o})}y$

$$(r6)\ \frac{x\Psi_{(\mathsf{v_s},\mathsf{o_{j+2}})}y, x\Psi_{(\mathsf{v_{s-1}},\mathsf{o_{j+2}})}y}{zR_{(\mathsf{v_s},\mathsf{o_{j+2}})}x, K \mid z\Psi_{(\mathsf{v_r},\mathsf{o_j})}y, K}$$
for any object variable z and $\mathsf{j} \in \{1,2\}$, $\mathsf{s} \in \{2,3\}$,
$\mathsf{r} < \mathsf{s}$, and $K = x\Psi_{(\mathsf{v_s},\mathsf{o_{j+2}})}y, x\Psi_{(\mathsf{v_{s-1}},\mathsf{o_{j+2}})}y$

A set of $\mathsf{RL_{QV}}$-formulas is said to be an $\mathsf{RL_{QV}}$-*axiomatic set* whenever it includes a subset of either of the following forms, for all object variables x, y, for every relational term T, for any $\star \in \mathbb{C}$, and for any $\# \in \mathbb{C} \setminus \{\star\}$:

(Ax1) $\{x1'x\}$
(Ax2) $\{x1y\}$
(Ax3) $\{xTy, x{-}Ty\}$
(Ax4) $\bigcup_{\star\in\mathbb{C}}\{x\Psi_\star y\}$
(Ax5) $\{x{-}\Psi_\star y, x{-}\Psi_\# y\}$

Let φ be an $\mathsf{RL_{QV}}$-formula. An $\mathsf{RL_{QV}}$-*proof tree for* φ is a tree with the following properties:

- The formula φ is at the root of this tree.
- Each node except the root is obtained by an application of an $\mathsf{RL_{QV}}$-rule to its predecessor node.
- A node does not have successors whenever its set of formulas is an $\mathsf{RL_{QV}}$-axiomatic set or none of the rules is applicable to its set of formulas.

Observe that the proof trees are constructed in the top-down manner, and hence every node has a single predecessor node. A branch of an $\mathsf{RL_{QV}}$-proof tree is said to be *closed* whenever it contains a node with an $\mathsf{RL_{QV}}$-axiomatic set of formulas. A tree is *closed* iff all of its branches are closed. An $\mathsf{RL_{QV}}$-formula φ is $\mathsf{RL_{QV}}$-*provable* whenever there is a closed $\mathsf{RL_{QV}}$-proof tree for it which is then refereed to as its $\mathsf{RL_{QV}}$-*proof*.

Theorem 2 (Relational Soundness and Completeness).
For every QV*-formula* φ *and for all object variables* x *and* y, φ *is* QV*-valid iff* $x\tau(\varphi)y$ *is* $\mathsf{RL_{QV}}$*-provable.*

Example 2. Let φ be a QV-formula of the following form

$$\varphi = (\mathsf{v},\mathsf{o}_1) \to [\otimes_{(\mathsf{v},\mathsf{o}_2)}]((\mathsf{v},\mathsf{o}_1) \vee (\mathsf{v},\mathsf{o}_2))$$

The translation of φ into $\mathsf{RL_{QV}}$-term is: $\tau(\varphi) = -\Psi_{(\mathsf{v},\mathsf{o}_1)} \cup -(R_{(\mathsf{v},\mathsf{o}_2)}\,;-(\Psi_{(\mathsf{v},\mathsf{o}_1)} \cup \Psi_{(\mathsf{v},\mathsf{o}_2)}))$. Figure 2 shows $\mathsf{RL_{QV}}$-proof of the formula $x\tau(\varphi)y$, which by Theorem 2 proves QV-validity of φ. In each node of the tree presented in the example we underline the formulas which determine the rule that has been applied during the construction of

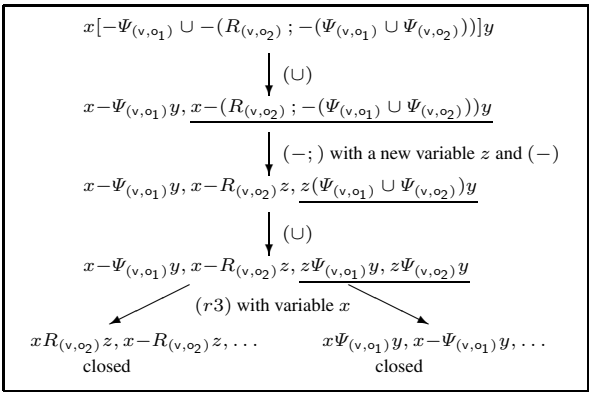


Fig. 2. $\mathsf{RL_{QV}}$-proof of QV-validity of the formula $(\mathsf{v},\mathsf{o}_1) \to [\otimes_{(\mathsf{v},\mathsf{o}_2)}]((\mathsf{v},\mathsf{o}_1) \vee (\mathsf{v},\mathsf{o}_2))$

the tree and we indicate which rule has been applied. If a rule introduces a variable, then we write how the variable has been instantiated. Furthermore, in each node we write only those formulas which are essential for the application of a rule and the succession of these formulas in the node is usually motivated by the reasons of formatting.

5 Conclusions and Future Work

We have presented a sound and complete relational dual tableau for verification of validity of QV-formulas. This system is a first step in order to provide a general framework for improving the capacity of reasoning about moving objects. The direction of our future work is twofold. First of all, we will focus on the extension of the logic by considering other spatial components (relative position, closeness, etc.). On the other hand, it would be needed a prover which is a decision procedure based on the dual tableau presented in this paper.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Bennett, B., Cohn, A., Wolter, A., Zakharyaschev, M.: Multi-dimensional modal logic as a framework for spatio-temporal reasoning. Applied Intelligence 17(3), 239–251 (2002)
3. Burrieza, A., Muñoz-Velasco, E., Ojeda-Aciego, M.: A PDL approach for qualitative velocity. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 19(01), 11–26 (2011)
4. Burrieza, A., Muñoz-Velasco, E., Ojeda-Aciego, M.: Closeness and Distance Relations in Order of Magnitude Qualitative Reasoning via PDL. In: Meseguer, P., Mandow, L., Gasca, R.M. (eds.) CAEPIA 2009. LNCS(LNAI), vol. 5988, pp. 71–80. Springer, Heidelberg (2010)
5. Burrieza, A., Ojeda-Aciego, M., Orłowska, E.: An implementation of a dual tableaux system for order-of-magnitude qualitative reasoning. International Journal of Computer Mathematics 86, 1852–1866 (2009)
6. Cohn, A., Renz, J.: Handbook of Knowledge Representation. Elsevier (2007)
7. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. Information Science 180, 2633–2634 (2010)
8. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
9. Delafontaine, M., Bogaert, P., Cohn, A.G., Witlox, F., Maeyer, P.D., de Weghe, N.V.: Inferring additional knowledge from $\mathsf{QTC}_{\mathsf{N}}$ relations. Information Sciences (2011), doi:10.1016/j.ins.2010.12.021
10. Duckham, M., Lingham, J., Mason, K., Worboys, M.: Qualitative reasoning about consistency in geographic information. Information Sciences 176(6), 601–627 (2006)
11. Escrig, M.T., Toledo, F.: Qualitative Velocity. In: Escrig, M.T., Toledo, F.J., Golobardes, E. (eds.) CCIA 2002. LNCS (LNAI), vol. 2504, pp. 28–39. Springer, Heidelberg (2002)
12. Faltings, B.: A symbolic approach to qualitative kinematics. Artificial Intelligence 56(2-3), 139–170 (1992)
13. Forbus, K., Nielsen, P., Faltings, B.: Qualitative kinematics: A framework. In: Proceedings of the Int. Joint Conference on Artificial Intelligence, pp. 430–437 (1987)
14. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)

15. Golińska-Pilarek, J., Mora, A., Muñoz-Velasco, E.: An ATP of a Relational Proof System for Order of Magnitude Reasoning with Negligibility, Non-closeness and Distance. In: Ho, T.-B., Zhou, Z.-H. (eds.) PRICAI 2008. LNCS (LNAI), vol. 5351, pp. 128–139. Springer, Heidelberg (2008)
16. Golińska-Pilarek, J., Muñoz-Velasco, E.: Dual tableau for a multimodal logic for order of magnitude qualitative reasoning with bidirectional negligibility. International Journal of Computer Mathematics 86, 1707–1718 (2009)
17. Hinrichs, T., Forbus, K., de Kleer, J., Yoon, S., Jones, E., Hyland, R., Wilson, J.: Hybrid Qualitative Simulation of Military Operations. In: Proc. Twenty-Third IAAI Conf. (2011)
18. Liu, W., Li, S., Renz, J.: Combining RCC-8 with qualitative direction calculi: Algorithms and complexity. In: Proceedings of IJCAI 2009, pp. 854–859 (2009)
19. Liu, H., Brown, D.J., Coghill, G.M.: Fuzzy qualitative robot kinematics. IEEE Transactions on Fuzzy Systems 16(3), 808–822 (2008)
20. Miene, A., Visser, U., Herzog, O.: Recognition and Prediction of Motion Situations Based on a Qualitative Motion Description. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 77–88. Springer, Heidelberg (2004)
21. Nielsen, P.: A qualitative approach to rigid body mechanics, University of Illinois at Urbana-Champaign, PhD thesis (1988)
22. Orłowska, E., Golińska-Pilarek, J.: Dual Tableaux: Foundations, Methodology, Case Studies. Trends in Logic, vol. 36. Springer Science (2011)
23. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
24. Randell, D., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: Proceedings of KR, pp. 165–176 (1992)
25. Rasiowa, H., Sikorski, R.: On gentzen theorem. Fund. Mathematicae 48, 57–69 (1960)
26. Roduit, P., Martinoli, A., Jacot, J.: A quantitative method for comparing trajectories of mobile robots using point distribution models. In: Proc. Intelligent Robots and Systems, IROS 2007, pp. 2441–2448 (2007)
27. Sokolsky, O., Hong, H.S.: Qualitative modeling of hybrid systems. In: Proc. of the Montreal Workshop (2001), http://www.cis.upenn.edu/~rtg/rtg_papers.html
28. Stolzenburg, F., Obst, O., Murray, J.: Qualitative Velocity and Ball Interception. In: Jarke, M., Koehler, J., Lakemeyer, G. (eds.) KI 2002. LNCS (LNAI), vol. 2479, pp. 283–298. Springer, Heidelberg (2002)
29. Tarski, A.: On the calculus of relations. Journal of Symbolic Logic 6, 73–89 (1941)
30. Van de Weghe, N., Kuijpers, B., Bogaert, P., De Maeyer, P.: A Qualitative Trajectory Calculus and the Composition of Its Relations. In: Rodríguez, M.A., Cruz, I., Levashkin, S., Egenhofer, M. (eds.) GeoS 2005. LNCS, vol. 3799, pp. 60–76. Springer, Heidelberg (2005)
31. Vries, D., Verheijen, P.J.T., den Dekker, A.J.: Hybrid system modeling and identification of cell biology systems: perspectives and challenges. In: Proc. 15th IFAC Symposium on System Identification, pp. 227–232 (2009)

Research of Neural Network Classifier Based on FCM and PSO for Breast Cancer Classification

Lei Zhang[1], Lin Wang[1], Xujiewen Wang[2], Keke Liu[2], and Ajith Abraham[3]

[1] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, 250022 Jinan, China
{Zhanglei,ise_wanglin}@ujn.edu.cn
[2] University of Jinan, 250022 Jinan, China
kevinxw@foxmail.com,dp_liukk@ujn.edu.cn
[3] Machine Intelligence Research Labs (MIR Labs), USA
ajith.abraham@ieee.org

Abstract. Breast cancer is one of the most common tumors related to death in women in many countries. In this paper, a novel neural network classification model is developed. The proposed model uses floating centroids method and particle swarm optimization algorithm with inertia weight as optimizer to improve the performance of neural network classifier. Wisconsin breast cancer datasets in UCI Machine Learning Repository are tested with neural network classifier of the proposed method. Experimental results show that the developed model improves search convergence and performance. The accuracy of classification of benign and malignant tumors could be improved by the developed method compared with other classification techniques.

Keywords: Floating Centroids Method, PSO, Neural Network, Breast Cancer Classification.

1 Introduction

Breast cancer is one of the most common malignant tumors among women. Various artificial intelligence techniques [1-6] have been used to improve the accuracy and efficiency of cancer diagnosis. In recent years, neural network has widely been applied to feature extraction, data compression, data classification, clustering etc. [7]. The learning algorithm of neural network is a supervised learning method. A classifier is usually constructed according to the character of a given datasets and analyzed the character of the given datasets. It is produced an exact model for each category data. In recent research, many classification techniques have been proposed, such as genetic algorithm [8], neural network [9-12], support vector machine [13-16], decision tree [17,18], etc.. Neural network model has been successfully used in many classification problems among these techniques.

Conducting neural network classifier, the number of parameters including input layer, hidden layer and output layer is decided by the property, category and characters of dataset. In neural network classifier, sample is mapped by neural network.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 647–654, 2012.

Centroid is a point in partition space and denotes the center of a class. In a conventional neural network classifier, position of centroids and the relationship between centroids and classes are set manually. In addition, number of centroids is fixed with reference to the number of classes. This fixed-centroid constraint decreases the chance to find optimal neural network. Therefore, a novel neural network classifier based on float centroids method (FCM) which removes the fixed-centroid constraint and spread the centroids throughout the partition space is proposed in our previous research [19]. However, despite the superior classification results, the training efficiency is not acceptable which limits the scope of application in practice. Therefore, this paper presents a developed method adopted particle swarm optimization algorithm with inertia weight as optimizer to improve the search convergence and performance of FCM.

The rest of the paper is organized as follows: Section 2 describes the particle swarm optimization algorithm with inertia weight. Section 3 provides the floating centroids method algorithm and the process of learning algorithm in detail. Experiment results are presented in Section 4, together with some comparisons with other classification techniques. In the end, Section 5 concludes with a brief summary of our study.

2 Particle Swarm Optimization Algorithm with Inertia Weight

The Particle Swarm Optimization (PSO) algorithm which is put forward by Kennedy and Eberhart is one of evolutionary algorithms [21]. It is an adaptive method that can be used to solve optimization problem. Conducting search uses a population of particle which corresponds to individuals. A population of particles is randomly generated initially. Each particle's position represents a possible solution point in the problem space. Each particle has an updating position vector x_i and updating velocity vector v_i by moving through the problem space. At each step, a fitness function evaluating each particle's quality is calculated by position vector x_i. The vector p_i represents the best ever position of each particle and p_g represents the best position obtained so far in the population.

For each iteration *t*, following the individual best position p_i and the global best position p_g , the velocity vector of particle *i* is updated by:

$$v_i(t+1) = v_i(t) + c_1\phi_1(p_i(t) - x_i(t)) + c_2\phi_2(p_g(t) - x_i(t)) \tag{1}$$

where c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. The velocity vector v_i is range of [-V_{max}, V_{max}]. Updating the velocity vector of particle by this method enables the particle to search around its

individual best position and global best position. Based on the updated velocity, each particle changes its position according to the following formula:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Based on this method, the population of particles tends to cluster together. To avoid premature convergence, the following formula can improve the velocity vector of each particle adding the inertia weight factor ω [21].

$$v_i(t+1) = \omega v_i(t) + c_1\phi_1(p_i(t) - x_i(t)) + c_2\phi_2(p_g(t) - x_i(t)) \tag{3}$$

where inertia weight factor ω is a real number. The inertia weight factor ω controls the magnitude of the velocity vector v_i.

The inertia weight factor ω is a ratio factor related to the previous velocity. The inertia weight factor ω decides the influence from the previous velocity to current velocity. The large inertia weight factor ω can make globally search and the small inertia weight factor ω will make local search [22]. The fellow method is adopted by the inertia factor ω from $\omega_{\max}$ to $\omega_{\min}$:

$$\omega = (\omega_{\max} - \omega_{\min}) * (\max iter - iter) / \max iter + \omega_{\min} \tag{4}$$

where *maxiter* is the total number of iteration, *iter* is the current iteration. Linearly decreasing the inertia factor ω from $\omega_{\max}$ to $\omega_{\min}$ can improve the search area firstly and locate the best position quickly. With the inertia weight factor ω gradually decreasing, the velocity of particle becomes slower and makes local search subtly. The convergence becomes faster owing to add the inertia weight factor.

3 The Procedure of Learning Algorithm for FCM and PSO

3.1 Floating Centroids Method

FCM [19] is an approach which has some floating centroids with class labels spread in the partition space. A sample is labeled by a certain class when the closest centroid of its corresponding mapped point is the closest centroids of this certain class. The partition space is decided by an irregular regions controlled by centroids. To get these floating centroids, training samples are mapped to partition space by neural network, and then, corresponding mapped points in partition space will be partitioned into several numbers of disjoint subsets using the K-Means clustering algorithm [20]. The floating centroids are defined as computing the cluster centers of these subsets. So, the number of centroids may be more than the number of classes. Finally, each of these centroids will be labeled by one class. If training points of one class are majority among all the points which belong to a centroid, the centroid is labeled by this class. More than one centroid can be labeled by one class.

The floating centroids method can achieve the optimization objective of neural network classifier, because it is no constraint in the number of centroids and the class label of centroids are computed by the mapped points in partition space. The floating centroids method has no fixed-centroid. The function of floating centroids method is that allocates points in the same class together as close as possible and separates points in the different classes from the partition space.

3.2 Model of Neural Network Classifier

A neural network classifier is a pair (F, P), where centroid in P have been colored. The model neural network classifier is described by Fig. 1.

If a new dataset need to be categorized, it can be mapped one point in P used by F, then predict the category according to the distance which measured by Euclidian distance. This process is illustrated in Fig.2. The dimension of partition space is set to two and the number of classes is three.

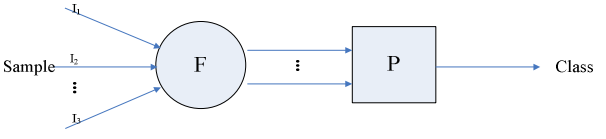


Fig. 1. Model of classifier

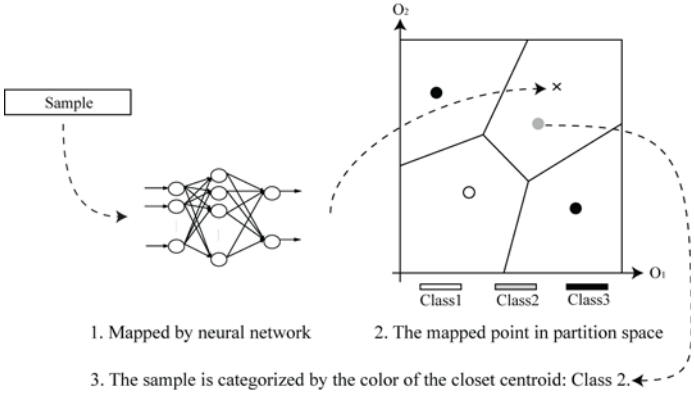


Fig. 2. Categorizing a new sample

3.3 Centroids Generation

Firstly, the training dataset is mapped to partition space using by neural network. The mapped point corresponding to training data is named colored point. The colored points in partition space are divided into k disjoint clusters used by K-means algorithm. Then each centroid is labeled category tag according to the principle which is

that if the color points of one class are majority in all of color points, the class should color the centroid. One class can color one or more centroids.

If the number of datasets between classes is different, it should keep equitableness using weights or other solutions. One class colors a centroid by a higher probability, if the number of this class dataset is larger than others.

3.4 The Process of Learning Algorithm

The goal of learning algorithm is to get the best neural network and it corresponding colored partition space. The process of learning algorithm is described by Algorithm 1.

Algorithm 1: The process of learning algorithm
Input: User-defined parameters and training data set;
Output: Best neural network and its corresponding colored partition space;
Step 1: Code neural network to form individual according to the particle swarm optimization mentioned with inertia weight in Section 2;
Step 2: while maximum generation has not been reached do;
Step 3: for *i*=1 to the number of individuals do;
Step 4: Decode individual id to form a neural network;
Step 5: Perform centroid generation with this neural network to get colored partition space;
Step 6: Use this neural network and its corresponding colored partition space to compute the value of optimization target function as the fitness of individual *i*;
Step 7: end for;
Step 8: Perform an iteration of improved particle swarm optimization algorithm, update individuals by their fitness;
Step 9: end while;
Step 10: return the best neural network and its corresponding colored partition space.

3.5 Target Function

The optimization target function is defined as the follow formula,

$$F = \sum_{i=1}^{s} \frac{1}{1 + e^{a(1-2Z_i)}} \tag{5}$$

where Z_i is the value of point *i* sample which is mapping to the partition space. *S* is total number of samples in the training dataset and *a* is a real constant which controls tortuosity. The target function is to put points in the same class together as close as possible and keep points in the different class away from others as far as possible.

4 Experiments and Results

The criterion of testing accuracy is defined to evaluate the performance of this method. Testing accuracy (TA) which is a method to obtain better generalization capability is adopted testing data to achieve the accurate rate.

$$TA = \frac{TP_{test}}{P_{test}} * 100\% \tag{6}$$

where TP_{test} represent the number of correctly classified samples in testing dataset and P_{test} is number of total samples in testing dataset.

To test the effectiveness of the proposed method, we select the Wisconsin breast cancer datasets for the experiments which is selected from the UCI machine learning database generated by Dr. William from the University of Wisconsin Madison [24]. Breast cancer can be diagnosed benign or malignant by puncture sampling. Wisconsin breast cancer dataset consists of 569 records and 2 classes, which 357 records are benign type and 212 records are malignant type. Each record has 30 attributes including ten quantifying feature, such as diameter, perimeter, symmetry etc.

The breast cancer datasets is divided into two subsets randomly, one is selected as the training dataset and the other is the testing dataset. There are 285 records as training dataset and 284 records as testing dataset for benign and malignant type. This two type experimental data have been done respectively. One class of sample is used for the benign and the other is used for the malignant. The normal data is labeled as 1 and the abnormal data is labeled as 0.

In our experiments, a three-layered feed-forward neural network with 15 hidden neurons is adopted and the particle swarm optimization algorithm with inertia weight is selected as optimizer. This proposed method is compared with the other neural network classifiers, including Flexible Neural Tree (FNT) [23], Neural Network (NN) and Wavelet Neural Network (WNN) [25]. The parameters of PSO algorithm are important factors on searching the optimal solution. Through trials on these datasets, the parameters setting of particle swarm optimization algorithm is defined by the follow Table 1.

Table 1. Parameters for experiments

Population size	20
Generation	3000
W_{max}	0.8
W_{mix}	0.4
C_1	2
C_2	2
V_{max}	3
V_{min}	-3

Table 2 depicts the accuracy results for testing data with comparison between our method and other methods. The result indicates that our proposed method is fully capable of improving the generalization ability of neural network classifier.

Table 2. Testing accuracy (%)

Cancer type	FNT	NN	WNN	Our method
Benign	93.31	94.01	94.37	96.47
Malignant	93.45	95.42	92.96	95.70

5 Conclusions

In this paper, a novel neural network is proposed based on floating centroids method and particle swarm optimization algorithm. This approach makes sample to be mapped into neural network and partition space is used to categorize data sample. The proposed method removes the fixed-centroid constraint. An improved particle swarm optimization with inertia weight is adopted as optimizer in this paper. It increases the variety of particles and improves the performance and convergence of standard particle swarm optimization algorithm. To evaluate the performance of our proposed approach, Wisconsin breast cancer datasets from UCI machine learning repository is selected to compare the accuracy measure with FNT, NN and WNN. Experimental results illustrate that our method is more efficient than other methods in classification accuracy of breast cancer dataset.

Acknowledgements. This work was supported by the Natural Science Foundation of Shandong Province No. ZR2011FL021, the Natural Science Foundation of China No.61173078, the Science and Technology Development Program of Shandong Province No. 2011GGX10116.

References

1. Hulka, B.S., Moorman, P.G.: Breast Cancer: Hormones and Other Risk Factors. Maturitas 38(1), 103–113 (2001)
2. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
3. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
4. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
5. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
6. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)

7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
8. Platt, J., Cristianini, N., Shawe Taylor, J.: Large Margin DAGs for Multiclass Classification. In: Advances in Neural Information Processing Systems, pp. 547–553 (2000)
9. Lu, H., Rudy, S., Huan, L.: Effect data mining using neural networks. IEEE Transcation Knowledge Data Engineer. 8(6), 957–961 (1996)
10. Misraa, B.B., Dehurib, S., Dashc, P.K., Pandad, G.: A reduced and comprehensible polynomial neural network for classification. Pattern Recognition 29(12), 1705–1712 (2008)
11. Daqi, G., Yan, J.: Classification methodologies of multilayer perceptions with sigmoid activation functions. Pattern Recognition 38(10), 1469–1482 (2005)
12. Chen, Y., Abraham, A., Yang, B.: Feature selection and classification using flexible neural tree. Neurocomputing 70(1-3), 305–313 (2006)
13. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)
14. Chua, Y.S.: Efficient computations for large least square support vector machine classifiers. Pattern Recognition 24(1-3), 75–80 (2003)
15. Koknar-Tezel, S., Latecki, L.J.: Improving SVM classification on imbalanced time series data sets with ghost points. Knowledge and Information Systems 28(1), 1–23 (2011)
16. Benjamin, X.W., Japkowicz, N.: Boosting support vector machines for imbalanced data sets. Knowledge and Information Systems 25(1), 1–20 (2011)
17. Qinlan, J.R.: Introduction of decision trees. Machine Learning 1(1), 81–106 (1986)
18. Wang, L., Yang, B., Chen, Y., Abraham, A., Sun, H., Chen, Z., Wang, H.: Improvement of Neural Network Classifier using Floating Centroids. Knowledge and Information Systems (2011), doi: 10.1007/s10115-011-0410-8
19. Freund, Y.: Boosting a weak learning algorithm by majority. Information Computing 121(2), 256–285 (1995)
20. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. Applied Stat. 28(1), 100–108 (1979)
21. Eberhart, R.C., Shi, Y.: Particle Swarm Optimization: Developments, Applications and Resources. In: Proc. of the Congress on Evolutionary Computation, pp. 81–86 (2001)
22. Huang, C.-P., Xiong, W.-L., Xu, B.-G.: Influnce of Inertia Weight on Astringency of Particle Swarm Algorithm and Its Improvement. Computer Engineering 34(12), 31–33 (2008)
23. Yang, B., Wang, L., Chen, Z., Chen, Y., Sun, R.: A novel classification method using the combination of FDPS and flexible neural tree. Neurocomputing 73(4-6), 690–699 (2010)
24. Blake, C.L., Merz, C.J., Newman, D.J., et al.: UCI Repository of Machine Learning Databases (2002), `http://www.ics.uci.edu/mlearn/MLRepository.html`
25. Chen, Y., Abraham, A.: Hybrid Neurocomputing for Detection of Breast Cancer. In: The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST 2005), pp. 884–892 (2005)

Improving Evolved Alphabet Using Tabu Set*

Jan Platos and Pavel Kromer

Department of Computer Science, FEECS
VSB-Technical University of Ostrava
17. listopadu 15, 708 00, Ostrava Poruba, Czech Republic
{jan.platos,pavel.kromer}@vsb.cz

Abstract. Data compression is very important today and it will be even more important in the future. Textual data use only limited alphabet - total number of used symbols (letters, numbers, diacritics, dots, spaces, etc.). In most languages, letters are joined into syllables and words. Both these approaches has pros and cons, but none of them is the best for any file. This paper describes a variant of algorithm for evolving alphabet from characters and 2-grams, which is optimal for compressed text files. The efficiency of the new variant will be tested on three compression algorithms and a new compression algorithm based on LZ77 will be also used with this new approach.

Keywords: genetic algorithms, data compression, tabu search, alphabet optimization, Huffman encoding, LZ77.

1 Introduction

Data compression is very important today and it will be even more important in the future. The reason is that more and more documents, pictures, videos and other data files must be stored or transferred into data storage or cloud. When we focus on text documents, we must think about collection of email, news and magazine articles, web encyclopedias, books, logs and other textual data. Textual data use only limited alphabet - total number of used symbols (letters, numbers, diacritics, dots, spaces, etc.). The number of used symbols depends on used language. English text files use up to 90 symbols (small and capital letters, numbers, dashes, strokes, dots, spaces, ...). But textual data has one amazing feature. The basic symbols are combined into sequences according defined rules. These rules are called grammar, linguistic, etc. which defines language. These rules should be studied from many points of view. Some papers studied language and its fractal features [2]. In text compression, other point of view is used. In most languages, letters are joined into syllables and words. These two language elements are very useful for data compression. The most efficient compression algorithms use context information for data compression. The context defines a frequency of a symbol according to the n previous symbols or existents of same

* This work was supported by the Grant Agency of the Czech Republic, under the grant no. P202/11/P142.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 655–666, 2012.

sequences in the data. Both, syllables and words, could be used a basic symbols in compression instead of characters.

The biggest problem is selection between these approaches. In [12], authors compared the single file parsing methods used on input text files of a size 1KB-5MB by means of the Burrows-Wheeler Transform for different languages (English, Czech, and German). They considered these input symbol types: letters, syllables, words, 3-grams, and 5-grams. Comparing letter-based, syllable-based, and word-based compression, they found out that character-based compression is the most suitable for small files (up to 200KB) and that syllable-based compression is the best for files of a size 200KB-5MB. Compression which uses natural text units such as words or syllables is 10-30% better than compression with 5-grams and 3-grams. For larger files, word-based compression methods will be the best. In [9] a three different so-called parsing strategies and their influence on Burrows-Wheeler based data compression [5]. In that paper a 1-grams, 2-grams, 3-grams, 4-grams and words were tested. The results show that the usage of words has meaning for large text files. And also that the second best approach is using of characters (1-grams). The 2-grams have always worse results than characters.

The approach, which is not mentioned in the previous paragraph, is usage of combination of characters, syllables and words as an alphabet. The problem is that such alphabet is big and the parsing of the file is ambiguous. Therefore, an algorithm which selects the proper alphabet should be defined. In our previous papers [16,15,17] we studied the ability of genetic algorithms (GA) to find an optimal alphabet for various text files. In first paper [17], we test ability of GA to find a minimal alphabet for a file, i.e. identify these symbols which are necessary for reading of a file. In the following papers [16,17] we try to evolve an optimal alphabet for various files. In this paper, we improved our algorithm using Tabu search. Moreover, we test our algorithm on new LZ77 algorithm, respectively on its modification LZSS.

The rest of the paper is organized as follows. The Section 2 describes the basic of data compression and the Section 3 describe basics of the genetic algorithms and Tabu search. Section 4 is focused on the description of proposed algorithm and Section 5 contain experimental result of the basic algorithm on several files. In the last Section, a conclusion of the first experiments and future work is defined.

2 Data Compression

The area of data compression is wide and complex and its origin dates back to the mid-20th century, when the theory of information was defined [21]. In information theory, information is represented as data. The amount of information contained in the data can be expressed as the Shannon entropy [21]. Entropy is measured in units such as bits and the same unit is used for the data. The rest of the data is redundant and may be eliminated. For example, in [22] Shannon experimentally discovers that the entropy for English is between 0.6 to 1.3 bits

per character. The elimination of the redundancy is also called data compression. Therefore, the entropy may also be explained as a maximal limit for data compression.

Many compression algorithms and transformations have been developed. Compression algorithms usually process data in their original form, but it is possible to use one or more transformations for the conversion of data into a more suitable form.

Compression algorithms may be divided into three main categories. The first category is statistical algorithm. Statistical methods use probability of the symbols for assigning shorter codes to more probable symbols and longer codes to least probable ones. Because of this, these methods use only necessary amount of bits to represent data and, therefore, they are able to eliminate all redundancy. The represent ants of this algorithms are Shannon-Fano encoding [21], Huffman Encoding [8], Arithmetic encoding [1,18,19] and Prediction by Partial Match algorithm [6,14].

The second category is dictionary-based algorithms. Dictionary-based methods use a dictionary as a memory of processed data (sequence of symbols) and the actual processed data are encoded as pointers to the previous occurrence of the symbol sequence. The type of dictionary and representation of the pointers depends on the algorithms. Almost all algorithms in this category are based on the following two algorithms. LZ77 [26] represent the dictionary as a structure called a sliding window of a certain size. This window contains processed symbols (in their raw form) and the new data are encoded as a triplet describing a similar sequence found in the sliding window. The triplet contains the length of the sequence found, the position of the sequence, and the first different symbol after the sequence. The last part of the triplet solves the problem, when no sequence is found in the dictionary. The LZ78 [27] algorithm uses different approach. The dictionary is dynamically created from the data by phrases. These phrases are created from the phrases in the dictionary and the next symbol. The pointers are encoded as a doublet. The first part of the doublet is the number of the phrase in the dictionary and the second one is the first different symbol.

The third category is Transformation for data compression. These transformations are used as a pre-processors before one of the compression algorithms from the first or second category is used. The most well know transformation for data compression are Run-Length encoding [20], Move-To-Front transformation [4] and Burrows-Wheeler transformation [5].

More information about data compression may be found in [20].

3 Genetic Algorithms

Genetic algorithms are a popular variant of evolutionary algorithms. They are based on the programmatical implementation of genetic evolution and they emphasize selection and crossover as the most important operations in the whole evolutionary optimization process [10,13].

Genetic algorithms evolve a population of chromosomes representing potential problem solutions encoded into suitable data structures. The evolution is performed by genetic operators modifying the chromosomes, i.e. the encoded forms of problem solutions. Proper encoding is vital for the effectiveness of the evolutionary searches. It defines the genotype, the space of all encoded problem solutions, which is different from the phenotype, the space of all problem solutions. Genetic algorithms explore the genotype of the problem being investigated and the size and shape of the problem genotype define its fitness landscape.

Finding good encoding is a non-trivial and problem-dependent task affecting the performance and results of an evolutionary search in a given problem domain. The solutions might be encoded into binary strings, integer vectors or real vectors, or more complex, often tree-like, hierarchical structures. The encoding choice is based on the needs of a particular application area. The original (canonical) GA described by John Holland encoded problems as fixed-length binary strings.

The iterative phase of an evolutionary search process starts with an initial population of individuals that can be generated randomly or seeded with potentially good solutions. Artificial evolution consists of the iterative application of genetic operators, introducing to the algorithm evolutionary principles such as inheritance, the survival of the fittest, and random perturbations. Iteratively, the current population of problem solutions is modified with the aim of forming a new and, it is hoped, better population to be used in the next generation. The evolution of problem solutions ends after specified termination criteria have been satisfied, and especially the criterion of finding an optimal solution. However, the decision as to whether a problem solution is the best one (i.e. a global optimum was reached) is impossible in many problem areas. After the termination of the search process, the evolution winner is decoded and presented as the most optimal solution found.

3.1 Tabu Set

A Tabu Search algorithm [7] is local search optimization technique that prevents testing of already tested solutions. The Tabu Search algorithm uses tabu sets of already tested solutions. Each new generated solution is compared to these sets and if a solution was already tested it is discarded and another one is created. Because of the size of search space, it is necessary to manage memory consumptions of the tabu set. The possible solution is represented as a vector of binary numbers in our algorithm. Such vectors should be stored very efficiently using binary trees, where similar vectors share its representing nodes of binary tree. Even such efficient implementation of tabu set needs a large amount of memory, when the length of the vectors is big. Therefore, a memory recycling method must be used. Many algorithms for memory recycling exist, such as Least Recently Used. In our algorithm we used a total releasing algorithm, i.e. when a maximum capacity is reached, all stored solution are released.

4 Optimizing Compression Alphabet

As was mentioned in Introduction, in our previous papers, we search for an optimal alphabet for text files. We used all single characters and all 2-grams from the compressed data as a search space. The problem of the approach defined in these papers is the speed of optimization. Therefore, a Tabu Search algorithm was introduced. The following paragraphs contain description of our algorithm.

The problem of finding optimal alphabet is very difficult. The main problem is to select elements of the alphabet. For character based compression it is very easy because the optimal alphabet is the same as the minimal alphabet. When only 80 symbols is used in the file, all these symbols must be put into alphabet, otherwise, the file in not readable for the algorithm. When the number of symbol is increased to 2-grams, 3-grams, etc., the problem becomes very difficult. The one way is to test all possible combination of input symbols. Unfortunately, the number of all possible combination for the alphabet containing n elements is 2^n. Therefore, some optimization algorithm that is able to search through this complex space must be used. In our algorithm a genetic algorithms was used. This decision was made because the genetic algorithms are very efficient in searching for optimal set of items from the complex space. Genetic algorithms were used in text compression in some previous works such as [24] and [11]. In both cases, GA are used for selection of optimal set of syllables.

4.1 Definition of the Algorithm

The algorithm consists from several steps. The first step is the extraction of the possible alphabet from the file. This is done using an algorithm that is able to find all possible symbols up to certain length. In our case the maximal length of the symbol is two characters. The best way how to explain function of the extraction algorithm is to present an example.

```
Input:  abcd
Output: a, ab, b, bc, c, cd, d
```

As may be seen, even the overlapping sequences are extracted as possible symbols. The reason is that there is no possibility to say, that `ab` is better than `bc`.

The second step is the main optimization algorithm based on Genetic algorithms. The used alphabet may be represented as the bit vector of length equal to the number of possible symbols, where the 0 means that the symbols was not used and the 1 means that the symbol was used. The selection operator was defined as *semi-elitary*. This means the one parent was selected randomly and the second is selected according its fitness value. This brings balance between convergences to local optimum and search of possible new areas of search space. The crossover operator was defined as *two-point crossover* and the mutation simply change the bit value according to defined probability. Each chromosome from the initial population has set ones for one character length symbols and

with probability of 0.5% has set one for other symbols. This initialization is motivated by the fact that using of minimal one-character length alphabet leads to the best possible compression in character based approach. The small probability of the setting of 2-grams is motivated by the idea that only few 2-grams have good impact on the compression results.

The last important step is the definition of the fitness function. The fitness function must be defined in the way that not only the size of the compressed file is taken into account, but also the size of the stored alphabet. The size of the compressed file depends on the used compression algorithm. The size of the alphabet is defined by the number of symbols and the way, how it is stored. The alphabet is stored using very simple algorithm - as a list of pairs *(length, sequence)* where *length* is the length of the sequence and *sequence* is a sequence of the characters.

The reason for counting the size of the alphabet into fitness function is, that if we take only the size of the compressed data, the GA will use most of the symbols from the search space to reduce the compressed data. This is because is we take all 2-grams then the number of symbols which must be compressed is $2\times$ smaller than is we use only characters.

The fitness function gets the length of the compressed file and the length of the stored alphabet and sum of these two numbers is the fitness value. The goal of the genetic algorithm is to minimize this fitness value.

5 Experiments

The experiments for this task were set in the following way. The goal is to test the efficiency of the evolved alphabet. The efficiency may differ for different files; we use several files with different parameters.

5.1 Testing Files

Because the whole algorithm is focused on optimization of text files compression, we select all text files from the Canterbury compression corpus [3] except artificial files. From the main corpus, all files with textual content were selected: *alice29.txt* - Alice's Adventures in Wonderland from Lewis Carroll, *asyoulik.txt* - a Shakespeare play As You Like It, *cp.html* - a html file, *fields.c* - a C source code, *grammar.lsp* - a Lisp source code, *lcet10.txt* - a technical text, *plrabn12.txt* - a poem Paradise Lost by John Milton and *xargs.1* - a GNU manual page. The summary of the files with their size and the number of unique 1-grams an 2-grams is depicted in Table 1. The most important column is the last one because values in this column define the sizes of the problem space.

5.2 Parameters of the Algorithm

The suggested algorithm has two types of parameters which must be set. The first group consists from parameters of the GA. The Genetic algorithms have

Table 1. Summary of testing files

File Name	Size (B)	1-grams	2-grams	Total
alice29.txt	152 089	74	1285	1359
asyoulik.txt	125 179	68	1125	1193
cp.html	24 603	86	1520	1606
fields.c	11 150	90	756	866
grammar.lsp	3 721	76	458	534
lcet10.txt	426 754	84	1934	2018
plrabn12.txt	481 861	81	1192	1273
xargs.1	4227	74	579	653

several parameters which must be set according to solved problem. Moreover, it is usual to design another genetic algorithm which evolves parameter to another GA algorithm. In our case, the parameters which must be set are the probability of the cross-over, probability of the mutation and size of the population. Other parameters were set as follows: the crossover type is two-point, selection is semi-elitist, the GA looking for the minimal fitness value and the termination criterion is set to 20000 generations. The probabilities of crossover and mutation were set to 90% and 0.5% respectively according to many simulations performed before the main experiments and published in [16]. These values satisfy the good ability of fitness function improvement. Moreover, this small value of mutation probability enables only very few added/removed symbols per iteration that is good for the maintaining of the ability of alphabet successfully parse the input file. The size of the population was set to 100. This number enables well enough heterogeneity of possible solution.

The second group of parameters which must be set is the used compression algorithm. For our experiment, LZW [25], Adaptive Huffman encoding [8,20] and Burrows-Wheeler based [5] compression algorithms were used. The same algorithm were used in the previous work. The LZW algorithm is very easy to implement, fast and uses context information from the input file for improving compression. This algorithm uses a variant with 128kb dictionary and tokens stored into files are stored with only necessary amount of bits. The Adaptive Huffman encoding is also very simple for implementation. We used a 32-implementation with initialization of the whole alphabet to same frequency. For the Burrows-Wheeler based compression algorithm we use an Adaptive Huffman Encoding and Run-length encoding in combination with Move-To-Front transform.

As an improvement, we also use a LZ77 based algorithm. LZ77 [26] is another type of dictionary based compression algorithms. It was developed one year before LZW, but its implementation is much complicated. It uses a so-called sliding window and the new data are encoded as a reference to this dictionary. The reference is in basic algorithm encoded as a triplet $(position, length, next\_char)$, where *position* and *length* is the position and length of the symbols sequence in the dictionary and the *next_char* is the first character that differs from the sequence. In our approach we use a variant of LZ77 developed by Storer and

Szymanski called LZSS [23]. In this approach a reference to the dictionary is encoded as a doublet, when the sequence is found, or a single character, when the sequence was not found. For identification of these two approaches a one bit flag is used. This algorithm is very popular in these days, most of the compression algorithm uses it, e.g. Zip, Rar, 7-zip. The main advantage of this algorithm is the possibility of combination of this algorithm with other algorithms. This combination means that the components of the reference to the dictionary should be compressed with another compression algorithm. In our algorithm we useAdaptive Huffman Encoding for compression of *next_char* and *length* and Fibonacci encoding for *lenght*. The dictionary size in our experiment is 128kB and the maximal length of the match is 64 symbols.

The main measured value is the compression ratio. The improvement against the 1-grams and 2-gram approach was studied in the previous papers. In this paper, we will study the ability of Tabu Search to improve the efficiency of the algorithm. In the tables we depict the size of the original file, results with a 1-gram alphabet, the results without Tabu Search and the results with Tabu Search. The 1-grams approach uses only the minimal alphabet as mentioned in table 1. The algorithm may decide to use all symbols - 1-grams and 2-grams without any constraint. The main task of this algorithm is to compare our suggested algorithm for three different compression algorithms. First we will try to compare achieved compression ratio and improvement in compression ratio for all groups of testing file and comparison of alphabets evolved for all three compression algorithms. All experiments were 10× repeated and results were averaged to get more relevant results.

The result tables contain several columns. Results achieved with 1-grams alphabet are depicted in column signed as *1-grams*, the results achieved with the alphabet evolved using suggested algorithm without Tabu Search are depicted in column *Without TS* and results achieved with the alphabet evolved using suggested algorithm with Tabu Search are depicted in column *With TS*. *CS* means the compressed size, *CR* means compression ratio and is computed as $CR = \frac{CS}{Orig.size}$, where *Orig.size* is the original size of the file. The column *AS* contains the number or the average number of symbols used for compression of the file. The column *Diff CR* contains a difference in compression ratio between algorithm without Tabu Search and compression ratio with Tabu Search. Each table contains in last line the average compression ratios for the 1-grams and both evolved alphabet and the average difference between two variants of the algorithm.

5.3 Results for the Files from Canterbury Corpus

The results for files from Canterbury corpus are depicted in Tables 2, 3 and 4. The improvements gained by evolved alphabet were described in our previous papers [16,17]. Therefore, their description here will be very short. The average improvement in compression ratio for LZW compression is almost 1.6% in average, but the best result is almost 2.5%. The average improvement for Burrows-Wheeler based compression is almost 2.2% against 1-gram alphabet.

Table 2. Results for Canterbury corpus and LZW compression algorithms

	Orig.	1-grams		Without TS		With TS		Diff.
FileName	**size [B]**	**CS [B]**	**CR [%]**	**CS [B]**	**CR [%]**	**CS [B]**	**CR [%]**	**CR [%]**
alice29.txt	152 089	62221	40.91	59338.7	39.02	59288.8	38.98	0.03
asyoulik.txt	125 179	54966	43.91	53690.2	42.89	53435.4	42.69	0.20
cp.html	24 603	11385	46.27	11095.5	45.10	11037.2	44.86	0.24
fields.c	11 150	5063	45.41	4773.7	42.81	4796.8	43.02	-0.21
grammar.lsp	3 721	1895	50.93	1817	48.83	1818.8	48.88	-0.05
lcet10.txt	426 754	163159	38.23	152744.1	35.79	152792.0	35.80	-0.01
plrabn12.txt	481 861	198443	41.18	192368.7	39.92	192077.6	39.86	0.06
xargs.1	4 227	2417	57.18	2342.1	55.41	2349.2	55.58	-0.17
Average			45.50		43.72		43.71	0.01

Table 3. Results for Canterbury corpus and BW compression algorithm

	Orig.	1-grams		Without TS		With TS		Diff.
FileName	**size [B]**	**CS [B]**	**CR [%]**	**CS [B]**	**CR [%]**	**CS [B]**	**CR [%]**	**CR [%]**
alice29.txt	152089	51477	33.85	47193.6	31.03	46965.0	30.88	0.15
asyoulik.txt	125179	45525	36.37	42860.0	34.24	42598.8	34.03	0.21
cp.html	24603	8600	34.96	8336.3	33.88	8306.0	33.76	0.12
fields.c	11150	3561	31.94	3381.1	30.32	3374.8	30.27	0.06
grammar.lsp	3721	1517	40.77	1450.1	38.97	1445.6	38.85	0.12
lcet10.txt	426754	140046	32.82	129455.3	30.33	129511.4	30.35	-0.01
plrabn12.txt	481861	183626	38.11	165352.4	34.32	164711.6	34.18	0.13
xargs.1	4227	2012	47.60	1941.0	45.92	1937.4	45.83	0.09
Average			37.05		34.88		34.77	0.11

All improvements are higher than 1% for all files, but the maximal improvement is almost 4%. Results for the Adaptive Huffman compression are different thanresults from the previous compression algorithms. The average improvement was more than 8% against 1-gram alphabet.

As may be seen, the improvement gained by Tabu Set is very slight. The largest improvement may be seen in Adaptive Huffman Encoding, but it is only 0.33 % in average. The achieved improvement for Burrows-Wheeler algorithm is only 0.11% in compression ratio and only 0.01% for LZW compression algorithm. These results show that using Tabu Sets has no big impact on the algorithm. Form the other point of view, it shows that the original algorithm was well designed and is able to find very good result. When we see the progress of the algorithm during optimization, the algorithm using TabuSearch is much more efficient. The convergence was much faster.

5.4 Results of LZSS for the Files from Canterbury Corpus

Results achieved by LZSS algorithm are depicted in Table 5. As may be seen, the average improvement in compression ratio (depicted in column Diff CR) is

Table 4. Results for Canterbury corpus and Adaptive Huffman compression algorithm

	Orig.	1-grams		Without TS		With TS		Diff.
FileName	**size [B]**	**CS [B]**	**CR [%]**	**CS [B]**	**CR [%]**	**CS [B]**	**CR [%]**	**CR [%]**
alice29.txt	152089	87914	57.80	76233.9	50.12	75870.4	49.89	0.24
asyoulik.txt	125179	76000	60.71	65925.2	52.66	65507.6	52.33	0.33
cp.html	24603	16434	66.80	14663.4	59.60	14492.0	58.90	0.70
fields.c	11150	7266	65.17	6254.7	56.10	6215.2	55.74	0.35
grammar.lsp	3721	2364	63.53	2151.2	57.81	2138.2	57.46	0.35
lcet10.txt	426754	250823	58.77	217507.2	50.97	216220.6	50.67	0.30
plrabn12.txt	481861	275857	57.25	238434.4	49.48	237399.8	49.27	0.21
xargs.1	4227	2796	66.15	2590.6	61.29	2582.8	61.10	0.18
Average			62.02		54.75		54.42	0.33

Table 5. Results for Canterbury corpus and LZSS algorithm

	Orig.	1-grams			With TS			Diff
FileName	**size [B]**	**CS [B]**	**CR [%]**	**AS**	**CS [B]**	**CR [%]**	**AS**	**CR [%]**
alice29.txt	152 089	61587	40.49	74	57238.0	37.63	325.0	2.86
asyoulik.txt	125 179	55804	44.58	68	51988.0	41.53	262.0	3.05
cp.html	24 603	8881	36.10	86	8700.7	35.36	125.3	0.73
fields.c	11 150	3584	32.14	90	3509.0	31.47	106.3	0.67
grammar.lsp	3 721	1413	37.97	76	1384.7	37.21	84.7	0.76
lcet10.txt	426 754	159978	37.49	84	146050.5	34.22	627.0	3.26
plrabn12.txt	481 861	220945	45.85	81	202647.0	42.06	489.0	3.80
xargs.1	4 227	1985	46.96	74	1955.5	46.26	83.0	0.70
Average			40.20			38.22		1.98

almost 2% which means that the algorithm works very well. Once again, it works better for longer files. The improvement of 3% was achieved for all four longer text files. When we look at Alphabet sizes, the algorithm use only few 2-grams for small files such as *xargs.1*, *grammar.lsp* and *fields.c* but it uses most of them for four larger files. These results show that border between character based alphabet and 2-grams based alphabet in not strict. It depends on the contents of the files, size of the files and many other features which confirm the meaning of this research.

6 Conclusion

This paper describes a new variant of an algorithm for optimization of compression alphabet. Basic algorithm is based on genetic algorithms. A new variant uses a Tabu Search method for improving the basic algorithm. The results show that usage of Tabu Search algorithm slightly improve the compression ratio. Moreover, it leads to faster convergence of the algorithm. Largest average improvement of 0.3% was achieved for Adaptive Huffman Encoding. The improvement

for Burrows Wheeler and LZW compression algorithm was smaller. The largest improvements were achieved for larger text files.

This paper also described results of alphabet optimization algorithm on LZSS compression algorithm. Results show that event for such complex compression algorithm should be improved using optimal alphabet. The average improvement was 2%, but it was more than 3% for all longer text files from the corpora.

Future work will be focused on evolving alphabet for group of files. Groups contain a text document written in some language like German or English, or it may contain documents from the same author, etc. Moreover, because the alphabet will be shared between all files in a group, it may contain more symbols. Another branch of future research will be focused on different optimization technique such as ant colony optimization, simulated annealing, etc. that may provide better results.

References

1. Abramson, N.: Information Theory and Coding. McGraw-Hill, New York (1963)
2. Andres, J.: On a conjecture about the fractal structure of language (2008) (preprint)
3. Arnold, R., Bell, T.: A corpus for the evaluation of lossless compression algorithms. In: Storer, J.A., Cohn, M. (eds.) Proc. 1997 IEEE Data Compression Conference, pp. 201–210. IEEE Computer Society Press, Los Alamitos (1997)
4. Bentley, J.L., Sleator, D.D., Tarjan, R.E., Wei, V.K.: A locally adaptive data compression scheme. Commun. ACM 29(4), 320–330 (1986)
5. Burrows, M., Wheeler, D.J.: A block-sorting lossless data compression algorithm. Tech. rep., Digital SRC Research Report (1994)
6. Cleary, J.G., Witten, I.H.: Data compression using adaptive coding and partial string matching. IEEE Transactions on Communications 32, 396–402 (1984)
7. Glover, F., McMillan, C.: The general employee scheduling problem: an integration of ms and ai. Comput. Oper. Res. 13, 563–573 (1986), http://dl.acm.org/citation.cfm?id=15310.15313
8. Huffman, D.A.: A method for the construction of minimum-redundancy codes. Institute of Radio Engineers 40(9), 1098–1101 (1952)
9. Isal, R.Y.K., Moffat, A.: Parsing strategies for bwt compression. In: DCC 2001: Proceedings of the Data Compression Conference, p. 429. IEEE Computer Society, Washington, DC (2001)
10. Koza, J.: Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Dept. of Computer Science, Stanford University (1990)
11. Kuthan, T., Lansky, J.: Genetic algorithms in syllable-based text compression. In: Pokorný, J., Snásel, V., Richta, K. (eds.) DATESO. CEUR Workshop Proceedings, vol. 235. CEUR-WS.org (2007)
12. Lansky, J., Chernik, K., Vlickova, Z.: Comparison of text models for bwt. In: DCC 2007: Proceedings of the 2007 Data Compression Conference, p. 389. IEEE Computer Society, Washington, DC (2007)
13. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)

14. Moffat, A.: Implementing the ppm data compression scheme. IEEE Transactions on Communications 38(11), 1917–1921 (1990)
15. Platos, J., Kromer, P.: Optimizing alphabet using genetic algorithms. In: 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011), pp. 498–503 (November 2011)
16. Platos, J., Kromer, P.: Reducing Alphabet Using Genetic Algorithms. In: Snasel, V., Platos, J., El-Qawasmeh, E. (eds.) ICDIPC 2011, Part II. CCIS, vol. 189, pp. 82–92. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-22410-2_7, doi:10.1007/978-3-642-22410-2_7
17. Platos, J., Kromer, P.: Reducing Alphabet Using Genetic Algorithms. In: Snasel, V., Platos, J., El-Qawasmeh, E. (eds.) ICDIPC 2011, Part II. CCIS, vol. 189, pp. 82–92. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-22410-2_7, doi:10.1007/978-3-642-22410-2_7
18. Rissanen, J.: Generalized kraft inequality and arithmetic coding. IBM Journal of Research and Development 20(3), 198–203 (1976)
19. Rissanen, J., Langdon Jr., G.G.: Arithmetic coding. IBM Journal of Research and Development 23(2), 149–162 (1979)
20. Salomon, D.: Data Compression - The Complete Reference, 4th edn. Springer-Verlag London Limited (2007)
21. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
22. Shannon, C.E.: Prediction and entropy of printed english. Bell Systems Technical Journal 30, 50–64 (1951)
23. Storer, J.A., Szymanski, T.G.: Data compression via textual substitution. Journal of the ACM 26(26/82), 928–951 (1982)
24. Üçoluk, G., Toroslu, I.H.: A genetic algorithm approach for verification of the syllable-based text compression technique. Journal of Information Science 23(5), 365–372 (1997), http://jis.sagepub.com/content/23/5/365.abstract
25. Welch, T.: A technique for high-performance data compression. Computer 17(6), 8–19 (1984)
26. Ziv, J., Lempel, A.: A universal algorithm for sequential data compression. IEEE Transactions on Information Theory IT-23(3), 337–343 (1977)
27. Ziv, J., Lempel, A.: Compression of individual sequences via variable-rate coding. IEEE Transactions on Information Theory IT-24(5), 530–536 (1978)

Rough Sets-Based Identification of Heart Valve Diseases Using Heart Sounds

Mostafa A. Salama[1], Aboul Ella Hassanien[2], Jan Platos[3], Aly A. Fahmy[2], and Vaclav Snasel[3]

[1] Department of Computer Science, British University in Egypt, Cairo, Egypt
mostafa.salama@gmail.com
[2] Cairo University, Faculty of Computers and Information, Cairo, Egypt
{aboitcairo,aly.fahmy}@gmail.com
[3] Faculty of Electrical Engendering and Computer Science, VSB-Technical University of Ostrava, Czech Republic
{jan.platos,vaclav.snasel@}vsb.cz

Abstract. Recently, heart sound signals have been used in the detection of the heart valve status and the identification of the heart valve disease. Heart sound data sets represents real life data that contains continuous and a large number of features that could be hardly classified by most of classification techniques. Feature reduction techniques should be applied prior applying data classifier to increase the classification accuracy results. This paper introduces the ability of rough set methodology to successfully classify heart sound diseases without the need applying feature selection. The capabilities of rough set in discrimination, feature reduction classification have proved their superior in classification of objects with very excellent accuracy results. The experimental results obtained, show that the overall classification accuracy offered by the employed rough set approach is high compared with other machine learning techniques including Support Vector Machine (SVM), Hidden Naive Bayesian network (HNB), Bayesian network (BN), Naive Bayesian tree (NBT), Decision tree (DT), Sequential minimal optimization (SMO).

Keywords: rought sets, heart sounds, identification, machine learning, feature reduction, classification.

1 Introduction

The diagnosis of diseases like heart valve diseases using data mining tools is an essential requirement in daily life. Most of heart valve diseases have an effect on the heart sound of patients [2]. Classification can be applied to detect whether the patient's heart sound signal is normal or not, and also can detect the type of the heart disease in sick patients. Such approach could be useful in the diagnosis of heart disease remotely, just by sending a record of the heart signals to a medical back-end system that replies automatically. Also it is considered as a low cost approach rather than the high cost medical examinations. A computerized system could provide physicians with suggestions about the diagnostic of the

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 667–676, 2012.

diseases. Due to the sensitivity of heart diagnosis results, a high classification accuracy and performance are required with the least error percentage. After extracting features from heart sound signals, preprocessing is applied on these features [3]. The most important preprocessing step is the feature reduction of the input data set. The data set contain features that are considered as noisy or irrelevant features, these features could have a negative impact on the classification accuracy of the instances - patients. Feature reduction methods are either feature extraction or feature selection method. Feature extraction method apply operation on the original features and extract a lower number of features that carries the same characteristics. Feature selection methods rank and select the most important features, where if only a subset of features with the highest rank is used in classification, a higher classification accuracy could be achieved.

The extracted heart sound data are three different data sets, each of 100 features where they are splitted into six different parts. The first data set is required to classify whether the heart of the patients is normal or not. The second and third data set is required for the detection of the heart valve disease. The heart valve diseases under investigation in this paper are aortic stenosis AS, aortic regurgitation AR, mitral stenosis MS and mitral regurgitation MR. This disease classification is performed in two steps where the first step is applied on the second data set for determining the type of the systolic murmur which means AS or MR, and the second step is applied on the third data set of a diastolic murmur diseases which means AR or MS. The second importance of feature selection method is to determine which stage of the heart sound could have the greatest indication to heart valve disease in the case of each murmur type. The fourth stages of a heart sound are the first heart signal S1, the systolic period, the second heart signal and the diastolic period [4].

When a rough sets discretization is applied on disease prediction from the heart disease data sets, the feature selection is not required due to the Reducts concept. The rough set produces the highest classification accuracy. Also the generated rules that are used in classification are also useful to detect some of the knowledge and facts that exist in the input data set. The rules generate from the rough set depends on the attributes selected by features like ChiMerge and information gain methods. The experimental study shows the results of applying rough set in classification with and without feature selection and also includes comparison between the classification results of different machine learning methods and rough set as a classifier. Finally, it shows the rules generated by rough set and how these rules depends, partially, on features selected by feature selection methods [5,1].

The rest of this paper is organized as follows: Section 2 reviews the basic concepts of the heart sound valve diseases and heart sound signals data characteristics. Section 3 discusses in details the proposed system and its phases including the pre-processing, analysis and rule generating and identification and prediction. The experimental results and conclusions are presented in Section 4.

2 The Heart Sound Signals: Data Collection and Its Characteristics

A lot of research have been applied on heart sound for the detection of heart valve disease. Features are extracted from the heart sound signal into a data set that is composed of a number of features. The processing phases result in a heart sound feature vector consisting of 100 components for each signal. These extracted features represent the four stages of a heart signal which are $S1$ signal, systolic period, $S2$ signal and diastolic period as shown in Fig. 1. These features are divided into six groups as follows:

- F1:F4 are the standard deviation of all heart sounds, $S1$, $S2$ and average heart rate.
- F5:F12 represents signal $S1$.
- F13:F36 represents the systolic period.
- F37:F44 represents signal $S2$.
- F45:F92 represents the diastolic period.
- F93:F100 the four stage of a heart signal are passed from four band-pass frequency filters. The energy of each output is calculated to form these last 8 features.

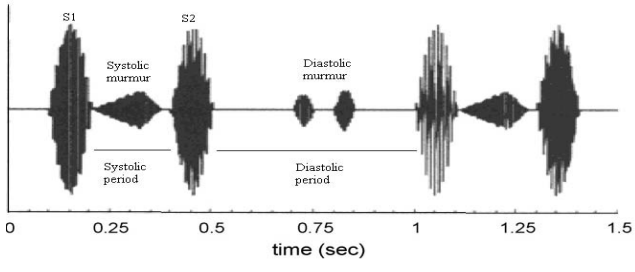


Fig. 1. Heart signal: systolic period and diastolic period [4]

3 The Proposed Rough Set-Based Identification of Heart Valve Diseases Approach

Fig. 2 illustrates the overall steps of the proposed identification of heart valve diseases system. It is composed of three consecutive phases, which are elaborated in the following subsections. These phases are:

- **Pre-processing phase**. This phase includes tasks such as extra variables addition and computation, decision classes assignments, data cleaning, completeness, correctness, feature creation, feature selection, feature evaluation and discretization.

– **Analysis and rule generating phase**. This phase includes the generation of preliminary knowledge, such as computation of object reducts from data, derivation of rules from reducts, rule evaluation and prediction processes.
– **Identification and prediction phase**. This phase utilizes the rules generated from the previous phase to predict the hart diseases.

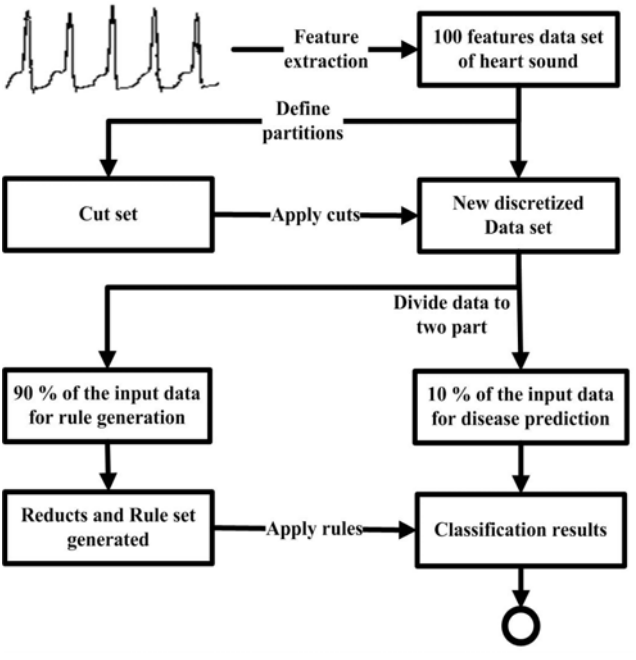


Fig. 2. Rough set-based identification of heart valve diseases system

3.1 Pre-processing Phase

Feature Reduction. In the proposed approach, two different feature reduction methods will be applied and both will give a value to the identification of the heart valve disease. The first method will depend on ChiMerge feature selection [6,7,8] for the ranking of features according to their relevance to the class labels. The reason of using this method is the nature of extracted features of heart sound, which represents a continuous feature. Nearly most of the feature selection method discretize the continuous feature which could leads to the distortion of data and loose of its characteristics and then decrease feature classification. ChiMerge method determines the Chi-Square χ^2 value while performs the discrimination of features which leads to more accurate results. The second method is a feature reduction using deep belief network [6]. The deep belief network is applied in a novel way in order to generate a new data set of a reduced number of features according the partition of the heart signal discussed in the previous section. An ChiMerge feature evaluation and deep belief network will be used in the proposed model.

Discretization Based on RSBR. A real world data set, like medical data sets, contains mixed types of data including continuous and discrete data. The discretization process divides the attributes value into intervals [9]. The discretization based on RS and Boolean Reasoning (RSBR) shows the best results in the case of heart disease data set. In the discretization of a decision table $S = (U, A \bigcap \{d\})$, where U is a non-empty finite set of objects and A is a non-empty finite set of attributes. And $V_a = [x_a, x_a)$ is an interval of real values x_a, w_a in attribute a. A partitioning P_a of V_a for any $a \in A$ is required. Any partition of V_a is defined by a sequence of the so-called cuts $x_1 < x_2 < .. < x_k$ from V_a. The main steps of the RSBR discretization algorithm are provided in algorithm 1.

Algorithm 1. RSBR discretization algorithm

Input: Information system table (S) with real valued attributes A_{ij} and n is the number of intervales for each attribute.
Output: Information table (ST) with discretized real valued attribute

1: **for** $A_{ij} \in S$ **do**
2: Define a set of boolean variables as follows:

$$B = \{\sum_{i=1}^{n} C_{ai}, \sum_{i=1}^{n} C_{bi} \sum_{i=1}^{n} C_{ci}, ..., \sum_{i=1}^{n} C_{ni}\} \tag{1}$$

3: **end for**
Where $\sum_{i=1}^{n} C_{ai}$ correspond to a set of intervals defined on the attributes a
4: Create a new information table S_{new} by using the set of intervals C_{ai}
5: Find the minimal subset of C_{ai} that discerns all the objects in the decision class D using the following formula:

$$\Upsilon^u = \wedge\{\Phi(i,j) : d(x_i \neq d(x_j)\} \tag{2}$$

Where $\Phi(i,j)$ is the number of minimal cuts that must be used to discern two different instances x_i and x_j in the information table.

3.2 Analysis and Rule Generating Phase

Unseen instances are considered in the discovery process, and the uncertainty of a rule, including its ability to predict possible instances, can be explicitly represented in the strength of the rule. The quality of rules is related to the corresponding reduct(s). We are especially interested in generating rules which cover largest parts of the universe U. Covering U with more general rules implies smaller size of a rule set. The main steps of the RSBR discretization algorithm are provided in algorithm 2.

Algorithm 2. Rule generation and classification

Input: reduct sets $R_{final} = \{r_1 \cup r_2 \cup .... \cup r_n\}$
Output: Set of rules

```
for each reduct r do
  for each corresponding object x do
    Contract the decision rule (c_1 = v_1 ∧ c_2 = v_2 ∧ .... ∧ c_n = v_n) ⟶ d = u
    Scan the reduct r over an object x
    Construct (c_i, 1 ≤ i ≤ n)
    for every c ∈ C do
      Assign the value v to the corresponding attribute a
    end for
    Construct a decision attribute d
    Assign the value u to the corresponding decision attribute d
  end for
end for
```

Identification and Prediction Phase. Classification and prediction are the last phases of our proposed approach. To transform a reduct into a rule, one only has to bind the condition feature values of the object class from which the reduct originated to the corresponding features of the reduct. Then, to complete the rule, a decision part comprising the resulting part of the rule is added. This is done in the same way as for the condition features. To classify objects which have never been seen before, the rules generated from a training set will be used. These rules represent the actual classifier. This classifier is used to predict to which classes the new objects are attached. The nearest matching rule is determined as the one whose condition part differs from the feature vector of the re-object by the minimum number of features. When there is more than one matching rule, we use a voting mechanism to choose the decision value. Every matched rule contributes with votes to its decision value, which are equal to the t times number of the objects matched by the rule. The votes are added and the decision with the largest number of votes is chosen as correct class. Quality measures associated with decision rules can be used to eliminate some of the decision rules. The global strength defined in [10] for rule negotiation is a rational number in [0, 1] representing the importance of the sets of decision rules relative to the considered tested object.

Let us assume that $T = (U; A \bigcup d)$ is a given decision table, u_t is a test object, $Rul(X_j)$ is the set of all calculated basic decision rules for T, classifying objects to the decision class $X_j(v_d^j = v_d), MRul(X_j; u_t) \subseteq Rul(X_j)$ is the set of all decision rules from $Rul(X_j)$ matching tested object u_t. The global strength of the decision rule set $MRul(X_j; u_t)$ is defined as given in [10]. The measure of the strengths of the rules defined above is applied in constructing classification algorithm. To classify a new case, matching rules are selected to the new object. The strength of the selected rule sets is calculated for any decision class, and then the decision class with maximal strength is selected with the new object being classified to this class.

4 Experimental Results and Discussion

4.1 The Heart Sound: Data Sets Characteristics

The identification of heart valve diseases proposed system were applied on three different data sets of heart sound signals with number of instances in every class. The first data set "HS_AS_MR" consists of systolic diseases where it contains 37 instances of aortic stenosis AS cases and 37 instances of mitral regurgitation MR cases. The second data set "HS_AR_MS" constains diastolic diseases where it contains 37 instances of aortic regurgitation AR cases and 37 instances of mitral stenosis MS cases. The third data set "HS_N_S" contains 64 instance, where 32 instances represent healthy patients and the other 32 represents unhealthy, murmur diseased patients.

4.2 Comparison between Different Classifiers and the Proposed Algorithm

Table 1 illustrates that the overall rough sets classification accuracy in terms of sensitivity and specificity compared with Hidden Naive Bayesian network (HNB), Bayesian network (BN), Naive Bayesian tree (NBT), Decision tree (DT) and Sequential minimal optimization (SMO). Empirical results reveal that the proposed rough set approach performs better than the other classifiers.

4.3 Analysis on the Generated Reducts to Explain the Results

Now lets take a deep look to the reasons the leads to such good results. For each data set, we reach the minimal number of reducts that contains a combination of attributes which has the same discrimination factor for each data set. Table 2 shows the final generated reducts for each data set, which are used to generate the list of the rules for the classification.

Table 1. Accuracy results: Comparative analysis among Hidden Naive Bayesian network (HNB), Bayesian network (BN), Naive Bayesian tree (NBT), Decision tree (DT), Sequential minimal optimization (SMO)

Classifier	HS_AS_MR	HS_AR_MS	HS_N_S
SVM	90.54	90.54	85.93
HNB	90.54	91.89	**90.625**
BN	86.48	83.78	84.37
DT	89.18	83.78	82.81
SMO	**93.24**	94.59	87.50
NBT	87.83	89.18	89.06
RS	**92.90**	**97.1**	**90.0**

Table 2. Rough reducts sets of the three data sets

Data type	Reduct sets
HS_AR_MS	3, 8, 31, 38,82
HS_AS_MR	3, 6, 36,39
HS_N_S	1, 9, 87, 94, 97

In order to evaluate the proposed rough sets classifier, lets discuss the reducts in comparison to the feature selection using Chimerge approach and find the results of the classifiers after feature selection for each data set.

HS_AR_MS. The selected features by the Chimerge approach are ordered as follows:
{F32,{**F31**},F30,F29,F100,F28,F33,F27,{**F3**},F5,
F4,F49,F48,F45,F46,F50,F25,F26}.

The classification accuracy of DT and SVM after feature selection are 83.9 and 92.0, respectively, where these results are still less than that of rough set accuracy results as shown in Table 1. Also, it noticed that the selected features includes feature $F36$ and feature $F3$, where feature $F36$ is the most important features.

HS_AS_MR. The selected features by the Chimerge approach are ordered as follows:
{{**F36**},F13,F12,F14,F15,F35,F4,F2,F18,
F17,F11,{**F3**},F92,F5,F16,F20,F19,F23,F21,F93,
F95,F10,F1,F94,F24,F28,F37,F25,F41,F29,F100,
F54,F22,F40,F26,F88,F55,F96}.
The classification accuracy of DT and SVM after feature selection are 89.0 and 94.5 respectively where these results are still less than that of rough set accuracy results as shown in Table (4). Also, it noticed that the selected features includes feature $F31$ and feature $F3$, where feature $F31$ is the second most important features.

HS_N_S. The selected features using the Chimerge approach are ordered follows:
{F54,F58,F53,F65,F64,F67,F59,{**F97**},F61,F70,F96,
F98,F55,F60,F66,F51,F52,F49,F62,F63,F23,F22,
F45,F50,F2,F56,F57,F71,F28,F27,F24,F26,F25,
F47,F46,F99,F21,F72,F68,F69,F3,F75,F73,F92,F48,
{**F87**},F7,F76}. The classification accuracy of DT and SVM after feature selection are 83.3 and 90.9 respectively where these results are still less than that of rough set accuracy results as shown in Table 1. Also, it noticed that the selected features includes feature $F97$ which is element in the Reduct set.

5 Conclusions

Heart sound data sets represents a real life data that contains continuous attributes and a large number of features that could be hardly classified by most of classification techniques. This paper introduces the ability of rough set theory to successfully classify heart sound diseases without the need applying feature selection. Discretizing the raw heart sound data and applying a feature reduction approach should be applied prior any classifier to increase the classification and prediction accuracy results. The experimental results obtained, show that the overall classification accuracy offered by the employed rough set approach is high compared with other machine learning techniques including Support Vector Machine (SVM), Hidden Naive Bayesian network (HNB), Bayesian network (BN), Naive Bayesian tree (NBT), Decision tree (DT), Sequential minimal optimization (SMO).

Acknowledgment. We would like to thanks Dr. Ilias Maglogiannis, University of Central Greece, Department of Computer Science and Biomedical Informatics, Lamia, Greece for providing us the hear sound signal data.

References

1. Salama, M.A., Soliman, O.S., Maglogiannis, I., Hassanien, A.E., Fahmy, A.A.: Rough set-based identification of heart valve diseases using heart sounds. To be published on Special volume(s) in Series: Intelligent Systems Reference Library dedicated to the memory of Professor Zdzislaw Pawlak (2011)
2. Chen, T., Kuan, K., Celi, L., Clifford, G.: Intelligent heartsound diagnostics on a cell phone using a hands-free kit. In: Proceeding of AAAI Artificial Intelligence for Development, AI-D 2010 (2010)
3. Kumar, D., Carvalho, P., Antunes, M., Paiva, R.P., Henriques, J.: Heart murmur classification with feature selection. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, pp. 4566–4569 (September 2010)
4. Maglogiannis, I., Loukis, E., Zafiropoulos, E., Stasis, A.: Support vectors machine-based identification of heart valve diseases using heart sounds. Comput. Methods Programs Biomed., 47–61 (March 6, 2009)
5. Janecek, A.G.K., Gansterer, W.N., Demel, M., Ecker, G.F.: On the relationship between feature selection and classification accuracy. In: JMLR: Workshop and Conference Proceedings, vol. 4, pp. 90–105 (2008)
6. Salama, M.A., Hassanien, A.E., Fahmy, A.A.: Uni-class pattern-based classification model. In: Proceeding of the 10th IEEE International Conference on Intelligent Systems Design and Applications, Cairo, Egypt (December 2010)
7. Liu, H., Setiono, R.: Chi2: feature selection and discretization of numeric attributes. In: Proceedings of the Seventh International Conference on Tools with Artificial Intelligence, Virginia, USA, November 8, p. 388 (1995)

8. Liu, H., Setiono, R.: Feature selection via discretization. IEEE Transactions on Knowledge and Data Engineering 9, 642–645 (1997)
9. Chao, S., Li, Y.: Multivariate interdependent discretization for continuous attribute. In: Proceeding of the 3rd International Conference on Information Technology and Applications, vol. 1, pp. 167–172 (2005)
10. Al-Qaheri, H., Hassanien, A.E., Abraham, A.: A Generic Scheme for Generating Prediction Rules Using Rough Sets. In: Rough Set Theory: A True Landmark in Data Analysis. SCI, vol. 174, pp. 163–186. Springer, Heidelberg (2009)

A Novel Hybrid Intelligent Classifier to Obtain the Controller Tuning Parameters for Temperature Control

José Luis Calvo-Rolle[1], Emilio Corchado[2], Héctor Quintian-Pardo[2], Ramón Ferreiro García[1], Jesús Ángel Román[2], and Pedro Antonio Hernández[3]

[1] Department de Ingeniería Industrial, Universidad de La Coruña
Avda. 19 de febrero, s/n, 15405, Ferrol, A Coruña, Spain
jlcalvo@cdf.udc.es, ferreiro@udc.es
[2] Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
{escorchado,hector.quintian,zjarg}@usal.es
[3] Departamento de Expresión Gráfica en la Ingeniería, Universidad de Salamanca
Av. Requejo, 33 Campus Viriato, 49022, Zamora, Spain
pedrohde@usal.es

Abstract. This study presents a novel hybrid classifier method to obtain the best parameters of a PID controller for desired specifications. The study presents a hybrid system based on the organization of existing rules and classifier models that select the optimal expressions to improve specifications. The model achieved chooses the best controller parameters among different closed loop tuning methods. The classifiers are based on ANN and SVM. The proposal was tested on the temperature control of a laboratory stove.

Keywords: Hybrid classifier, PID, closed-loop tuning, intelligent control.

1 Introduction

This study describes a hybrid classifier to obtain the controller parameters based on PID (Proportional-Integral-Derivative) closed-loop tuning. Although the PID controller is one of the most traditional types of controller, researchers are still working to improve its behaviour and performance [1-11]. There have been several studies with the same objective, but they have always been oriented to a specific system [5, 9, 11].

Nevertheless, there are many controllers operating well below the optimal state [14], overcoat controllers that are not self-tuning. It has therefore become critical to achieve new ways to solve this problem. Many studies related to the PID controller try to either establish optimal parameters according to the plant, or achieve self-tuning controller topologies [11, 12, 13].

The proposed topology described in this research has two phases. The first phase obtains characteristics of the plant response, while the second is applied to achieve the controller parameters by means of a hybrid classifier. The proposal makes it possible to achieve an intelligent topology with satisfactory results. The proposed topology only

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 677–689, 2012.

contemplates techniques with hard and previously tested implantation in the industry. One of the aims of the implementation of the hybrid classifier is to contemplate the largest number of possibilities.

This study is organized as follows: section 2 provides a brief description of the general model; section 3 describes the tuning controller topology and briefly reviews the PID controller tuning in a closed-loop. Section 4 describes the hybrid classifier, section 5 presents empirical verification, and finally, section 6 provides conclusions and suggests future works.

2 Steps to Obtain the Best Plant Controller Parameters

The procedure to obtain the best plant controller parameters to improve a given specification is illustrated in figure 1.

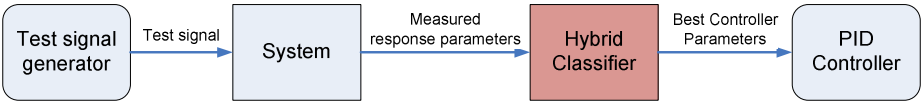


Fig. 1. Flowchart to obtain controller parameters

As shown in figure 1, the first step in obtaining the best combination of the plant controller parameters involves a test signal generator. The test signal is then applied to the system. The next step is to measure the response characteristics. The best combination of the plant controller parameters is obtained by using the characteristics as input to the hybrid classifier, which supplies the controller with its parameters. In general terms, the hybrid classifier can be represented by the illustration in figure 2, where models based on rules [14-19] and machine learning techniques, such as Artificial Neural Networks (ANN) [24, 25] and Support Vector Machine [21, 22], are applied. After comparing the results, the best ones are chosen.

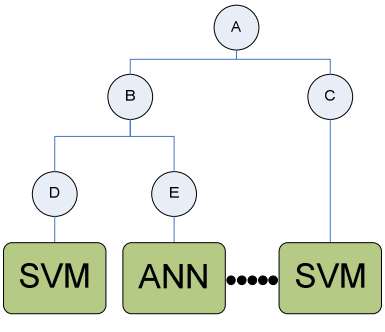


Fig. 2. Model final structure

3 Tuning Controller Topology Based on Hybrid Classifier

The topology proposed in this study is shown in figure 3. The following subsections describe the different aspects of the proposed topology.

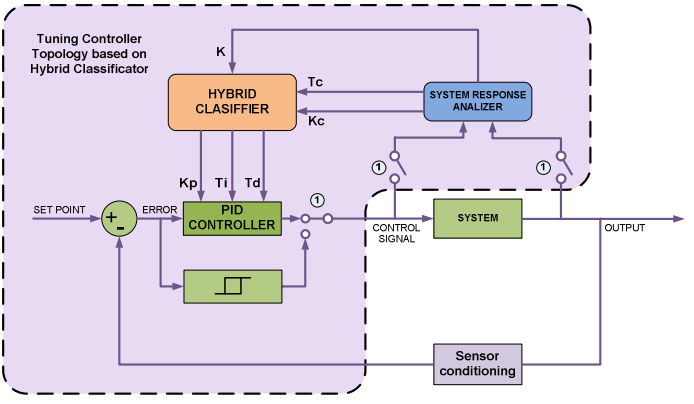


Fig. 3. Tuning controller topology

3.1 PID Controller Format

There are several topologies for PID controllers, but in this study the standard format represented in equation 1 is used [14, 15].

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right] \tag{1}$$

where '*u*' is the control variable and '*e*' is the control error given by '$e = SP - y$' (the difference between the set point '*SP*' and conditioned output '*y*'). The other terms are the tuning controller parameters: proportional gain '*K*', integral gain '*Ti*' and derivate gain '*Td*'.

3.2 PID Controller Tuning in Closed-Loop

General Procedure to Calculate Parameters. Two steps are necessary to obtain the controller tuning parameters in a closed loop:

- It is first necessary to set the system response to a permanent state of oscillation. Certain characteristics of the response must then be measured.
- According to the information gathered from the plant response, appropriate expressions must be applied to obtain correct controller parameters for the desired specifications.

Obtaining Response Characteristics in a Closed-Loop. Different methods can be used to obtain the controller parameter conditions. The present study uses the relay-feedback method proposed by Aström and Hägglud [14]. The results are very similar to those obtained by the traditional method proposed by Ziegler-Nichols [16]; however, the former offers some very important advantages, such as:

- The system operation is not nearly as unstable.
- The tuning process can be carried out at any time for any working point.

The implementation scheme of relay feedback is shown in figure 3 (switches '1' to another position). A relay with hysteresis centred on a zero value with an amplitude *d* and a hysteresis window of width *h* is recommended for the general method.

The system oscillation has a period (*Tc*) with approximately the same as the Ziegler-Nichols method. The critical gain (*Kc*) of the process is obtained with equation 2, where *a* is the peak-to-peak value of the oscillation.

$$K_C = \frac{4d}{\pi\sqrt{a^2 - h^2}} \tag{2}$$

Obtaining Controller Parameters with Formulas. After obtaining the *Tc* and *Kc* from the previous step, the controller parameters can be calculated. Many expressions have been developed by different authors [14, 16-19] with the aim, among others, of:

- Improving a particular specification of the system controlled response.
- Making the system robust to a particular criteria (Load Disturbance or Set Point Criteria)

There are several studies [14-19] that have developed different expressions. Even control equipment manufacturers have developed their own expressions according to their products line.

In this study, four methods (table 1) were taken into account: Ziegler-Nichols, Ziegler-Nichols some overshoot, Ziegler-Nichols without overshoot, and Tyreus-Luyben [16-19]. All of them are for Load Disturbance rejection criteria.

Table 1. Expressions of Controller parameters

	Kp	Ti	Td	Appl. Range
Z-N	0.6 x Kc	0.5 x Tc	0.125 x Tc	2<k·Kc<20
Z-N Some Ov.	0.33 x Kc	Tc / 2	Tc / 3	2<k·Kc<20
Z-N Whitout Ov	0.2 x Kc	Tc	Tc / 3	2<k·Kc<20
Tyreus-Luyben	0.45 x Kc	2.2 x Tc	Tc / 6.3	2<k·Kc<20

4 Hybrid Classifier Proposal

The proposed hybrid classifier is a fusion of rules and intelligent classification techniques. It can be divided into two different blocks:

- Knowledge of existing rules (1st block). The aim of this block is to organize different rules for application ranges, authors expressions, criterions, and so forth.
- Intelligent classifiers (2nd block). This part of the model selects the most appropriate expressions to obtain the controller tuning parameters.

In general terms, this novel hybrid intelligent classifier selects the best tuning parameters, according to the system and the desired specifications of operation.

The next two subsections describe the hybrid model. The first shows the flowchart used to select the intelligent classifier. The second provides details of the classifiers.

4.1 Flowchart of Knowledge of Existing Rules

After applying different methodologies of PID controller tuning in closed-loop, it is possible to obtain a flow-chart, as shown in figure 4. Many PID tuning rules in closed loop were taken into consideration to create this diagram, with the aim of achieving a generalized knowledge of the field. The following paragraphs explain the diagram in greater detail.

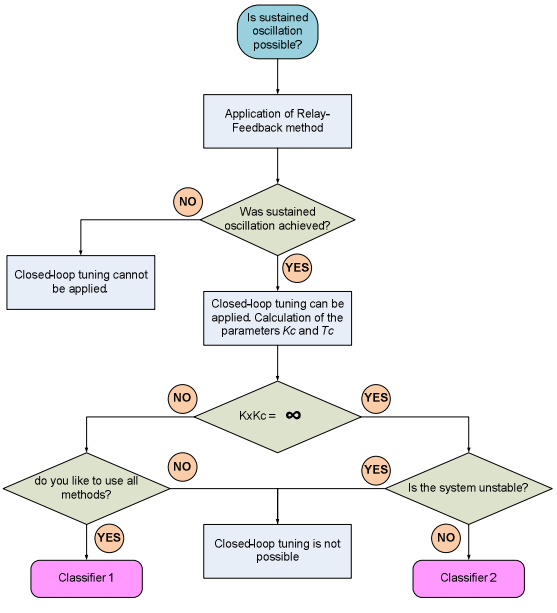


Fig. 4. Flowchart of knowledge of existing rules

The flowchart is based on the premise that the plant engineer can set the system operation in sustained oscillation. As explained in section 3 there are several techniques to perform PID controller tuning in closed loop. The Relay-Feedback method is the most robust, making it possible to achieve better results. It is then necessary to determine if the system can achieve the sustained oscillation. If it is not possible, then this method cannot be applied to tuning in closed loop. Otherwise, with the system in this state it is necessary to calculate *Tc* and *Kc* parameters.

With *Tc* and *Kc* values it is possible to obtain the *K.Kc* indicator, where *K* is the gain of the process. This indicator, among other functions, defines the best expressions to achieve the desired system response.

The model must be able to know if the value of *K.Kc* is infinite or not. It is then possible to follow the flowchart in two ways:

- When *K.Kc* is not infinite, the operators must decide if they want to use all the expressions anyway. If this is not the case, the closed loop tuning is not applicable for the contemplated expressions; otherwise, it is possible to use the Classifier 1.
- When *K.Kc* is infinite, the operator must find out if the system is unstable. If it is unstable, closed loop tuning is not applicable. This means that the system has an integrator in its transfer function. It is then possible to use the Classifier 2.

4.2 Classifiers to COMPLETE the Model

As shown in figure 4, with an organization of set rules, there are two blocks corresponding to the intelligent classifiers (1 and 2). Three techniques were applied to create these blocks: decision tree, artificial neural networks (ANN) and support vector machines (SVM). The following paragraphs describe the formulation of the model.

Model Input. As seen in the flowchart of figure 4 and its description, *K,Kc* is a very important indicator. It defines, for instance, the application range of expressions. In all classifiers that were created, *K.Kc* was the input defining the system type, the system dynamics and, consequently, its controllability.

Dataset for Model Creation. As in other studies, it is necessary to select representative systems with the objective of generalizing the model as much as possible. Consequently, this study is based on [20]. This research includes a list of very representative systems, where all real systems behave in a similar fashion.

There is a problem with the initial dataset, which is that the *K.Kc* values of the initial systems have very close values; however the last systems have widely separated values. For this reason, many systems were created to solve the problem. To this end, a delay time is added to the 1th and 2nd order systems with Pade approximation [14]. Thus, a difference in the *K.Kc* values between consecutive systems is achieved, less than that of the unit. Finally 1704 systems were obtained to implement the classifiers. The systems obtained are balanced by controller parameter expressions.

Systems Specifications for Each Expression. Each system is tested with the four expressions contemplated in table 1. Four specifications are then tested for system response to step input: response time (*Tr*), settling time (*Ts*), overshoot (*Ov*) and peak time (*Tp*). As a result, it is possible to obtain the expression that gives the best specification value. The present study used the following tuning methods: *Z&N* (Ziegler-Nichols), *Z&N wOv* (Ziegler-Nichols Whitout Overshoot), *Z&N sOv* (Ziegler-Nichols some Overshoot), *T&L* (Tyreus-Luyven).

Classification Techniques Analyzed to Complete the Model. Three techniques were taken into account to complete the model:

- Decision tree using the J48 learning algorithm: One of the classification methods contemplated in this research is the decision tree [23-25]. The decision tree approach is one of the most common approaches in automatic learning and decision making. The true purpose of decision trees is to classify the data into different groups, according to the dependent variable [23]. The decision trees were obtained by using the J48 algorithm [23, 26, 27]. The J48 algorithm was chosen because of its superior performance in most circumstances [27].
- Multilayer Perceptron (MLP): A multilayer perceptron is a feed forward artificial neural network [28]. It is one of the most typical ANNs due to its robustness and relatively simple structure. However the ANN architecture must be well selected to obtain good results.
- Support Vector Machine (SVM): is a concept used in statistics and computer science for a set of related supervised learning methods that analyze data and recognize patterns. It is used for classification and regression analysis [21, 22] and trains a classifier by finding an optimal separating hyperplane which maximizes the margin between classes of data in a kernel-induced feature space [21, 22].

Classification Results. For each specification (response time, settling time, overshoot and peak time) a classifier was created. Each set of expressions was categorized as follows:

- Class A: Z-N.
- Class B: Z-N Some Ov.
- Class C: Z-N Whitout Ov
- Class D: Tyreus-Luyben

Five different parameters were used to measure performance: Sensitivity (SE), Specificity (SPC), Positive Prediction Value (PPV), Negative Prediction Value (NPV) and Accuracy (ACC) (see equations from 3 to 7 respectively).

$$SE = \frac{TP}{(TP + FN)} \tag{3}$$

$$SPC = \frac{TN}{(FP + TN)} \tag{4}$$

$$PPV = \frac{TP}{(TP + FP)} \tag{5}$$

$$NPV = \frac{TN}{(TN + FN)} \tag{6}$$

$$ACC = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{7}$$

where TP is the number of True Positive, TN is the number of True Negative, FN is the number of False Negative and FP is the number of False Positive.

Table 2. Percentage of correct classification for two Classifier Models

Model data	Training data	Response Time (Tr)			Overshoot (Ov)		
		J48	MLP	SVM	J48	MLP	SVM
Classifier 1	1704	81	93	**95**	83	92	**94**
Classifier 2	1704	78	94	**96**	86	**95**	93
Model data	Training data	Settling Time (Ts)			Peak Time (Tp)		
		J48	MLP	SVM	J48	MLP	SVM
Classifier 1	1704	87	**94**	91	79	91	**93**
Classifier 2	1704	83	**94**	93	80	93	**95**

Table 3. Confusion Matrix for classification rate of Ov of Classifier 2 using MLP

	Method chosen by model			
Desired Method	A	B	C	D
A	**399**	9	15	3
B	9	**401**	6	10
C	10	18	**383**	15
D	17	12	19	**378**
TP	399	401	383	378
TN	1242	1239	1238	1250
FP	36	39	40	28
FN	27	25	43	48
SE	0,937	0,941	0,899	0,887
SPC	0,972	0,969	0,969	0,978
PPV	0,917	0,911	0,905	0,931
NPV	0,979	0,980	0,966	0,963
ACC	0,963	0,962	0,951	0,955

In all cases k-fold cross-validation was used to split the dataset into a reasonable value that obtains good results. The k value is 10 for all models. For the decision tree classification the algorithm chosen was J48, which has the following configuration parameter values: 0.25 for the confidence threshold for pruning, and 2 for the minimum number of instances per leaf. In the case of MLP, tests were performed with

2 and 3 hidden layers, with the second value providing the best results. The number of neurons in hidden layers is within the range of 10-15. The activation functions tested in the hidden layer were: log sigmoid and tangent sigmoid. The tangent function achieved the best results. The activation function of the output layer is the log sigmoid. The Winner Take All (WTA) technique was used to obtain the class provided by the MLP output. For SVM the selected kernel was the Gaussian radial basis function. For this technique, 15 different values were assigned for parameter γ (in a range from 2^{-12} to 2^{3}) and 17 different values for parameter C (from 2^{-5} to 2^{12}). A total of 255 (15x17) different combinations of parameters were taken into account.

Table 2 shows the percentage of correct classification using the previously mentioned techniques for the two classifiers. In each case, the selected classifier is the one that achieves the best percentage of correct classification (table 2 values in bold).

For all the cases considered in table 2, the best configuration for each technique used was selected. The confusion matrix was created in each case. An example of confusion matrix is shown in table 3 where Overshoot (Ov) is tested for Multi Layer Perceptron case.

5 Empirical Verification with a Physical Plant

An empirical verification of the Hybrid Classifier presented in this study was performed at a laboratory plant (figure 5) in which the temperature is controlled by adjusting the power provided to the heater element inside.

5.1 The Physical Description

The temperature variable depends on the following parameters: T1(t) is the temperature measured outside the stove; V is the air volume in the stove; SP(t) is the set point for the desired temperature; T2(t) is the measured temperature in the recipient; u(t) is the signal control to operate the heating element; Kv and Kt are constants related to the features of the heating element properties and the temperature sensor respectively.

5.2 Implementation of the Control

The test was performed in the Labview® environment. For operations at the plant, a National Instruments data acquisition card (model USB-6008 12-bit 10 KS /s Multifunction I/O) was chosen. The diagram of the process is implemented in Labview® editor with the control Scheme shown in figure 6. Different gain blocks were added to adapt signals to all operation range.

It was necessary to add a filter block with an edge frequency of 1.5 rad/sec (9.5Hz) in order to reduce noise from the analog input. Using a switch it is possible to select either a PID control or Relay-Feedback configuration. Figure 7 shows the internal implementation of a PID block. PID controller gains (Ti, Td, and Kp) are programmed manually.

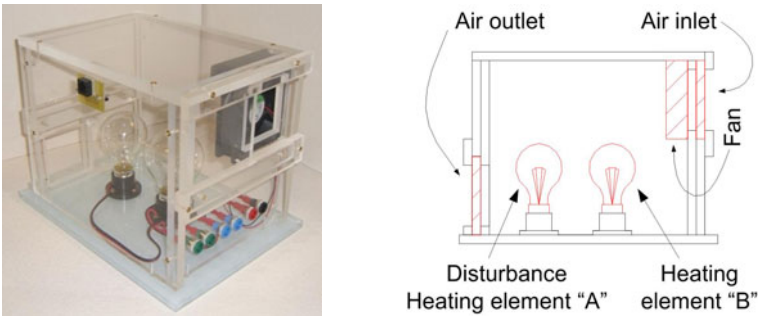


Fig. 5. Photograph and scheme of the real plant

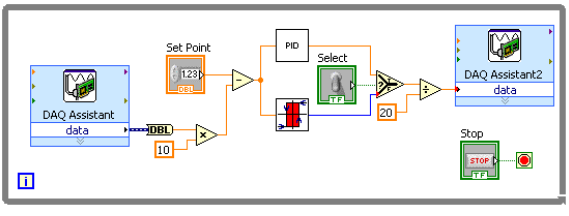


Fig. 6. Control scheme implemented in Simulink

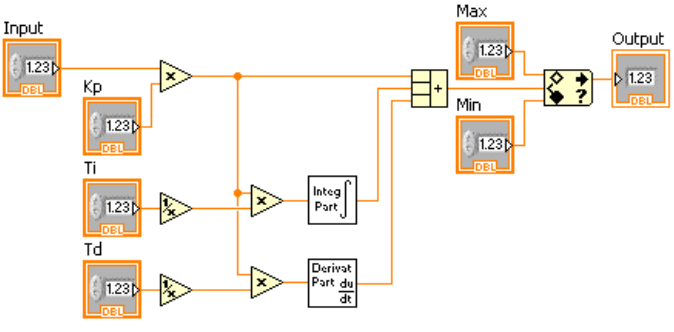


Fig. 7. PID block internal scheme

5.3 Practical Behavior of Hybrid Classifier Application onto the Plant

Some tests were performed on the physical system previously described in section 5.1, with the aim of checking the behavior of the novel Hybrid classifier model.

If the hybrid classifier is applied to the physical system with a set point of 35 ºC and the temperature outside is 20 ºC, the expressions chosen are Tyreus-Luyben. The system response with a step input for the working point of the design is shown in figure 8. Figure 9 shows the response with a step input for a traditional Ziegler-Nichols method in closed loop. As seen in these figures, the hybrid classifier model takes the best expressions contemplated in this study to achieve the best overshoot. Perturbations are introduced in both cases, first with the fan and then with the heating element "A", both of which are included in the stove. Comparing figures 8 and 9, it is apparent that the first method (from the hybrid classifier) is more robust.

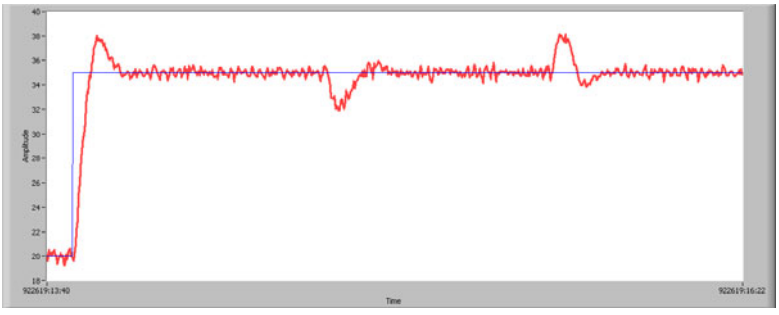


Fig. 8. System response with method from Hybrid Classifier

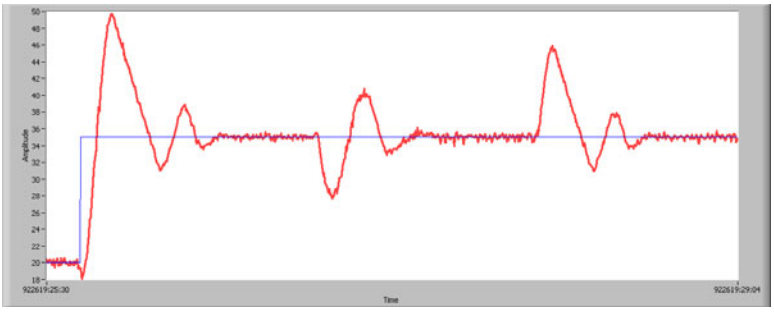


Fig. 9. System response with traditional Ziegler-Nichols method in closed-loop

6 Conclusions

A Hybrid Classifier was presented in this study, and tested with a practical application where a PID controller was used. The best advantage offered by the method is that it ensures the most appropriate selection of the contemplated expressions in order to obtain the PID controller parameters.

It should be noted that the novel model is easy to expand to other system types, such as: level control, pressure control, humidity control, and so forth. At that point, it is only necessary to contemplate the new systems in order to create the Hybrid Classifiers. The rest of the model is completely valid.

With the aim of obtaining the best results, more than one technique was taken into account. The final classifier, among the typical rules of PID controller, includes two techniques in the final stage (ANN and SVM), according to where they achieve the best results.

Several future research lines will be considered, two of which are of particular importance. Firstly it is necessary to consider the responses of the physical plant in creating the hybrid classifier. The other important future research line is to perform tests with real industrial plants and include them in the adaptive mechanism model in order to create a more versatile the tool.

Acknowledgments. This research is partially supported by projects TIN2010-21272-C02-01 from the Spanish Ministry of Science and Innovation. The authors would also like to thank the manufacturer of components for vehicle interiors, Grupo Antolin Ingeniería, S.A. which provided support through MAGNO 2008 – 1028 – CENIT funded by the Spanish Ministry of Science and Innovation.

References

1. Hsu, C., Chen, G., Lee, T.: Robust intelligent tracking control with PID-type learning algorithm. Neurocomputing 71, 234–243 (2007)
2. Gottwald, S.: Mathematical Fuzzy Control. A Survey of Some Recent Results. Logic Journal of IGPL 13, 525–541 (2005)
3. Zhang, J., Zhuang, J., Du, H., Wang, S.: Self-organizing genetic algorithm based tuning of PID controllers. Information Sciences 179, 1007–1018 (2009)
4. Juang, Y., Chang, Y., Huang, C.: Design of fuzzy PID controllers using modified triangular membership functions. Information Sciences 178, 1325–1333 (2008)
5. Liu, H., Coghill, G.: A model-based approach to robot fault diagnosis. Knowledge-Based Systems 18, 225–233 (2005)
6. Lu, H., Chang, J., Yeh, M.: Design and analysis of direct-action CMAC PID controller. Neurocomputing 70, 2615–2625 (2007)
7. Sala, A., Cuenca, Á., Salt, J.: A retunable PID multi-rate controller for a networked control system. Information Sciences 179, 2390–2402 (2009)
8. Sumar, R.R., Coelho, A.A.R., Coelho, L.D.S.: Computational intelligence approach to PID controller design using the universal model. Information Sciences 180, 3980–3991 (2010)
9. Thangaraj, R., Chelliah, T.R., Pant, M., Abraham, A., Grosan, C.: Optimal gain tuning of PI speed controller in induction motor drives using particle swarm optimization. Logic Journal of IGPL 19, 343–356 (2010)
10. Juang, Y., Chang, Y., Huang, C.: Design of fuzzy PID controllers using modified triangular membership functions. Information Sciences 178, 1325–1333 (2008)
11. Ye, J.: Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot. Neurocomputing 71, 1561–1565 (2008)
12. Romero, J.A., Sanchis, R., Balaguer, P.: PI and PID auto-tuning procedure based on simplified single parameter optimization. Journal of Process Control 21, 840–851 (2011)
13. Sun, J., Zhang, D., Li, X., Zhang, J., Du, D.: Smith Prediction Monitor AGC System Based on Fuzzy Self-Tuning PID Control. International Journal of Iron and Steel Research 17, 22–26 (2010)
14. Astrom, K.J., Hagglund, T.: Advanced PID Control. Pearson Education, Madrid (2009)
15. Feng, Y.L., Tan, K.C.: Process identification and PID Control. John Wiley & Sons, USA (2009)
16. Visioli, A.: Practical PID Control. Springer, London (2010)
17. Johnson, M.A., Moradi, M.H.: PID Control: New identification and Design methods. Springer, London (2010)
18. Seki, H., Shigemasa, T.: Retuning oscillatory PID control loops based on plant operation data. J. Process Control 20, 217–227 (2010)
19. Tyreus, B.D., Luyben, W.L.: Tuning PI controllers for integrator/dead time processes. Industrial Engineering Chemistry Research 11, 2625–2628 (1992)

20. Astrom, K.J., Hagglund, T.: Benchmark Systems for PID Control. In: Preprints FAC Workshop on Digital Control. Past, Present and Future of PID Control, Tarrasa, pp. 181–182 (2000)
21. Vapnic, V.: The nature of statistical learning theory. Springer, New York (1995)
22. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
23. Parr, O.: Data Mining Cookbook. Modeling Data for Marketing, Risk, and Customer Relationship Management. John Wiley & Sons, Inc., New York (2001)
24. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Inc., Canada (2001)
25. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
26. Frank, E., Witten, I.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann (2005)
27. Rokach, L., Maimon, O.: Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing, USA (2008)
28. Alpaydin, E.: Introduction to Machine Learning. The MIT Press, Oxford (2009)

SpaGRID: A Spatial Grid Framework for High Dimensional Medical Databases

Harleen Kaur[1], Ritu Chauhan[2], Mohd. Afshar Alam[2], Syed Aljunid[1], and Mohd. Salleh[1]

[1] United Nations University - International Institute for Global Health
Kuala Lumpur, Malaysia
{harleen.k,syed.aljunid,mohamed.salleh}@unu.edu
[2] Dept. of Computer Science, Hamdard University
New Delhi, India
rituchauha@gmail.com,aalam@jamiahamdard.ac.in

Abstract. The outgrowth of technology in geographical databases has enhanced the growth of spatial databases, to deal with such enlarging databases scientists are laying down enormous efforts that can efficiently process these databases. Spatial data mining techniques has been collaboratively applied to extract implicit knowledge from spatial as well as non-spatial attributes. These techniques are efficiently applied in several fields such as healthcare, environmental, marketing and remote sensing databases to improve planning and decision making process. In this paper, we have designed and implemented SpaGRID framework for detection of spatial clusters. The framework has unprecedented efficiency to extract implicit knowledge of spatial data, due to its accessibility to handle and discover hidden patterns from spatial databases. We have also illustrated the usage of spatial variations among the United States men with prevalence of prostate cancer disease. The data of age group was taken from (15-65+) years in this group prostate cancers were examined and several stages of disease diagnosis was taken into account. The population of data was characterized by white, black and others were too small to be taken into account. Numerous challenges were encountered due to complexity of spatial datasets hence being resolved by certain statistical measures. The approach is to discover knowledge from spatial databases and design different aspects of knowledge discovery process from spatial databases.

Keywords: SpaGRID, Spatial Data, Spatial clustering, Octree data structure.

1 Introduction

Large number of techniques has been employed for mining of spatial databases. Traditional data mining techniques were not capable to deal with spatial data [10]. Hence, we require an appropriate mining technique to retrieve hidden relevant information from high dimensional spatial databases. Spatial data mining can be called as an extended version of data mining techniques which are capable to handle store and manipulate high dimensional spatial datasets.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 690–704, 2012.

A spatial database can be defined as platform to understand the relation between spatial as well as non-spatial attributes [14], [16], [17], [18]. Spatial attributes consists of data related to geographical values such as longitude, latitude zip code; whereas non-spatial attributes include temperature, height, weight, marital status, etc. The spatial databases can be collected from various sources such as geographic maps, satellite images, remote sensing, census database and climate or weather database.

Spatial data mining techniques were evolved to extract knowledge from high dimensional spatial data sets. The application of spatial data mining can be extensively studied in several research areas such as medical analysis, Geographic Information Systems (GIS), business analysis, artificial intelligence and several market survey related to geographical location [14], [35], [38]. Artificial intelligence and spatial data mining methods can be linked together to constitute for evaluating, monitoring and decision making process [36]. To detect patterns from high dimensional spatial datasets several spatial data mining techniques are categorized such as: spatial association, spatial clustering, spatial trend detection and spatial classification [1], [4]. The existing challenge in spatial data mining techniques is to discover mining algorithms to retrieve complex spatial datasets for accessing processes. The contribution of our work focuses in the field of spatial data mining, especially spatial clustering.

The spatial data mining tasks has being extensively studied in past decade for variety of medical domains to improve medical diagnosis [37]. This paper provides some facts on prostate cancer disease. The prostate cancer is among the leading cause of deaths among the United States men (Center for Disease Control, 2001). The ongoing study focuses on spatial variation of prostate cancer diagnosis for year (1992-1997). The data were collected and substantial distribution was found for individual case attributes, including age, race, year of diagnosis and geographical location of patients were examined. The proposed framework will help in strategic decision making for healthcare practioners.

The rest of the paper is organized as follows. Section 2 briefly discusses literature survey of several spatial data mining techniques for detecting patterns. In Section 3, we discuss the proposed SpaGRID framework for detecting clusters in three dimensional grid structures. Section 4 describes the overall relevance of spatial data structure for retrieval of information from SpaGrid framework with its implementation details. In Section 5 experimentation results were conducted on different spatial datasets for retrieval of clusters and conclusion is finally referred in last Section.

2 Related Works

Spatial clustering can be defined as process to group the data according to similarity between the spatial objects. It generally exists in two or more dimensional, as compared to traditional data clustering algorithms. The objective of spatial clustering algorithm is to maximize intra cluster similarity and minimize inter cluster similarity; it generally involves a criterion to measure similarity among data objects within cluster. Several statistical measures are applied such as Euclidian formula, mahalanobis distance to measure the similarity among objects inside the cluster. Application of spatial clustering techniques can be utilized in several domain areas such as medical

databases, marketing, education sector, galaxy clustering, remote sensing, micro array data and several other research areas. To discover clusters of different shapes several clustering algorithms exists such as: partitioning method, hierarchical method, density-based method and grid-based method [13], [19].

Partitioning based approach is the most popularized clustering technique developed to cluster data objects based on similarity. There are several preexisting partitioning based algorithms such as K-means, K-Medoids method, K-medoids also include Partitioning Around Medoids (PAM), Clustering Large Applications (CLARA) [12] and Clustering Large Applications based upon Randomized Search (CLARANS). The clustering technique which decomposes the data sets into the form of tree is known as hierarchical based clustering. They are generally classified into two approaches such as Agglomerative (AGNES) [12] and Divisive (DIANA) [12] approach. The algorithms existing under hierarchy are Clustering Using Representatives (CURE) [5], ROCK [7], Balanced iterative reducing and clustering using Hierarchies (BIRCH) [8] and CHAMELEON [11].

In density based algorithms data sets are merged according to density of object in space. The density based clustering algorithm discovers clusters of arbitrary shape. There are different types of density based clustering algorithms such as Density Based Spatial Clustering of Applications with Noise (DBSCAN) [3], [19], Ordering Points to Identify Clustering Structure (OPTICS) [2], Clustering Based on Density Distribution Function (DENCLUE) [9] and Distribution Based Clustering of Large Spatial Databases (DBCLASD) [4].

Grid based clustering method helps to determine the clusters of arbitrary shape using grid as a data structure. The existing grid based algorithms are Statistical based information grid (STING) [16], WaveCluster [15], CLIQUE [1], and Merging of Adaptive Intervals Approach to Spatial Data Mining (MAFIA) [6].

3 SpaGRID: A Spatial Grid Framework for High Dimensional Medical Databases

The outgrowth of technology in geographical databases has enhanced the growth of spatial medical databases, to deal with such enlarging databases the scientists are laying down enormous efforts that can efficiently process these databases. The optimal challenge is to create a unique environment with human interference and computational technology to discover effective and efficient spatial clusters for knowledge discovery process. To address issues related to discovery of integrated environment for retrieval of spatial clusters. A framework is designed to discover spatial clusters from high dimensional spatial medical databases. Figure.1, represents a SpaGRID framework for spatial abstraction of data, to retrieve clusters of variant shapes.

Given high dimensional spatial datasets as D, which is combination of spatial as well as non-spatial attributes $S <s_1,s_2,s_3, \ldots, s_k>$ collected over time interval $T < t_0,t_1,t_2,t_3, \ldots, t_n>$, where each data sets are collected and stored in D for certain specific time interval. The incremental D_i is subjected to the data mining algorithm resulting in discovery of spatial clusters. The proposed SpaGRID framework is a process to extract hidden interesting spatial clusters from high dimensional data using domain expert knowledge.

Usually high dimensional medical databases consists of multiple features which can be irrelevant for decision making; to handle such enlarged datasets feature selection technique is applied to find relevant subsets for spatial clustering approach. The first approach lay down focus on feature selection algorithms that falls into several categories such as (1) the filter approach, (2) the wrapper approach and (3) Embedded approach, [23], [22], [20]. The extracted features are now free from meaning less and redundant, [20], [23], [24] data and can be efficiently accessed by spatial clustering algorithms.

The second step focuses on determination of relevant spatial indices for storage and retrieval of high dimensionality of data. Spatial data structures are efficiently applied on to the grid to fetch the relevant data and detect several useful patterns with help of algorithms for decision making process. In past several spatial data structures have being studied for storage and retrieval of high dimensional data by, [21], [25], [26]. Our approach focuses on octree as a spatial hierarchy data structure for retrieval and indexing of high dimensional data.

The upper layer of framework is subjected to spatial clustering algorithm where combined approach of hierarchical and grid based clustering technique is applied to extract meaningful spatial patterns from the concerned knowledge of expert as earlier knowledge discovery process are relevant to user at specific time. The goal of current framework is to develop an integrated approach to detect spatial pattern from efficient and effective data mining techniques for exploration of complex large and high dimensional spatial datasets. The research focuses on spatial clustering technique to discover unknown hidden clusters from spatial medical databases.

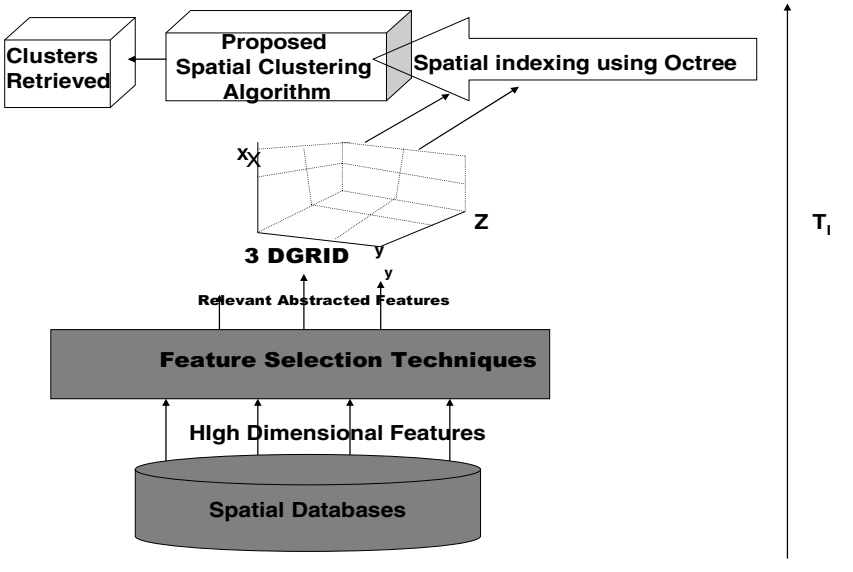


Fig. 1. SpaGRID: A Spatial Grid Framework

3.1 Feature Selection Techniques

The large amount of databases usually contain noise or irrelevant features such exceptionality of data can cause serious problems for analysis of data. Under such circumstances, we require an appropriate technique to select relevant features for analysis. The data mining technique such as feature selection is able to deal with discrimination between relevant and irrelevant features of datasets. There are several theoretical and empirical evidence which proves, that learner's performance starts to degrade in presence of irrelevant features, [24].

In past several feature selection algorithm have being proposed for supervised learning, rather than unsupervised. The fact behind less development in unsupervised feature selection is due to absence of class labels that can relatively search for efficient information. The relative criterion is to determine relevant features from large databases which finally depend upon evaluation of algorithm applied for retrieval of efficient subsets. The feature selection algorithms are generally divided into three categories such as filter approach, wrapper approach and embedded approach, [22], [27].

The filter approach is more efficient while handling high dimensional data. The evaluation begins while determine relevant features and then applying clustering algorithm. We can also say that filter approach first determines the relevant features and then clustering algorithm works with pre-selected features. These techniques are efficient applied for real world application domain for retrieval of efficient patterns. The Laplacian Score method is filtering process in which features are selected according to locality using weighted nearest neighbor graph, [29]. In wrapper approach the relevance of each feature is determined by specific clustering algorithm on the subset of datasets. These algorithms usually scale data in low dimensional space. Hence numerous clustering algorithms have being approached wrapper approach such as K-means, K-medoids where features are accessed by clustering algorithm for appropriate index, [28].

Definition 1.

The process of feature selection can be called as subset of datasets which satisfies the original data

$$H^{'} \subset H$$

where $H^{'}$ *represents subset features reduced from original data set H and output of features are determined from* $H^{'}$

Feature selection can be considered as preprocessing step to analyze and reduce the dimensionality of data, it proves to be essential step for retrieval of subsets from large databases. It can be defined as unsupervised learning technique but can be applied with supervised technique for retrieval of data. A key component to extract relevant patterns depends upon the dimensionality of data; it plays vital for determination of learning process. Learning process gets simplified and faster if extracted data has low dimension. Hence it increases the accuracy of algorithm.

Several feature selection approaches rely on the expert to determine the application problem domain with attributes; certainly the focus is to retrieved features or subset of data from original datasets to derive relevant attributes for data analysis, [30]. Several researchers in past has proved that usage of feature selection improves the performance of algorithms, prediction of data analysis, it reduces the computation cost and requirements, reduction in the storage space for high dimensional data, the performance of classifier improves as the error rate reduces down and easy understanding of model by expert has proven to certain benefits of feature selection technique, [20].

There are several other techniques applied to deal with high dimensionality of data reduction such as Principal Component Analysis (PCA) where the original data sets get reduced to low dimensional by converting itself into several other lower dimensions, these techniques can be useful for reducing the dimensionality but have certain constraints on clustering such as new dimensional can cause problems for domain expert to access the newly developed clusters with different dimensions and hence can cause expert to make improper decision making for the analysis of data.

3.1.2 Feature selection for Unsupervised Learning

As we know that data is progressively increasing from all sources, due to increase in computational world, constant high demand is to process the data for analysis and retrieve of knowledge. Feature selection techniques are primary applied to retrieve subset of data from raw data to find evitable patterns from it. Several feature selection techniques have being developed in past for supervised learning algorithm rather unsupervised algorithms, [31]. The supervised learning algorithm predefines the class before the retrieval of information, [4].

The unsupervised learning algorithms are difficult to implement as there are no predefined class in beginning. In literature we discussed several feature selection approaches such as filter and wrapper approach, to retrieve relevant subsets of features. Wrapper approach focuses on feature subsets, where the constant efforts are raised to investigate specific clustering algorithm on these subsets, subsets are retrieved by conducting several searches through feature space. These algorithms work efficiently for low dimensional data, whereas filter method works more efficiently for high dimensional data. Filter approach first implies certain properties then remove irrelevant features, before the clustering methods are finally applied. Recently many studies are conducted in field of unsupervised clustering to find the optimal subsets of features, [28]. The prime focus of these clustering algorithms is to retrieve information through specific clustering algorithm and finally find optimal feature subsets such as Expectation Maximization (EM) approach relied on initialization of objects and also retrieval of circular clusters, [28]. Similarly k-means algorithm was also implemented using certain cases for retrieval of effective patterns.

3.2 Spatial Clustering Methods

After feature selection technique is applied for retrieval of subset of data next step proceeds towards detection of clusters using proposed spatial clustering techniques.

The fetched data obtained from feature selection technique is now stored on SpaGRID. To determine patterns from SpaGRID has being found for spatial clustering. The spatial clustering approach measures the similarity of data by using statistical measuring techniques. Numerous applications of statistical techniques are applied in the past for discovery of spatial clustering algorithm [31], have utilized the approach of redefining spatial autocorrelation to derive the concept and algorithm for K-order neighbors based on Delaunay's triangulation,[33]. [34], discuss deterministic analysis of random walks on weighted graph generated from data to cluster spatial data, and even brute-force exhaustive search. The clustering approach can be further stated as:

Definition 2.

In given datasets D = {x_1, x_2, …, x_n), where each data point occurs in some dimensional space K (k<h), it partitions the data space into h number of objects depending upon the similarity hence clusters are formed with G= {g_1 ,g_2, …, g_m}

Spatial clustering algorithm proposed discover clusters in three dimensions space based on spatial hierarchical structure of data using known knowledge of spatial data structure. The construction of spatial hierarchy is evaluated using octree as spatial data structures. Octree is applied on SpaGRID for retrieval of relevant and efficient information from spatial data. It has being overlapped on grid to retrieve data which is spread on grid for storage and retrieval of clusters. The grid is formed in three dimensional planes as octree works on spatial hierarchical structure for three dimensional space, it recursively divides itself on multiples of eight. The grid is defined as finite structure with finite number of cells which helps in fast processing of data. Our spatial clustering algorithm efficiently scales the data in three dimensional grid spaces, hence octree has being identified as relevant spatial data structure for spatial clustering algorithm.

The proposed spatial clustering algorithm efficiently scales the data in three dimensional grid structure over the spaces. Several definitions are discussed below which are used as a part of spatial clustering techniques for retrieval of clusters:

- Let mean centre of each dimensions can be defined as $k(k_1,k_2,k_3,\ldots, k_n)/n$, $l(l_1,l_2,l_3,\ldots, l_n)/n$ and $m(m_1,m_2,m_3,\ldots,m_n)/n$ where n is defined as number of elements in list.
- The spatial correlation can be defined as measure of similarity or correlation among the variables, it can be clearly stated as below:

$$r_{kl} = \frac{n\sum kl - \sum k \times \sum l}{\sqrt{\left[n(\sum k^2) - (\sum k)^2\right]\sqrt{\left[n(\sum l^2) - (\sum l)^2\right]}}} \tag{1}$$

$$r_{km} = \frac{n\sum km - \sum k \times \sum m}{\sqrt{\left[n(\sum k^2) - (\sum k)^2\right]\sqrt{\left[n(\sum m^2) - (\sum m)^2\right]}}}. \tag{2}$$

$$r_{lm} = \frac{n\sum lm - \sum l \times \sum m}{\sqrt{\left[n(\sum l^2) - (\sum l)^2\right]\sqrt{\left[n(\sum m^2) - (\sum m)^2\right]}}}. \tag{3}$$

This type of standard spatial correlation exists between two variables which are represented as:

$r_{kl=}$ correlation coefficient between k and l
$r_{km=}$ correlation coefficient between k and m
$r_{lm=}$ correlation coefficient between l and m

Standard spatial correlation is computed by (1), (2) and (3) where n is number of observations in sample. Three dimensional correlations can be computed by multiple correlation coefficients. Multiple correlations involve more than two variables for computation. It can be evaluated by using the (4), where R signifies multiple spatial correlation as:

$$R = \sqrt{\frac{r_{km}^2 + r_{lm}^2 - 2(r_{km})(r_{lm})(r_{kl})}{1 - r_{kl}^2}}. \tag{4}$$

We know substitute the interpreted values to (4) which are derived from (1), (2) and (3). The larger the multiple correlations stronger are the relation between the variables and hence better prediction criterion.

The merging criterion is set as threshold inputted by expert using domain knowledge. Threshold value α_{value} is regarded as criterion for merging of similar spatial data in three dimension space:

$$\alpha_{value} = \begin{cases} 0, & \text{iff no similarity exists such that totally different} \\ 0.5, & \text{average correlation exists} \\ 0.8, & \text{strong correlation} \\ 1, & \text{highly similiar needs to be merged} \end{cases}$$

The α_{value} is always positive hence exists between 0.5 α_{value} +1 range.

4 Implementation

The proposed framework is implemented on high dimensional spatial datasets for determination of relevant spatial clusters. As discussed earlier octree has being used spatial data structure for storage and retrieval of spatial data. We have designed and implemented the structure of octree node for determination of clusters. The octree root node contains the information related to number of child nodes, *nodeid* of current root node, *mean value* of child nodes in three dimensional spaces and also the list of items. The list of items contains all statistical information as well as *nodeid*. Each node contains respective information of data available and hence clustering approach can be applied at different levels of hierarchy.

If the node is not a root node then structure for leaf or non-leaf node can be defined as below. The node values have been defined for each dimensional space such that k, l and m. The *parentnode* feature extract information related to node number of parent using information of immediate child nodes, whereas emerging node is leaf node or

non-leaf node has been determined by *Inode* value, node number of current node is also stored for further processing. The child node numbers of all non-leaf nodes are also determined by *parentnode* value. Statistical information of each node such as *spatial mean_value* has being derived from immediate child node values and spatial correlation of data among two nodes has been calculated using (4), in the end level of each node is also included to find the height of tree.

```
Struct   node {
 Float kcordinate, // value of k coordinate
 Float lcordinate, //value of l coordinate
 Float mcordinate, //value of m coordinate
 Int parentnode, //determine parentnode
 Bool lnode, // is leaf or non-leaf node
 Int nodenum, //node number of current node
 Float spatialmean _valuek, //mean value of k Coordinate
of child node
 Float spatialmean _valuel, //mean value of l Coordinate of
child node
 Float spatialmean _valuem, //mean value of m Coordinate
of child node
 Int childnode [8] [8], //array of child node
 Int childnode_number, // child node number of non-leaf
nodes
 Float spatial_correlation,//spatial   correlation
 Int listid[][]//list of elements
 Int level//level of node
 }
```

We have developed an algorithm to determine the node number of each node at different level of hierarchy.

Algorithm to Find Node Number:

Step 1: Store the parent node number to determine the child node numbers.

Step 2: If the existing node is leaf node than exit

Step 3: Else derive child nodes such as

(8k-6, 8k-5, 8k-4, 8k-3, 8k-2, 8k-1, 8k, 8k+1)

Where k represents the parent node number, store the node number of respective child nodes. The process iteratively continues until or unless each parent node determines its child nodes except leaf nodes.

Step 4: Process terminates after determining the child node number at different level of hierarchy except the leaf nodes

We have developed an algorithm to determine the parent node number of each node at different level of hierarchy.

Algorithm to determine the parent node number:

Step 1 Determines the child node number whose parent node value needs to be found.
Step 2 Store the child node number.
Step 3 If (nodenum%8 >1)
{ Parentnode =nodenum\8 +1; }
Else
{Parentnode = nodenum\8; }
Step 4 Return type of all parentnode is integer
Step 5 Determined nodes is root node then algorithm will exit

5 Experimentation

The proposed experiments are based on several public domain spatial medical data sets [39].The data contains substantial distribution for individual case attributes, including age, race, year of diagnosis and geographical location of patients were examined for United States. The population of data was characterized by white, black and others were too small to be taken into account. The data of age group was taken from (15-65+) years in this group prostate cancers were examined and several stages of disease diagnosis was taken into account. The multivariate characteristics of datasets exists which contain year of study from 1992-1997. The interesting results are formulated with the context of SpaGRID framework, whereas the feature selection techniques are approached and spatial clustering algorithms are implemented with MATLAB 7.1. Specific threshold values are input with domain expert knowledge.

Experiment

In this study we have selected the experiment on spatial health data sets where the approach was to reduce irrelevant and missing features and extract relevant reduced features using feature selection techniques. The dataset was examined using Naïve Bayes classifier and applying MIFS (Mutual Information Feature Selection) algorithm. A five fold cross validation is applied to retrieve relevant individual feature subset on the basis of target attribute. In current approach we have target spatial attribute state_name for retrieval of features related to spatial data, hence we have observed that attribute selected are relevant for further analysis of data. Table.1. observes the classifier performance Naïve Bayes on spatial datasets for retrieval of attributes.

Table 1. Classifier Performance

Value	Recall	1-Precision	Value	Recall	1-Precision
Alabama	0.8507	0.1739	Connecticut	1	0.7419
Alaska	0.9259	0.1379	Delaware	0.3333	0.6667
Arizona	0.7333	0.0833	Distt Co-lumbia	0	1
Arkansas	0.8933	0.2209	Florida	0.6567	0.2143
California	0.7414	0.1731	Georgia	0.8596	0.0926
Colorado	0.6094	0.1333	Tennessee	0.8462	0.1951
			Wyoming	1	0.4

The second experiment focuses on proposed spatial clustering technique and implemented using MATLAB 7.1. The reduced features abstracted from feature selections techniques are optimized for proposed spatial clustering to retrieve spatial clusters. The experiment calculates correlation between subsequent levels at each node; the correlation associates the similarity among different clusters when examined independently. The goal of our research is to retrieve spatial patterns and display output as clusters. It was approached that threshold value increased the compactness of the cluster. Fig. 2. depicts a cross-section view of spatial data obtained from a 3D pattern where the original data was visualized before clustering algorithm is applied.

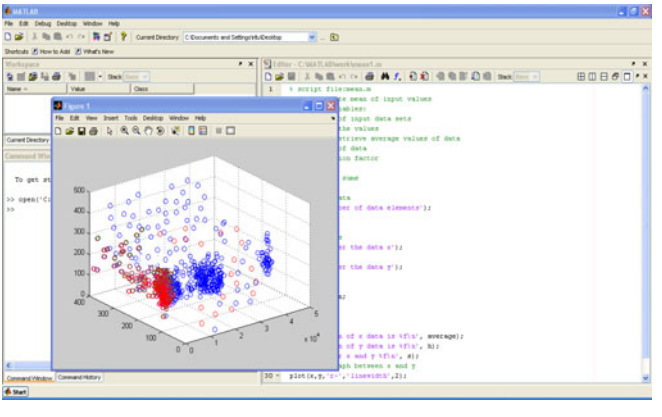


Fig. 2. Three Dimensional views of spatial data

Fig. 3 shows a 3D pattern formed by threshold value between (0 and 0.5) after clustering algorithm is applied and threshold was fixed between certain specific values.

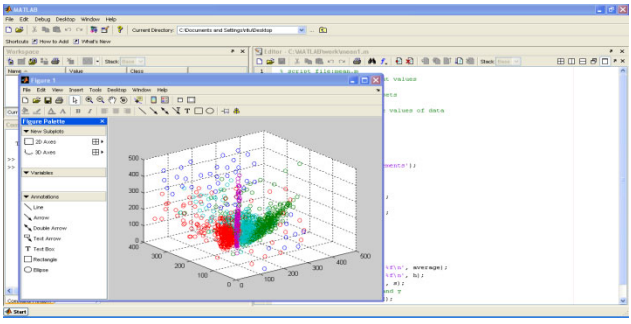


Fig. 3. 3D pattern formed by threshold value between (0 and 0.5)

Fig. 4 represents 3D pattern formed by threshold value between (0.5 and 0.1) after clustering algorithm are applied and threshold was fixed between certain specific values.

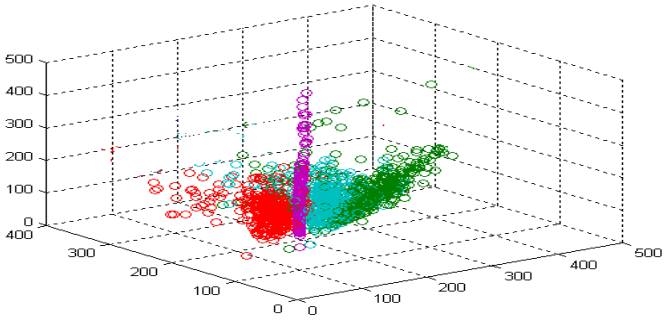


Fig. 4. 3D pattern formed by threshold value between (0.5 and 1)

In current approach our goal was to identify clusters having a range between certain specific threshold values. Fig.2, depicts the input data sets containing insignificant patterns. Whereas the output clusters are determined from Fig.3 and Fig.4, we analyzed the figure with different threshold values; we initially reduce the threshold value and started incrementing it. The four significant clusters indicate of high and low rates of each outcome for prostate cancer cases in United States during 1992-1997. The patterns of clusters individually change while we make changes in the independent attributes of data. We observed that increasing the threshold value till specific range clearly view the boundary of clusters. The output is generated using MATLAB 7.1. The purpose was to combine cluster detection analysis techniques with prostate cancer disease in order to enable a relationship between spatial and health care databases.

6 Conclusion and Future Works

The approach is to determine octree as an efficient and suitable index structure for three dimensional grid structures. We have demonstrated a spatial grid framework for three dimension grid with the help of octree as index data structure for spatial clustering of spatial attribute value. They are several algorithms implemented inside octree for independence of each data elements inside the tree and to discover hidden knowledge at different level of tree. The octree data structure is really efficient to deal with the spatial data and hence can prove to be backbone for future spatial data mining techniques.

The future work will be focused on other spatial index structure that can be helpful for determination of spatial data mining techniques for discovering clusters of variant shapes. To develop algorithm for spatial clustering with the help of octree and space filling curve for *n* dimensional data space. Also a knowledge decision making process to check the relevancy of each clusters determined.

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: Proceedings of the 1998 ACM-SIGMOD International Conference on Management of Data, Seattle, Washington, pp. 94–105 (June 1998)
2. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points To Identify the Clustering Structure. In: Proceedings of the ACM SIGMOD Conference, Philadelphia, PA, USA, pp. 49–60 (1999)
3. Borah, B., Bhattacharyya, D.K.: An Improved Sampling-based DBSCAN for Large Spatial Databases. In: Proceedings of the International Conference on Intelligent Sensing and Information, p. 92 (2004)
4. Xu, X., Ester, M., Kriegel, H.-P., Sander, J.: A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases. In: Proceedings of the International Conference on Data Engineering (ICDE 1998), Orlando, FL, pp. 324–331. AAAIPress (1998)
5. Giha, S., Rastogi, R., Shim, K.: CURE: an efficient clustering algorithm for large databases. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, pp.73–84 (June 1998)
6. Goil, S., Nagesh, H., Choudhary, A.: MAFIA: Efficient and Scalable Clustering for very large data sets. Technical Report No. CPDC – TR – 9906 – 010 ©1999 Center for Parallel and distributed Computing (June 1999)
7. Guha, S., Rastogi, R., Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. In: Proceedings of the 15th International Conference on Data Engineering, pp. 512–521 (March 1999)
8. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: Proceedings of SIGMOD International Conference, pp. 103–114 (1996)
9. Hinneburg, A., Keim, A.D.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD 1998), New York, NY, USA, pp. 58–65 (August 1998)

10. Kaur, H., Wasan, S.K.: An Integrated Approach in Medical Decision Making for Eliciting Knowledge, Web-based Applications in Health Care & Biomedicine. In: Lazakidou, A. (ed.) Annals of Information Systems (AoIS). Springer, Heidelberg (2009)
11. Karypis, G., Eui-Hong, H., Kumar, V.: CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer 32(8), 68–75 (1999)
12. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster in Analysis. John Wiley and Sons (1990)
13. Ng, R.T., Han, J.: Efficient and Effective clustering methods for spatial data mining. In: Bocca, J., Jarke, M., Zaniolo, C. (eds.) 20th International Conference on Very Large Data Bases, pp. 4–155. Morgan Kaufmann Publishers, USA (1994)
14. Miller, H.J., Han, J.: Geographic data mining and knowledge discovery. Taylor and Francis (2001)
15. Sheikholeslami, G., Chatterjee, S., Zhang, A.: WaveCluster: A Multi Resolution Clustering Approach for Very Large Spatial Databases. In: Proceedings of 24th Very Large Databases Conference (VLDB 1998), New York, NY, USA, pp. 428–439 (1998)
16. Wang, W., Yang, J., Muntz, R.: STING: A Statistical Information Grid Approach to Spatial Data Mining. In: Proceedings of the 23rd VLDB Conference, Athens, Greece, pp. 186–195 (1997)
17. Laurini, R., Thompson, D.: Fundamentals of spatial Information systems. Academic Press (1992)
18. Cressie, N.: Statistics for a Spatial Data, revised edn. Wiley, NY (1990)
19. Chauhan, R., Kaur, H., Alam, M.A.: Data Clustering Method for discovering clusters in Spatial Cancer Databases. International Journal of Computer Applications (2010)
20. Kaur, H., Chauhan, R., Alam, M.A.: An Optimal Categorization of Feature Selection Methods for Knowledge Discovery. In: Zhang, Segall, Cao (eds.) Visual Analytics and Interactive Technologies:Data, Text and Web Mining Applications. IGI Publishers Inc. (2010)
21. Seeger, B., Kriegel, H.P.: Techniques for design and implementation of spatial access methods. In: Proceedings of 14th International Conference on Very Large Databases, pp. 360–371 (1988)
22. Kohavi, R., John, G.: Wrappers for feature subset election. Artificial Intelligence 1-2, 273–324 (1997)
23. Dash, M., Liu, H.: Feature selection methods for classifications. Intelligent Data Analysis. An International Journal 1, 131–156 (1997)
24. Langley, P.: Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall Symposium on Relevance. AAAI Press, New Orleans (1994)
25. Peucker, T.K., Chrisman, N.: Cartographic data structures. American Cartographer 2, 55–69 (1975)
26. Overmars, M.H., Leeuwen, J.V.: Dynamic multi-dimensional data structures based on quad tree and k-d-trees. Acta Informatica 17(3), 267–285 (1982)
27. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97, 245–271 (1997)
28. Dy, J.G., Brodley, C.E.: Feature Subset Selection and Order Identification for Unsupervised Learning. In: Proceedings of the 17th International Conference on Machine Learning, pp. 247–254. Stanford University, CA (2000)
29. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: NIPS, pp. 507–514 (2005)
30. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)

31. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Norwell (1998)
32. Zhang, C., Murayama, Y.: Testing local spatial autocorrelation using k-order neighbors. International Journal of Geographical Information Science 14, 681–692 (2000)
33. Kang, I.S., Kim, T.W., Li, K.J.: A spatial data mining method by Delaunay triangulation. In: The 5th International Workshop on Advances in Geographic Information Systems, LasVegas, Nevada (1997)
34. Harel, D., Koren, Y.: Clustering spatial data using random walks. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California (2001)
35. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
36. Mann, S., Benwell, G.L.: The integration of ecological, neural and spatial modeling for monitoring and prediction for semi-arid landscapes. Computers and Geosciences 22(9), 1003–1012 (1996)
37. Jacquez, G.: Spatial analysis in epidemiology: Nascent science or a failure of GIS? Journal of Geographical Systems 2, 91–97 (2000)
38. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
39. `http://fha.maryland.gov`
40. Cohen, J., Cohen, P.: Applied multiple regression/correlation analysis for the behavioral sciences, 2nd edn. Erlbaum, Hillsdale (1983)

Author Index