Fusion of Visualization Induced SOM

Bruno Baruque and Emilio Corchado

Department of Civil Engineering, University of Burgos, Spain
{bbaruque, escorchado}@ubu.es

Abstract. In this study ensemble techniques have been applied in the frame of topology preserving mappings with visualization purposes. A novel extension of the ViSOM (Visualization Induced SOM) is obtained by the use of the ensemble meta-algorithm and a later fusion process. This main fusion algorithm has two different variants, considering two different criteria for the similarity of nodes. These criteria are Euclidean distance and similarity on Voronoi polygons. The goal of this upgrade is to improve the quality and robustness of the single model. Some experiments performed over different datasets applying the two variants of the fusion and other simpler models are included for comparison purposes.

1 Introduction

Nowadays, with the incredibly high amounts of data that industrial and business operations processes as well as research studies generate; the main problem is not the extraction of the data itself, but the extraction of useful information from within those huge databases. Among the great variety of analysis tools for multivariate data analysis and pattern recognition that are being developed, we can find the artificial neural networks.

Topology Preserving Maps [1], which include the Self-Organizing Maps (SOM) [2] and the Visualization Induced SOM [3], were originally created as a visualization tool; enabling the representation of high-dimensional datasets onto two-dimensional maps and facilitating the human expert the interpretation of data.

A general way of boosting the classification capabilities of classic classifiers (such as decision trees) is the construction of ensembles of classifiers [4], [5]. Following the idea of a 'committee of experts', the ensemble technique consists of training several identical classifiers on slightly different datasets in order to constitute a 'committee' to classify new instances of data. The topology preserving mapping algorithms can be considered in a certain way generic classifiers due to their inherent pattern storing and recognition capabilities [6]. Each of the units or neurons of these maps can be considered a classification unit specialized in the recognition of several similar patterns, as it reacts with higher values than the other to the presence of those patterns and not to the presence of others.

The guiding idea of this work is using the ensemble meta-algorithm approach on several topology preserving models to improve their stability and visualization performance.

E. Corchado et al. (Eds.): Innovations in Hybrid Intelligent Systems, ASC 44, pp. 151–158, 2007.
springerlink.com

2 The SOM and the ViSOM

Both models belong to a family of techniques with a common target: to produce a low dimensional representation of the training samples while preserving the topological properties of the input space. The best known technique is the Self-Organizing Map (SOM) algorithm [2]. It is based on a type of unsupervised learning called competitive learning; an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, sets of samples in a specific domain of the input space [1].

One interesting extension of this algorithm is the Visualization Induced SOM (ViSOM) [3], [7] proposed to directly preserve the local distance information on the map, along with the topology. The ViSOM constrains the lateral contraction forces between neurons and hence regularises the interneuron distances so that distances between neurons in the data space are in proportion to those in the input space.

The difference between the SOM and the ViSOM hence lies in the update of the weights of the neighbours of the winner neuron as can be seen from Eq. (1) and Eq. (2).

Update of neighbourhood neurons in SOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)\left(x(t) - w_v(t)\right) \tag{1}$$

Update of neighbourhood neurons in ViSOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)\left(\left[x(t) - w_v(t)\right] + \left[w_v(t) - w_k(t)\right]\left(\frac{d_{vk}}{\Delta_{vk}\lambda} - 1\right)\right) \tag{2}$$

where w_v is the winning neuron, α the learning rate of the algorithm, $\eta(v,k,t)$ is the neighbourhood function where v represents the position of the winning neuron in the lattice and k the positions of the neurons in the neighbourhood of this one, x is the input to the network and λ is a "resolution" parameter, d_{vk} and Δ_{vk} are the distances between the neurons in the data space and in the map space respectively.

To evaluate the quality of the adaptation of the map to the dataset represented a very widely known measure is the mean square quantization error. This can be represented as:

$$MSQE = \frac{1}{|D|}\sum_{x_i \in D}\left\|x_i - m_{c(x_i)}\right\|^2 \tag{3}$$

where $|D|$ is the number of data in the dataset D, and $m_{c(x_i)}$ is the best matching unit of the map to the data sample x_i of the dataset [8].

Another way of determining the quality of a map, when datasets include a class attribute for each of its entries, is the classification accuracy of the network. A high accuracy in the classification rate implies that the neurons of the network are reacting in a more consistent way to the classes of the samples that are presented, As a consequence, the map should represent the data distribution more accurately [9].

3 Mapping Fusion

3.1 The Ensemble Meta-algorithm

The ultimate goal for designing pattern recognition systems is to achieve the best classification performance possible for the task at hand. It has been observed in several studies that although one of the classifiers in an ensemble would yield the best performance, the sets of patterns misclassified by the different classifiers would not necessarily overlap. This suggests that different classifier designs potentially offer complementary information about the patterns to be classified and could be harnessed to improve the performance of the selected classifier [4]. As explained before, topological preserving maps can be considered as classifiers. Their learning algorithm specifies that their composing units (or neurons) specialize during the algorithm iterations in recognizing a certain type of patterns. This process is clearly a kind of classification, as that process can be used to group patterns by their similarity. The main problem of competitive learning based networks is that are inherently unstable due to the nature of their learning algorithm. Considering the topology preserving maps as a kind of classifiers; the main idea of this work is that the effect of this instability may, however, be minimized by the use of ensembles [10].

The algorithms to combine classifiers can be divided into two broad classes. The simpler variety of algorithms merely combines, by averaging in some way, the results of each of the composing classifiers of the ensemble yields into a final result. More complex types of algorithms try to combine not only the results but the whole set of classifiers in order to construct a single better one that should outperform its individual components. In the case of this paper this second perspective, the concept of a single "summary" or "synthesis" of the patterns stored within the whole ensemble, is the one followed to improve the model performance. This is because obtaining a unique map is the easiest way to obtain a clear visualization of data for human inspection, rather than representing several similar maps. Our main objective is to obtain a unique map that may be seen to represent all of the features of the maps in the ensemble.

3.2 Proposed Combinations

A number of ensemble techniques are applied to the SOM [11], ViSOM and other topological mapping networks, mainly for classification purposes.

In the context of the visualization, some adaptations are necessary to build a meaningful combination of the maps they represent. In this work a main algorithm for mapping fusion with two different variants is used for the first time in combination with the ViSOM. The objective is the comparison of the two in order to obtain conclusions that can be used in further studies to generate a more accurate model.

The procedure is the same for the training of the networks that compose the ensembles. All are trained using typical cross-validation, with the dataset divided into several folders, leaving one of them out to test the classification accuracy. To train the network ensemble the meta-algorithms called bagging [12] is used. It consists on obtaining n subsets of the training dataset through re-sampling with replacement and trains individual classifiers on such re-sampled subsets. This permits to generate *n* different trained networks which are combined into a final network that is expected to outperform each of them individually. The combination of maps is done once all the networks composing the ensemble have finished their training.

This combination is done in a neuron by neuron basis. That is, neurons that are considered 'near enough' one to the other are fused to obtain a neuron in the final fused network. This is done by calculating the centroid of the weights of the neurons to fuse:

$$w_{neuAvg} = 1/n \cdot \sum_{i=1}^{n} w(i) \tag{4}$$

That process is repeated until all neurons in all trained networks are fused into a unique final one. The criteria to determine which neurons are 'near enough' to be fused is what determines the two variants of the main algorithm.

Voronoi Polygons: Each neuron in a Self-Organizing Map can be associated with a portion of the input data space called the Voronoi polygon. That portion of the input multi-dimensional space is the portion that contains data for which that precise neuron is the Best Matching Unit (BMU) of the whole network [1]. It is therefore a logical conclusion to consider that neurons that are related to similar Voronoi polygons can be considered similar between them, as they should be situated relatively close in the input data space.

To calculate the dissimilarity between two neurons, a record of which data entries activated each neuron as the BMU can be kept. This can be done easily associating a binary vector to the neuron which length is the size of the dataset and contains ones in the positions where the neuron was the BMU for that sample and zeros in the rest of positions. The dissimilarity (i.e. the distance) between neurons can therefore be calculated as:

$$d(v_r, v_q) = \frac{\sum_{l=1}^{n} XOR(v_r(l), v_q(l))}{\sum_{j=1}^{n} OR(v_r(j), v_q(j))} \tag{5}$$

being *r* and *q* the neurons to determine their dissimilarity and v_r and v_q the binary vectors relating each of the neurons with the samples recognized by it. A much more detailed explanation can be found in [13].

The main problem with this proximity criterion is that it depends on the recognition of data by the network, rather than on the network definition itself. This means that a neuron that does not react as the BMU for any data could be considered similar to another neuron with the same characteristic, although they can be relatively far from each other in the input data space. To avoid this, all neurons with a reacting rate lower than a threshold are removed before calculating the similarities between them. This implies that the neighbouring properties of the network are no longer considered. To keep a notion of neighbouring between the neurons of the fused network the similarity criteria must be used again. Neurons with dissimilarity less than a threshold will be considered as neighbours in the fused network.

Algorithm 1. Fusion based on similarity in Voronoi Polygons

```
1: Being nNet the number of networks to be fused
fus the resultant fused network
$\theta_u$, $\theta_f$ and $\theta_c$ the usage, fusion and connection thresholds respectively
2: for i=1:nNet
3:   remove from the network the neurons that have a recognition rate lower than a usage
     threshold ( $\Sigma_i v_r(i) < \theta_u$ )
4:   add all the rest of the neurons of network(i) to the set of all nodes of ensemble
5: end
6: calculate the dissimilarity between all neurons contained in the set obtained in 3-6 using
     Eq. 5
7: group in different sub-sets the nodes that satisfy that the dissimilarity between all of them
     is lower than the dissimilarity threshold and the distance between each of them and the
     rest of nodes in other sub-set is higher than that threshold.
   $\begin{cases} ds(v(n_r),v(n_q)) < \theta_f & \text{for all } n_r,n_q \in S_k \\ ds(v(n_r),v(n_q)) \geq \theta_f & \text{for all } n_r \in S_k,\ n_q \in S_l,\ k \neq l \end{cases}$
   The result will be a set of sub-sets (S).
8: 'fuse' all the nodes in each sub-set to form a node of the final fused network by calculating
     the centroid of the nodes in each sub-set (see Eq. 4). The fused network will have as
     many nodes as sub-sets in are contained in S.
9: create the final network (fus) including in it all the fused nodes
10: create connections between fused nodes in the fused network (fus) to represent neuron
     neighbourhood. Connections will be established if the distance between fused nodes is
     lower than the connection threshold, considering this distance as:
                 $\min_{n_r \in N_r, n_q \in N_r} ds(v(n_r),v(n_q)) < \theta_c$
```

Euclidean Distance: This method involves comparing the networks neuron by neuron in the input space. This implies that all the networks in the ensemble must have the same size. First, it searches for the neurons that are closer in the input space

Algorithm 2. *Fusion based on Euclidean Distance*

```
0: Train several networks by using the bagging (re-sampling with replacement) meta-
algorithm
1: Being nNet the number of networks to be fused
   nNeur the number of neurons composing each network
   fus the resultant fused network
2: Initialize fus with the neuron weights of the first network
3: for i=2:nNet
4:   for j=1:nNeur
5:     neuFus : neuron (j) of the fus network
6:     calculate Euclidean Distance (ED) between neuFus and ALL neurons of network(i)
7:     neuNet: neuron with the minimum ED
8:     calculate neuAvg: neuron whose weights are the average of the weights of neuFus and
       neuNet i.e. the centroid of both neurons' weights (see Eq. 4).
9:     remove neuNet from the set of neurons of the network
10:    replace neuFus by neuAvg in the fus network (in position j of the network)
11:  end for
12: end for
```

(selecting only one neuron in each network of the ensemble) then it “fuses” them to obtain the final neuron in the “fused” map. This process is repeated until all the neurons have been fused. To deal with the high computational complexity of the algorithm, it can be implemented using dynamic programming. A more detailed description of this procedure can be found in [14].

The difference with the previous criteria is that, in this case, a pair wise match of the neurons of each network is always possible, so the final fused network has the same size as the single composing ones. This implies that a certain neighbouring structure can be kept and reconstructed in the fused network.

4 Experiments and Results

To test the characteristics and capabilities of the fusion of ViSOM and compare both of its variants several real datasets have been employed. Data were extracted form the UCI repository [15] and include the iris, the Wisconsin breast cancer and the wine datasets.

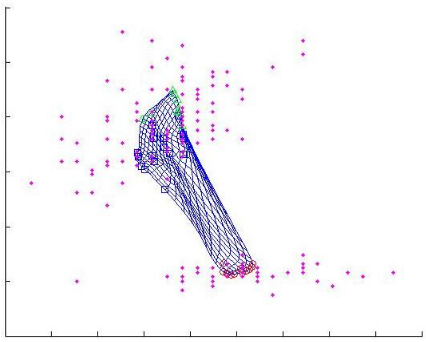

Fig. 1. A single ViSOM network represented over the iris dataset

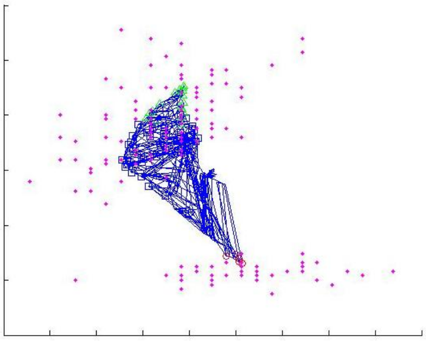

Fig. 2. The fusion of 5 ViSOM networks using the Euclidean Distance criterion

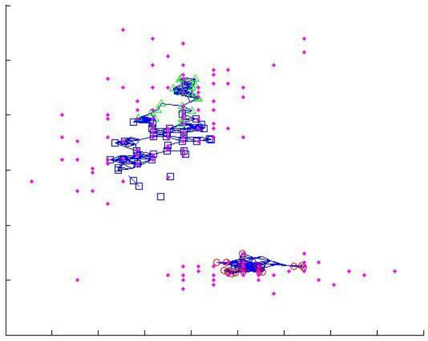

Fig. 3. The fusion of 5 ViSOM networks using the Voronoi polygon similarity criterion

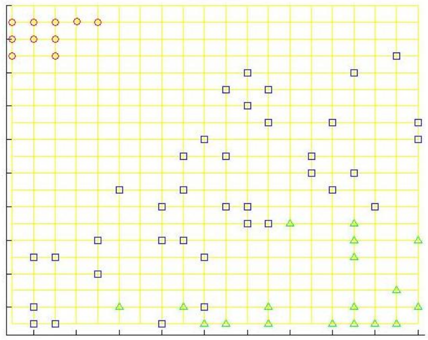

Fig. 4. The 2D map representation of the fused network represented on Fig. 2

The first thing that is important noting is a structural difference of the two variants of fusion. As explained before, the Euclidean distance variant enables the pair wise fusion of nodes of networks, so topology preservation is still valid in the fused network. This allows obtaining 2D maps such as the one showed in Fig. 4., which can be also easily obtained from single maps. This is impossible to do with the Voronoi similarity, as some neurons are not related to others and a position in the map in relation with the rest can not be determined.

Regarding the performance results, several fusions were calculated varying the amount of networks for the fusion from 1 to 20 (see Fig.5 to Fig. 8). Complementary results can be observed. For the Mean square error, this decreases when using more networks for the fusion if the Voronoi polygons similarity criterion is used, but will increases when using the Euclidean distance (Fig 5 and Fig 6). The same phenomenon can be appreciated when talking about the classification accuracy of the networks.

This suggests that while as the number of networks increases the Euclidean distance fusion variant tends to make network nodes to move away from the data in the input space. It also tends to avoid overfitting, while the Voronoi similarity variant has the opposite effect.

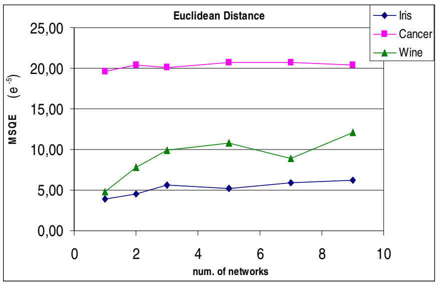

Fig. 5. Mean Square Quantization Error in the fusion using Euclidean distance for the three datasets when varying the number of networks to fuse

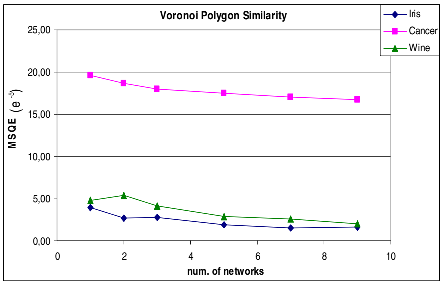

Fig. 6. Mean Square Quantization Error in the fusion using Voronoi polygons similarity for the three datasets when varying the number of networks to fuse

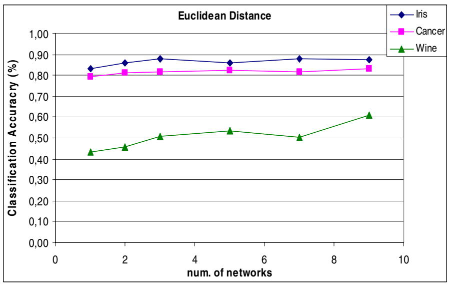

Fig. 7. Classification Accuracy (in %) in the fusion using Euclidean distance for the three datasets when varying the number of networks to fuse

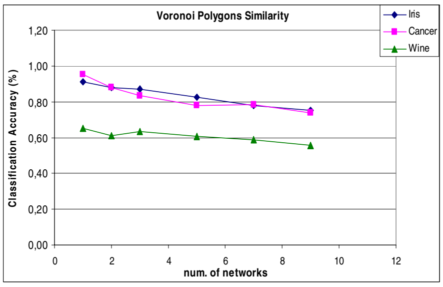

Fig. 8. Classification Accuracy (in %) in the fusion using Voronoi polygons similarity for the three datasets when varying the number of networks to fuse

5 Conclusions and Future Work

A study of a technique of improving the robustness and accuracy of a topology preserving model such as the ViSOM is presented in this work. This is achieved by the use of the ensemble theory and a fusion process. Two different variants of the algorithm are considered and studied, each one obtaining rather complementary results regarding their performance and the visual inspection suitability of their structure. Future work should include a further study of the capabilities of each one of the variants. This is considered as a previous step to the proposition of a fusion algorithm developed to bring together the best characteristics of both variants. Also application and comparison of this new fusion algorithm with some other topology preserving ensemble models will be performed.

Acknowledgments. This research has been supported by the MCyT project TIN2004-07033 and the project BU008B05 of the JCyL.

References

1. Kohonen, T., et al., *A principle of neural associative memory.* Neuroscience, 1977. 2(6): p. 1065-1076.
2. Kohonen, T., *The self-organizing map.* Neurocomputing, 1998. 21(1-3): p. 1-6.
3. Yin, H., *ViSOM - a novel method for multivariate data projection and structure visualization.* Neural Networks, IEEE Transactions on, 2002. 13(1): p. 237-243.
4. Kuncheva, L.I., *Combining Pattern Classifiers: Methods and Algorithms.* 2004: Wiley-Interscience.
5. Ron, M. and R. Gunnar, *An Introduction to Boosting and Leveraging.* Advanced Lectures on Machine Learning: Machine Learning Summer School 2002, Canberra, Australia, February 11-22, 2002. Revised Lectures. 2003. 118-183.
6. Kaski, S., *Data exploration using self-organizing maps*, in *Department of Computer Science and Engineering*. 1997, Helsinki University of Technology: Espoo, Finland.
7. Yin, H., *Data visualisation and manifold mapping using the ViSOM.* Neural Networks, 2002. 15(8-9): p. 1005-1016.
8. Kaski, S. and K. Lagus, *Comparing Self-Organizing Maps*, in *International Conference on Artificial Neural Networks (ICANN'96).* 1996 Springer-Verlag.
9. Kraaijveld, M.A., J. Mao, and A.K. Jain, *A nonlinear projection method based on Kohonen's topology preserving maps.* Neural Networks, IEEE Transactions on, 1995. 6(3): p. 548-559.
10. Ruta, D. and B. Gabrys, *A theoretical analysis of the limits of Majority Voting errors for Multiple Classifier Systems.* Pattern Analysis and Applications, 2002. 5(4): p. 333-350.
11. Petrakieva, L. and C. Fyfe, *Bagging and Bumping Self Organising Maps.* Computing and Information Systems, 2003.
12. Breiman, L., *Bagging Predictors.* Machine Learning, 1996. 24(2): p. 123-140.
13. Saavedra, C., et al. *Fusion of Self Organizing Maps.* in *9th International Work-Conference on Artificial Neural Networks, IWANN 2007.* 2007. San Sebastián, Spain: Springer.
14. Georgakis, A., H. Li, and M. Gordan, *An ensemble of SOM networks for document organization and retrieval.* Int. Conf. on Adaptive Knowledge Representation and Reasoning (AKRR'05), 2005: p. 6.
15. Newman, D.J., et al. *UCI Repository of machine learning databases.* 1998; Available from: http://www.ics.uci.edu/~mlearn/MLRepository.html.