

Avances en Informática y Automática

Duodécimo Workshop



VNIVERSIDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL



Avances en Informática y Automática

Duodécimo Workshop

Avances en Informática y Automática

Duodécimo Workshop

Editores

André Sales Mendes

Publicado en España por:

Departamento de Informática y Automática Facultad de Ciencias
Universidad de Salamanca
Plaza de los Caídos s/n
37008, Salamanca, España
Tel.: + 34 923294653
Fax: + 34 923294514
Web: <http://mastersi.usal.es>
Email: mastersi@usal.es

ISBN 978-84-09-02436-0

Editores:

André Sales Mendes

Prólogo

El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. El congreso organizado por el Departamento de Informática y Automática que se celebra dentro del Máster en Sistemas Inteligentes de la Universidad de Salamanca proporciona la oportunidad ideal para que sus estudiantes presenten los principales resultados de sus Trabajos de Fin de Máster y obtengan una realimentación del interés de los mismos.

La duodécima edición del *workshop* “Avances en Informática y Automática”, correspondiente al curso 2017 - 2018, ha sido un encuentro interdisciplinar donde se han presentado trabajos pertenecientes a un amplio abanico de líneas de investigación. Todos los trabajos han sido supervisados por investigadores de reconocido prestigio pertenecientes a la Universidad de Salamanca, proporcionando el marco idóneo para sentar las bases de una futura tesis doctoral. Entre los principales objetivos del congreso se encuentran:

- Ofrecer a los estudiantes un marco donde exponer sus primeros trabajos de investigación.
- Proporcionar a los participantes un foro donde discutir ideas y encontrar nuevas sugerencias de compañeros, investigadores y otros asistentes a la reunión.
- Permitir a cada estudiante una realimentación de los participantes sobre su trabajo y una orientación sobre las futuras direcciones de investigación.
- Contribuir al desarrollo del espíritu de colaboración en la investigación.

Organización

El *workshop* “Avances en Informática y Automática” está organizado por el Departamento de Informática y Automática de la Universidad de Salamanca.

Comité Organizador

Ángel Luis Sánchez Lázaro

María Moreno García

Belén Curto Diego

Francisco Javier Blanco Rodríguez

Roberto Therón Sánchez

Ángel Francisco Zazo Rodríguez

Índice general

Extracción de parámetros rítmicos para clasificación y/o adaptación musical	1
<i>Alejandro Acosta Vellón, Belén Pérez Lancho, Ana Gil, María Navarro Cáceres</i>	
Sistema de Seguimiento, Detección y Localización para Vehículos Autónomos	18
<i>André Sales Mendes, Juan Francisco De Paz, Gabriel Villarrubia González, Juan Manuel Corchado Rodríguez</i>	
Sistema de Recomendación de Música Sensible al Contexto	40
<i>F. Andrés Yepes Vélez, Vivian F. López Batista, María N. Moreno</i>	
Sistema de Recomendación de Canciones Basado en Aspectos Emocionales	65
<i>Carlos J. Gomes González, María N. Moreno García y Ana B. Gil</i>	
Sistema de Monitorización y Detección de Patrones Anómalos Mediante el Uso de Sensores Médicos	81
<i>Christian Bonal Martín, Juan Manuel Corchado Rodríguez, Gabriel Villarrubia González, Juan Francisco de Paz Santana</i>	
Sistema para la Minería de Opiniones	97
<i>Daniela Elizabeth Sanjinés Flores, Vivian F. López Batista, Ana B. Gil-González</i>	
Sistemas Conexionistas y de Visión Artificial para Detección de Iconografía Radical	110
<i>Henri Estuardo Castro León, Angélica González Arrieta</i>	
Prediction of Future Immediate Actions	125
<i>Mehmet Ozturk</i>	
Análisis de Datos para el Mantenimiento Predictivo	138
<i>Ricardo Martínez, Belén Pérez, Sara Rodríguez</i>	
Autores	165

Extracción de parámetros rítmicos para clasificación y/o adaptación musical

Alejandro Acosta Vellón, Belén Pérez Lancho, Ana Gil, María Navarro Cáceres

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{ale.av, lanch, abg, maria90}@usal.es
<http://diaweb.usal.es/>

Resumen El presente estudio analiza la utilización de los sistemas inteligentes para la creación de aplicaciones que se adapten a las necesidades habituales de las personas. En particular, se ha abordado la interacción de los usuarios con la música durante una actividad física: correr. A lo largo de este proyecto se desarrolló una aplicación *fitness*¹ para móvil que reproduce y adapta canciones mientras el usuario está en movimiento. Mediante el uso de los sensores del dispositivo móvil, la aplicación es capaz de ajustar el ritmo o tempo² de una canción a la velocidad de carrera del individuo. Previo al desarrollo ha sido necesario realizar una revisión del estado actual del mercado de las aplicaciones *fitness*.

Un grupo de voluntarios (n=12) con experiencia previa en el uso de aplicaciones *fitness* fue seleccionado para probar la aplicación. Los participantes fueron encuestados sobre su impresión al usar esta aplicación en comparación con las aplicaciones *fitness* clásicas y se les pidió estimar los límites aceptables de distorsión del tempo de las canciones.

Los resultados revelan que a más de un 50% de los participantes le resultaría incómodo correr sin música. Los límites aceptables de alteración del tempo en las canciones están entre el 26% de aumento y 15% de disminución respectivamente. Un 83% indicó que le gustaría que su aplicación *fitness* de uso cotidiano incluyera un ajuste manual o adaptativo del tempo de las canciones. Se comprueba que existe un margen desaprovechado que puede permitir a los usuarios beneficiarse en el uso de sus canciones favoritas ajustadas a su medida. Beneficios más allá de psicológicos que aumentarían nuestro rendimiento en las actividades aeróbicas.

Keywords: Sistemas inteligentes, Correr, Tempo, *Fitness*, Adaptable

¹ Aplicación para el entrenamiento físico y cuidado de salud

² Ritmo, o grado de celeridad de una composición musical

1. Introducción

En un mundo donde la generación de contenidos multimedia crece a un ritmo exponencial, la música se ha convertido en parte inseparable de nuestra vida cotidiana. Ya sea en un mercado, una tienda de ropa, una fiesta en la plaza: la música nos acompaña muchas veces sin darnos cuenta.

Resulta que, con el continuo avance de las tecnologías, la música se ha convertido en un espectro gigantesco de géneros y formas que alcanzan a cubrir la gran mayoría de los gustos de los consumidores. Pareciera como si esta forma de arte intentara cubrir cada posible nicho cultural y social donde exista un público que se pueda identificar con una nueva forma de hacer música. Siguiendo, a todas luces, los patrones de la naturaleza dinámica de los fenómenos de demanda y oferta dentro de la sociedad. (Jacobson et al. 2016)[9] destaca que el descubrimiento y la personalización es parte clave de la experiencia y fundamental para el éxito del creador y el ecosistema de consumo.

El impacto de la moda es especialmente obvio en la música popular. En las sociedades modernas, el cambio cultural parece incesante. (Mauch et al. 2015)[17] resaltan en su trabajo que aunque mucho se ha escrito sobre el origen y evolución de los géneros musicales la mayoría de las afirmaciones sobre la evolución de la música popular suelen ser de naturaleza anecdótica en lugar de científica. Esto se debe, en gran parte, a que la música no suele cambiar de forma brusca de la noche a la mañana. Los géneros musicales van evolucionando y engendrando otros nuevos. En cualquier caso, algo queda claro, existe una tendencia natural a que la música se acerque cada vez más al público y se adapte a los más disímiles oídos y gustos. Esto es especialmente cierto en el mundo de la música orientada a acompañar rutinas de ejercicios o para hacer running donde tanto el ritmo, intensidad y carga emocional de las canciones están destinadas a motivar a los corredores y proveer un paso consistente. La cadencia de la música prueba ser importante en cuanto puede impactar espontáneamente en el ritmo de carrera, como observaron (Van Dyck et al. 2015)[25].

1.1. Objetivos

1. Hacer un estudio del estado del arte de las aplicaciones *fitness* para correr.
2. Desarrollar una aplicación móvil *fitness* que, con el uso de sistemas inteligentes, permita la adaptación del tempo de las canciones a las necesidades del usuario.
3. La aplicación incluirá las funcionalidades básicas de las aplicaciones *fitness* para correr: contador de pasos, cálculo de distancia, estimación de pasos por minuto y reproductor de audio.
4. La aplicación será capaz del calcular, en tiempo real, el tempo del paso del usuario y ajustar la canción en curso para que su BPM¹ coincida.
5. Realizar una encuesta para comprobar cómo los usuarios usan la música para correr y comprobar la aceptación de las nuevas funcionalidades ofrecidas por la aplicación en comparación a las aplicaciones *fitness* clásicas.

¹ Unidad de medida del tempo. Número de pulsaciones por minuto

6. Determinar los límites tolerables de distorsión del tempo en las canciones a través de la información proporcionada por los participantes.

2. Estado del Arte

La música representa para la mayoría de los corredores un empuje motivacional que se asocia con un mejor rendimiento, especialmente en carreras y otros ejercicios aeróbicos (Lane et al. 2011)[13]. La predisposición natural de los humanos para responder a las características rítmicas de la música ha sido reconocida y documentada desde principios del siglo XX (Macdougall 1903)[16]. En sus estudios entre los años 1999 y 2006, Karageorghis et al. se refirieron a este fenómeno como: *rhythm response* (respuesta rítmica) mientras desarrollaban de su Inventario de Calificación Musical de Brunel (BMRI), una medida utilizada para calificar las cualidades motivacionales de la música (Karageorghis et al. 1999)[11], (Karageorghis et al. 2006)[10].

En las personas sedentarias o sin preparación física podría parecer obvio el efecto favorable del uso de la música, especialmente para impulsarles a iniciar a correr y empezar de forma más motivada la actividad física (Tenenbaum et al. 2004)[22]. Investigaciones más recientes muestran que lo anterior es también cierto para deportistas de alto nivel.

En un estudio desarrollado por la Universidad de Southern Queensland en Australia sobre los efectos del uso de música síncrona en 11 triatletas de élite mientras corren sobre cintas de correr o *treadmills* (Terry et al. 2012)[23], se encontró que la música proporciona beneficios psicológicos y fisiológicos durante el entrenamiento aeróbico intenso, beneficios que están probablemente interconectados.

Los autores de dicho trabajo condujeron una serie de pruebas para medir el rendimiento de los triatletas (n=11) durante el ejercicio. Los participantes fueron sometidos a tres tareas en donde se les pidió previamente elegir música motivacional de gusto personal y música que consideraban neutral con el mismo tempo.

El estudio intentó demostrar la correlación entre el uso de la música síncrona¹ y un mayor rendimiento de los atletas durante el entrenamiento. Las tres pruebas del experimento desarrollado por (Terry et al. 2012)[23] fueron:

1. Correr al ritmo de música motivacional de elección personal
2. Correr al ritmo música neutral al mismo tempo que la motivacional
3. Correr sin música

El estudio obtiene, entre otros resultados, medidas del rendimiento de los atletas considerando tres medidas de rendimiento bajo las tres condiciones mencionadas anteriormente:

- Tiempo de actividad física hasta la fatiga

¹ Se entiende por música síncrona aquella con un tempo bien marcado y estable

- Intensidad del ejercicio: RPE¹ (Rated Perceived Exertion) en el momento de alcanzar la fatiga
- Concentración de lactato² en el momento de fatiga

Los resultados se muestran de forma resumida en la Tabla 1. Desde donde el estudio confirma que la música síncrona puede potencialmente proporcionar beneficios a todo tipo de actividades aeróbicas y actividades de ejercicios hasta la fatiga. Aunque la música neutral no produjo, en general, el mismo nivel de beneficios psicológicos que la música motivacional, resultó igualmente beneficiosa en términos de rendimiento hasta la fatiga y en el consumo de oxígeno. En términos funcionales, por lo tanto, las cualidades motivacionales percibidas de la música pueden ser menos importantes que, por ejemplo, la prominencia de su ritmo y el grado en que los participantes son capaces de sincronizar sus movimientos con su tempo (Terry et al. 2012)[23].

Prueba	Motivacional	Neutral	Sin Música
Tiempo hasta fatiga (s)	509	516	431
Intensidad (fátiga) (RPE)	17.91	17.82	17.73
Lactato ($mmol^{-1}$)	6.47	6.16	5.94

Tabla 1: Rendimiento hasta fatiga, RPE, y concentración de lactato para 11 triatletas. Media estándar.

El estudio concluye determinando que el uso de música síncrona durante el entrenamiento debería ser considerado debido a que muestra claras ventajas.

Hallazgos clave del estudio

- Se debe asegurar que el tempo de la música corresponde con el tempo del movimiento deseado.
- Realizar un análisis de la marcha puede ayudar a la elección de la música con tempo óptimo para diferentes cadencias de ejecución.
- Se debe alentar a los atletas a seleccionar su propia música motivacional, ya que prueba ser más efectiva.

2.1. Las Aplicaciones *fitness*

Cada vez con más frecuencia, el consumo de música durante la realización de actividades físicas se realiza a través de alguna aplicación destinada al entrenamiento o a cumplir unas metas de salud. Según un estudio de (Krebs et al. 2015)[12] dentro de USA un 58.23 % de los usuarios de móviles habían usado alguna vez alguna aplicación *fitness* o de nutrición.

¹ Rated Perceived Exertion, una escala para medir la intensidad de una actividad física

² Forma ionizada del Ácido Láctico. Producido en los músculos durante la actividad física

Si bien el mercado online está saturado de aplicaciones *fitness*, la gran mayoría ofrecen poca personalización o bien ofrecen listas de reproducción predefinidas con ritmos estrictos. En la Tabla 2 se exponen algunas de las aplicaciones *fitness* para correr más populares en Google Play que de alguna forma permitan interactuar con la música.

Aplicación	Características
Spotify Running	Busca canciones con Tempo similar, selección libre. Carece de ajuste personalizado del tempo para cada canción. La aplicación fue cancelada en febrero de 2018.
RockMyRun	Selección de canciones para correr, diferentes ritmos. Selección fija, tempo de canciones no puede ser ajustado.
Runkeeper	Registro de actividad física, carece de información de tempo. Sin música.
Nike+ Run Club	Entrenador en audio personalizado, no usa indicadores de tempo.
TempoRun	Sin tempo objetivo, solo registra la actividad física

Tabla 2: Características de las aplicaciones más populares para correr

Aunque existe una tendencia creciente al uso de aplicaciones para acompañar las actividades físicas, el abandono de estas es también bastante común. En USA cerca de la mitad de los usuarios reportó haber abandonado una aplicación *fitness* a causa de una exigencia alta de entrada de datos personales para el registro, falta de interés o costes ocultos (Krebs et al. 2015)[12]. Los hallazgos de (Krebs et al. 2015)[12] sugieren que si bien muchas personas usan aplicaciones de salud, una proporción sustancial de la población no lo hace, y que incluso entre quienes usan aplicaciones de salud, muchos dejan de usarlas. Estos datos sugieren que los desarrolladores de aplicaciones deben abordar mejor las preocupaciones del consumidor, como el costo y la alta carga de entrada de datos, y que los ensayos clínicos son necesarios para probar la eficacia de las aplicaciones de salud para ampliar su atractivo y adopción.

Hemos comprobado que efectivamente, varios estudios demuestran que el uso de la música, especialmente música sincrona puede resultar muy efectiva para aumentar tanto la motivación como el rendimiento de los corredores (Lane et al. 2011)[13], (Van Dyck et al. 2015)[25], (Terry et al. 2012)[23]. Esto claro está, siempre y cuando el ritmo de la música sea bien marcado, estable y el tempo de la música se ajuste a los requisitos de la actividad física. La solución a estos requisitos sería trivial si todas las personas corriéramos a un ritmo similar, pero la realidad es que la cadencia natural de los corredores recreacionales se encuentra entre los 130 y 200 pasos por minuto, según confirma (Van Dyck et al. 2015)[25]. Este es un intervalo considerablemente amplio. Pasa que, por ejemplo, canciones de 180 pulsaciones por minuto (BPM) comúnmente utilizadas en muchas aplicaciones funcionarían bien para algunas personas, pero definitivamente no

para todas, por eso muchas personas terminan seleccionando manualmente sus propias listas de reproducción desechando muchas veces sus canciones favoritas para quedarse con aquellas que se acercan más a su propio paso.

Aún con el gran número de aplicaciones que brindan la reproducción de música a modo de guía o efecto motivador, son pocas las que ofrecen la funcionalidad de ajustar el tempo a medida y todas carecen de la posibilidad de modificar el tempo de una canción.

Es por eso que en este estudio se intentan cubrir las posibles ventajas y límites que tiene el uso del control procedural del tempo de las canciones de forma personalizada a través de los requisitos del usuario. Con este fin hemos desarrollado este prototipo de aplicación adaptable, a la que se ha llamado *Tempo Running*, con la que intentaremos comprobar si este enfoque es un paso en la dirección correcta.

3. Métodos y Herramientas

3.1. La Aplicación: Tempo Running

Como prueba de concepto se ha desarrollado un prototipo de aplicación Android que cubre las funcionalidades básicas de las aplicaciones *fitness* para carrera. Se ha evitado el uso de cualquier tipo de registro o forma de enlace a redes sociales. En su estudio (Gowin et al. 2015)[7] comprobaron que la mayoría de los usuarios (específicamente jóvenes estudiantes) se oponen fuertemente a enlazar sus cuentas sociales con aplicaciones de *fitness* o de alguna forma proporcionar datos personales que puedan ser usados por el desarrollador del software.

Una de las formas de incentivar al usuario al uso de una app *fitness* es hacer que esta contenga mini retos o sistemas dinámicos que hagan que la interacción entre el usuario y la aplicación sea más divertida y gratificante. Elementos y técnicas que son conocidos popularmente con el anglicismo **gamificación** que intentan potenciar la motivación y el refuerzo de conductas (Sandí et al. 2013)[19]. La gamificación en si no es un juego; la gamificación es un concepto distinto y que tiene menos que ver con los juegos que con el marketing, la motivación o la fidelización de los usuarios (Cortizo et al. 2011)[4].

Esta tendencia a insertar elementos de juego en las aplicaciones de salud y fitness se ha vuelto inmensamente popular, como lo demuestra la cantidad de aplicaciones que se encuentran en la App Store de Apple que contienen minijuegos (Lister et al. 2014)[15]. Sin embargo, existe una falta de consenso sobre qué elementos son importantes en la teoría del comportamiento en la industria de las aplicaciones, lo que puede afectar a la eficacia de estas aplicaciones para inducir y conservar comportamientos más activos con hábitos saludables en los usuarios. Las aplicaciones fitness representan un mercado creciente y muy prometedor en el que difundir intervenciones de cambio de comportamiento de salud (Lister et al. 2014)[15]. Por esta razón resulta importante tener en cuenta el carácter interactivo y dinámico de la aplicación que vamos a desarrollar utilizando pinceladas de técnicas de gamificación.

Inclusión de Funcionalidades Dinámicas, Retroalimentación:

1. Detección de cantidad de pasos y distancia recorrida en tiempo real.
2. Detección del ritmo de carrera en pulsaciones por minuto (BPM).
3. Reproducción de música con indicación explícita y gráfica de su BPM base.

La clave del triunfo de una app fitness es lograr incentivar la actividad física. A pesar de la proliferación de aplicaciones de *smartphones* comercialmente disponibles orientadas a fitness, hay una escasez de evidencia empírica para apoyar su efectividad en crear hábitos más saludables (Direito et al. 2015)[5].

Entorno de Desarrollo La aplicación Android ha sido desarrollada desde el sistema operativo Windows mediante el uso del Entorno Integrado de Desarrollo (IDE): Android Studio en su versión 3.1.3 (Smyth 2017)[21]. El lenguaje de programación principal utilizado para los procedimientos y programación de algoritmos es Java, un lenguaje de alto nivel muy avanzado que permite desarrollar de forma rápida y eficiente con abundante documentación disponible (Schildt 2014)[20]. La interfaz gráfica ha sido programada usando XML para la construcción de elementos de visualización de forma modular y sencilla (Benz et al. 2004)[3].

La Interfaz Gráfica La aplicación ofrece una vista única donde se disponen todos sus elementos:

Elementos de Interfaz

1. Controles de ajuste de tempo
2. Lista de Reproducción
3. Elementos de visualización

En la Figura 1 se pueden apreciar los tres componentes principales mencionados con anterioridad que intentan responder a los requisitos de funcionalidad básicos para una aplicación fitness de este estilo. Se ha utilizado como referencia de diseño la aplicación móvil Spotify Running (actualmente fuera de funcionamiento), la cual gozó de una gran una gran aceptación, pero a diferencia de la aplicación desarrollada en este proyecto, Spotify Running era incapaz de modificar el tempo de las canciones. Su función principal era la de recomendar canciones dado un tempo prefijado por el usuario usando botones para incrementar o disminuir el valor en 10 BPM (Greenhalgh et al. 2016)[8]. Esta funcionalidad fue contemplada en el desarrollo de la aplicación Tempo Running.

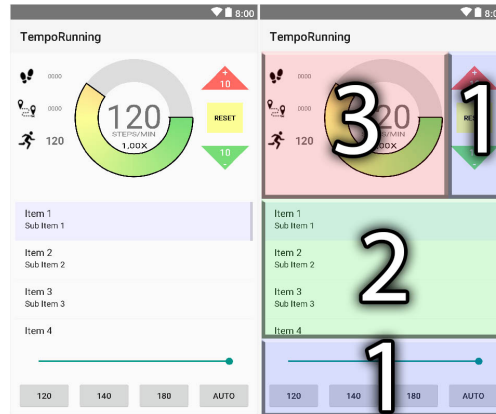


Figura 1: Interfaz de Tempo Running

Controles Los elementos de control marcados en color azul en la Figura 1 se encuentran separados en dos grupos. **Elementos de Control**

1. Controles relativos en la parte superior (en colores rojo, amarillo y verde)
2. Controles absolutos en la parte inferior

Los **Controles Relativos** están compuestos por tres botones:

1. Aumentar el tempo en 10 BPM (Triángulo rojo)
2. Restaurar el tempo original de la canción (Rectángulo amarillo)
3. Disminuir el tempo en 10 BPM (Triángulo verde)

Los controles absolutos (Figura 1) están compuestos por una barra de búsqueda y cuatro botones. Se han utilizado los valores prefijados de 120, 140 y 180 BPM para tener un paso inicial ligero hasta cubrir el ritmo de 180 BPM el más alto comúnmente utilizado dentro del intervalo de 130 y 200 BPM predominante en los corredores recreacionales según comprobaron (Van Dyck et al. 2015)[25].

Controles Absolutos

1. Barra de búsqueda para selección manual de tempo (60 - 200 BPM)
2. Botón de 120 BPM para poner el ritmo de reproducción actual a ese valor
3. Botón de 140 BPM
4. Botón de 180 BPM
5. Botón AUTO: Activa el modo de detección del tempo del usuario y lo aplica a la canción actual

El límite superior de tempo de 200 BMP para nuestra aplicación quedó confirmado en consonancia con los hallazgos de (Van Dyck et al. 2015). Por otro lado según varias bases de datos de tempo (SongBpm, bpmdatabase.com) la gran mayoría de las canciones está por encima de los 60 BPM. De esta forma nuestra aplicación tendrá un rango dinámico de ajuste de BPM entre los 60 y 200 BPM.

Vistas En la Figura 1 se pueden apreciar en la esquina superior izquierda tres iconos que representan, de arriba hacia abajo: **Iconos**

1. Número de pasos dados desde iniciar la aplicación
2. Distancia en metros recorrida
3. Tempo actual de carrera del usuario en pasos por minuto

A la derecha del Área de Vista se observa un anillo con un gradiente de color que se mueve entre verde, amarillo y luego rojo para simbolizar el tempo actual con que se está reproduciendo la canción. Dentro del anillo el valor en pasos por minuto que representa el tempo de reproducción actual de la canción en curso. Debajo, el factor de deformación ya sea incremento (o decremento) del tempo original de la canción en curso.

Manejo de sensores A partir de la API 19 los dispositivos Android cuentan con el acceso de sensores de detección de pasos (Smyth 2017)[21]. En particular, 2 de ellos son candidatos para su uso en nuestra aplicación ya que directamente son capaces de detectar posibles pasos del usuario utilizando el acelerómetro interno del dispositivo móvil.

Sensor Detector de Pasos (Step Dectector Sensor) El sensor del detector de pasos activa un evento cada vez que el usuario da un paso. Se espera que la latencia sea inferior a 2 segundos, su precisión es bastante baja y da muchos falsos positivos.

Sensor Contador de Pasos (Step Counter Sensor) El sensor contador de pasos proporciona la cantidad de pasos que ha dado el usuario desde el último reinicio del sistema siempre y cuando el sensor esté activo. El sensor contador de pasos tiene una mayor latencia (hasta 10 segundos), la ventaja de su eso está en la mayor precisión que este aporta en comparación con el sensor detector de pasos. Debido a que nos interesa tener un mayor grado de precisión para poder detectar el ritmo del corredor en tiempo real de forma aceptable, se ha utilizado el sensor contador de pasos.

Para recibir las actualizaciones de este sensor se ha implementado un **SensorEventListener** en la clase principal. Esta implementación exige la definición de dos métodos:

- **onSensorChanged:** Se ejecuta cuando el sensor lanza que ha detectado una cantidad X de pasos. La variable **event** de tipo **SensorEvent** contiene la información relativa al evento que incluye, entre otros datos, el instante de tiempo en el que fue lanzado el evento (**timestamp**) y la cantidad de pasos que ha dado el usuario desde el último reinicio del sistema hasta este momento.
- **onAccuracyChanged:** Este evento es llamado cuando el sensor sufre algún cambio de precisión, para este sensor en particular no se producen cambios de precisión.

Finalmente solo es necesario registrar nuestro **SensorEventListener** para que quede a la escucha de actualizaciones del Sensor Contador de Pasos. Para ello basta con hacer el registro a través de un **SensorManager**.

Estimación de la Distancia Recorrida Diversos acercamientos se han valorado a lo largo de los años para estimar la distancia recorrida por una persona a partir de la cantidad de pasos que esta persona ha dado. En su estudio sobre la relación entre la distancia de los pasos y las características morfológicas en adultos (Tripathy 2004)[24] analiza un total de 209 individuos y obtiene que la longitud media de los pasos es de 61.175 cm con un error estándar de 0.595 cm.

A partir de este hallazgo se puede implementar una sencilla estimación de la longitud recorrida por el usuario usando este valor como factor de conversión:

```
[fontsize=\scriptsize]
\Cantidad de pasos recorrida desde el inicio de la aplicación
steps = (int) event.values[0] - reset;

distance=61.175*steps/100.0;
```

Estimación del Tiempo y Adaptación de las Canciones El corazón de la aplicación está en su capacidad de hacer una estimación en tiempo real y a demanda del ritmo de carrera del usuario en BPM. Todas las piezas musicales, por variadas y únicas que sean, suelen contar un ritmo o tempo bien claro y marcado (New World Encyclopedia 2015)[6].

Las canciones utilizadas en el presente estudio han sido sometidas a un procesamiento inicial para el cálculo de su BPM base. Se ha utilizado un algoritmo basado en la detección de picos de energía para la estimación de la frecuencia de los pulsos del ritmo de la canción. Este algoritmo fue desarrollado como parte de mi Trabajo de Fin de Grado: Grado en Ingeniería Informática, Universidad de Salamanca (Acosta 2017)[1].

El algoritmo desarrollado se puede separar en unas 5 fases que se mencionan a continuación:

1. Cálculo de energías medias en intervalos de 1024 y 44100 muestras de audio.
2. Detección de picos de energía dentro de los intervalos.
3. Cálculo de intervalos de tiempo de separación entre picos de energía.
4. Cálculo de intervalos de separación con mayor presencia en la canción.
5. Estimación de BPM a través de la media aritmética de los intervalos con mayor frecuencia

Este algoritmo fue inspirado en el estudio realizado por (Patin 2003)[18] donde describe matemáticamente un método de estimación del tempo en una pieza musical a través del análisis de sus energías. En la Figura 2 se muestra la representación gráfica generada por la aplicación del proceso de detección de picos de energía que marcan el tempo de la canción (señalados con un borde azul).

Finalmente, una vez obtenido el tempo para cada una de las pistas de audio, la información fue añadida en los metadatos a través de la etiqueta ID3 (Baumann et al. 2003)[2] de los ficheros MP3 utilizando la aplicación Mp3tag.

La siguiente cuestión a resolver es ser capaces de detectar el tempo al que se mueve el usuario que porta el dispositivo móvil. Con esta información ya seríamos capaces de reajustar el tempo de la música para hacerlo coincidir con el

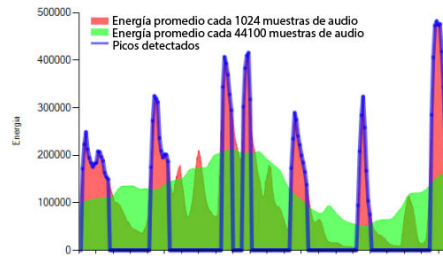


Figura 2: Cálculo de BPM

del usuario a través de un factor de conversión. Este factor de conversión puede ser alimentado al reproductor de audio del dispositivo móvil para que aumente o disminuya el tiempo.

Para estimar el tiempo al que se esta moviendo el usuario se han utilizado **dos mecanismos:**

- El cálculo del tiempo instantáneo (livebpm) que se muestra a la izquierda y arriba en el área de visualización en Figura 1. Con cada actualización del evento `onSensorChange` este valor es actualizado.
- Cálculo del tiempo promedio para un intervalo de 5 actualizaciones del evento `onSensorChange`. Utilizado para la funcionalidad de detectar el ritmo de carrera del usuario de forma más precisa.

Una vez calculado el tiempo instantáneo (livebpm), se procede a verificar si ha sido activado el modo automático teniendo en cuenta también que el usuario esté en movimiento. Esto es, no basta con que el usuario presione el botón para el modo automático, el sistema ha de detectar que el usuario está en movimiento (moviéndose a un ritmo entre 130 y 200 BPM) [25] para iniciar su cálculo del promedio del tiempo en 5 instancias del evento `onSensorChanged` (aprox. 5-10 segundos)

Una vez presionado el botón de control **AUTO**, el sistema actualizará la interfaz gráfica para mostrar el mensaje **RUN** (Figura 3) en el indicador de tiempo de canción. De esta forma se notifica al usuario de que ya puede empezar a correr. Entonces, de forma resumida, para el cálculo automático del tiempo para ajustarse al ritmo de usuario el sistema sigue los siguientes pasos, ver en Figura 4:

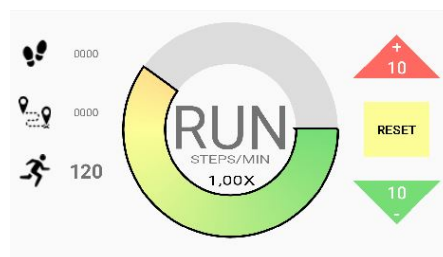


Figura 3: Activación del modo AUTO

Caso de uso

1. El usuario presiona el botón **AUTO** y el sistema registra el evento marcando una bandera lógica
2. El sistema espera a que se dé un tiempo instantáneo entre 130 y 200 bpm (usuario está en movimiento)
3. Al detectar un valor adecuado el sistema registra la cantidad de pasos (entra en la zona activa) Figura 4 en el momento y la marca de tiempo
4. Transcurridas 5 llamadas al evento `onSensorChanged` el sistema calcula el BPM promedio y lo asigna a canción actual en curso a través del reproductor.

La distorsión del tiempo se asigna a las canciones activas en forma de parámetro de reproducción. Para encontrar este factor de distorsión solo es necesario

dividir el tiempo de los movimientos del usuario con el tiempo de la canción que quedó anteriormente registrado en su etiqueta ID3.

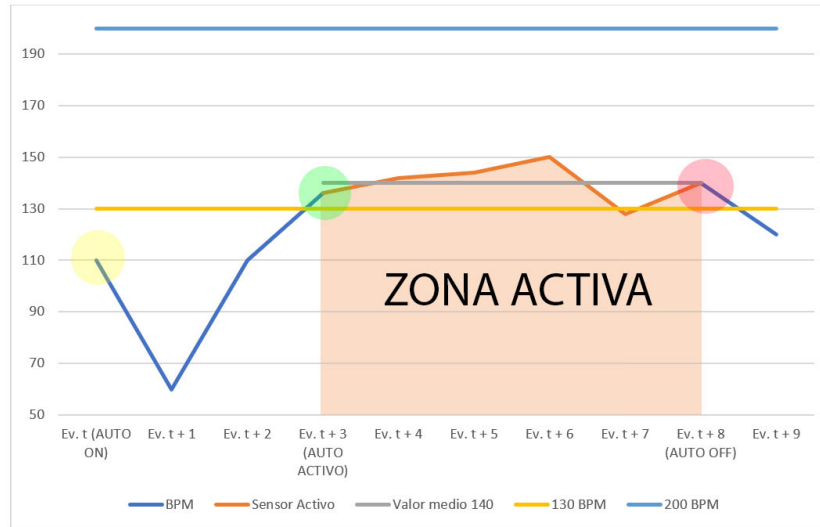


Figura 4: Diagrama de la zona activa de detección del tiempo del usuario

3.2. Participantes en el estudio

Los participantes en el estudio son un grupo de 12 jóvenes entre 19 y 33 años, 8 hombres y 4 mujeres. El grupo es de composición heterogénea en cuanto a la frecuencia en que corren. Del total de participantes 4 indicaron que corren unas 3 veces a la semana, 6 indicaron que corren al menos una vez a la semana y 2 que no corren todas las semanas. Los 12 participantes han indicado que suelen escuchar música mientras corren. Todos cuentan con un móvil Android.

3.3. Elementos de Encuesta

Las encuestas realizadas a los usuarios están compuestas de 8 preguntas. El cuestionario fue desarrollado a través de los formularios de Google (*Google Forms*) que facilitan la distribución de la encuesta a través de un enlace¹ directo.

Para determinar los límites aceptables de distorsión de las pistas de audio sin afectar su naturalidad percibida, los participantes han sido introducidos con el concepto de **radio de distorsión**, que caracteriza el factor multiplicativo de incremento o decremento al que ha sido sometida la canción para poder fijar un tiempo específico. Teniendo en cuenta que el tiempo promedio de las canciones modernas se encuentra entre los 90 y 120 BPM (Elias 2017)[14], se ha

determinado usar un radio de distorsión entre 0.5 y 2.0 para así poder cubrir cómodamente los límites de nuestra aplicación (entre 60 y 200 BPM)

Datos recogidos en la encuesta:

1. Edad y Sexo
2. La música me es indispensable para correr (1-5) (En total desacuerdo - Muy de acuerdo)
3. Suelo escuchar predominantemente este género musical cuando hago running (Reggaetón, Rock, Pop, Indie, Electrónica-Disco, Listas de Reproducción para Entrenamiento)
4. Me encuentro satisfecho|a con mi aplicación fitness habitual (1-5) (En total desacuerdo - Muy de acuerdo)
5. Me gustaría que mi app fitness habitual contara con un ajuste personalizado del tempo al igual que Tempo Running (1-5) (En total desacuerdo - Muy de acuerdo)
6. Radio de máximo aumento del tempo de una canción que te resulta natural, entre 1.0 y 2.0
7. Radio de máximo decremento del tempo de una canción que te resulta natural, entre 0.5 y 1.0

3.4. Procedimiento

Se les entregó a los participantes la aplicación Android en forma de apk firmada para su instalación. Posteriormente se les pidió que probarán la aplicación con las diferentes canciones disponibles con el fin de responder a la encuesta. Las canciones instaladas con la aplicación cubren un intervalo de 78 a 180 BPM. El BPM de cada canción ha sido leído directamente a través de su etiqueta ID3 para su indicación en pantalla y posterior procesamiento por la aplicación como se explicó con anterioridad.

4. Resultados

Un total de 12 individuos respondieron a todas las preguntas de la encuesta. De ellos 8 hombres y 4 mujeres con edades entre 19 y 33 años. La media de edad: 25.8 años.

4.1. Papel de la Música para los Corredores. La respuesta a la nueva iniciativa.

El 50% de los participantes siente que está muy de acuerdo con que el uso de la música le es indispensable para correr. En total solo el 16.3% de los participantes afirma que definitivamente no necesita de la música para realizar esta actividad física.

¹ Las apk son paquetes para el sistema operativo Android usados para instalar y distribuir componentes

El género musical más utilizado son las Listas de Reproducción para Entrenamiento con un 41.7 % de los participantes. Este resultado era previsible puesto que las listas de reproducción orientadas al entrenamiento gozan de gran popularidad. Suelen estar llenas de música energética y con un ritmo bien marcado y consistente que resulta atractivo y motivante. En muchos casos, el tempo base de las canciones, que suele ser generalmente 180 bpm, resulta no adecuado para muchas personas frenando su potencial rendimiento (Terry et al. 2012)[23].

El Reggaetón resultó ser el segundo estilo musical más usado siendo seleccionado por el 25 % de los participantes. Resulta interesante comprobar que el 66.7 % de los entrevistados manifestó no estar satisfecho con su aplicación fitness habitual. Por otro lado, la aceptación del nuevo enfoque adaptativo introducido por nuestra aplicación Tempo Running tuvo muy buena recepción, solo un 8.3 % mostró rechazo al a iniciativa de utilizar la nueva funcionalidad de tempo personalizado y adaptativo. La gran mayoría, un 83.4 % mostró interés en que se introdujeran estas funcionalidades en sus aplicaciones fitness habituales.

4.2. Límites Tolerables de alteración de Tempo

Como se comprueba en la Tabla 3, existe un potencial margen de ajuste para el tempo de las canciones sin que la aparición de artefactos o distorsiones apreciables ocurra para las pistas de audio. En particular se comprueba que de forma comparativa los usuarios son mucho más tolerantes al incremento del tempo que a su disminución. Destaca la similitud de los resultados con una tendencia musical moderna llamada **NightCore**, una forma de edición musical que consiste en acelerar su material fuente en un 15-30 % y aumentar su tono (Winston 2017)[27], (Wikipedia contributors 2018)[26].

Factor	Radio (media)
Límite máximo de incremento de tempo	1.26 (126 %)
Límite mínimo de decremento de tempo	0.85 (85 %)

Tabla 3: Límites tolerables de disminución y aumento del tempo

5. Discusión

Los resultados son increíblemente alentadores y confirman que existe una dimensión bastante desaprovechada en el mercado de las aplicaciones fitness. Se comprueba en consonancia con los resultados de ediciones musicales modernas como Nightcore (Winston 2017)[27] que las pistas musicales pueden ser aceleradas hasta un 30 % sin que sufran distorsiones serias, y que además el resultado podría ser incluso más llamativo para algunos usuarios (Wikipedia contributors 2018)[26]. En el futuro se podría añadir la funcionalidad Night Core a la aplicación para comprobar su recepción con los entrevistados.

Tengamos, por ejemplo, una canción que Enrique considera en extremo motivante, pero resulta que el tempo base es de la canción 120 BPM y a Enrique

le gustaría empujar sus límites hasta los 140 BPM en una carrera usando su canción favorita, bastaría con aumentar su tempo en la aplicación llegando a un radio de 1.17 (un aumento del 117 %) y listo. Ahora puede correr cómodamente siguiendo el ritmo de la canción con la tranquilidad de que mantendrá el paso y ganará motivación y mayor rendimiento como confirman (Lane et al. 2011)[13].

Es válido destacar que no todas las canciones son aptas para correr, si una pista de audio cuenta con un tempo base de 80 BPM o menos, resultaría muy difícil ajustarla para poder tener un ritmo consistente de carrera dentro de los 130 - 200 BPM sin que sufra de distorsiones serias.

La inmensa mayoría de los usuarios (83 %) mostró interés en esta nueva funcionalidad y en su inclusión en las aplicaciones fitness habituales. Al mismo tiempo el 66,7 % manifestó no estar satisfecho con los servicios que brinda su aplicación, insatisfacciones que podrían llevar al abandono apoyando los resultados del estudio de (Krebs et. al 2015)[12]

Estos hallazgos podrían indicar que un enfoque de mínima entrada de datos y alta personalización podría ser un paso en la dirección correcta. Para aumentos de rendimiento y motivación con el uso de música síncrona ajustada para cada individuo(Terry et al. 2012)[23], (Lane et al. 2011)[13].

6. Conclusiones

La investigación sugiere que muchas mejorías se pueden hacer aún en el mundo de las aplicaciones fitness. La mayoría de los usuarios siente la presión de una gran exigencia en la entrada de datos que contrasta con un bajo nivel de personalización en respuesta en sus aplicaciones fitness.

Existe un gran potencial en el uso de música síncrona como estimulante tanto psicológico como físico para lograr mayores rendimientos y hacer de la actividad física un momento más ameno. La personalización y ajuste adaptativo parecen llevar en la dirección correcta.

Investigaciones futuras podrían llevarse a cabo para comprobar los efectos en el estado emocional, motivación y rendimiento del uso de este ajuste personalizado y adaptativo del tempo de las canciones durante la carrera de los usuarios.

Agradecimientos. A mis tutoras por su apoyo incondicional a lo largo del desarrollo de este trabajo.

Referencias

1. Alejandro Acosta Vellón. Sistema para el análisis e interpretación de características en canciones. *Universidad de Salamanca*, 2017.
2. Stephan Baumann and Andreas Klüter. Super-convenience for non-musicians: Querying mp3 and the semantic web. In *ISMIR*, 2002.
3. Brian Benz and John R.Durant. *XML programming bible*, volume 129. John Wiley & Sons, 2004.

4. José Carlos Cortizo Pérez, Francisco Manuel Carrero García, Borja Monsalve Piqueras, Andrés Velasco Collado, Luis Ignacio Díaz del Dedo, Joaquín Pérez Martín, et al. Gamificación y docencia: Lo que la universidad tiene que aprender de los videojuegos. 2011.
5. Artur Direito, Yannan Jiang, Robyn Whittaker, and Ralph Maddison. Smart phone apps to improve fitness and increase physical activity among young people: protocol of the apps for improving fitness (aimfit) randomized controlled trial. *BMC public health*, 15(1):635, 2015.
6. New World Encyclopedia. Tempo — new world encyclopedia,, 2015. [Online; accessed 20-June-2018].
7. Mary Gowin, Marshall Cheney, Shannon Gwin, and Taylor Franklin Wann. Health and fitness app use in college students: a qualitative study. *American Journal of Health Education*, 46(4):223–230, 2015.
8. Chris Greenhalgh, Adrian Hazzard, Sean McGrath, and Steve Benford. Geotracks: Adaptive music for everyday journeys. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 42–46. ACM, 2016.
9. Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. Music personalization at spotify. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 373–373. ACM, 2016.
10. Costas I Karageorghis, David-Lee Priest, Peter C Terry, Nikos LD Chatzisarantis, and Andrew M Lane. Redesign and initial validation of an instrument to assess the motivational qualities of music in exercise: The brunel music rating inventory-2. *Journal of sports sciences*, 24(8):899–909, 2006.
11. Costas I Karageorghis, Peter C Terry, and Andrew M Lane. Development and initial validation of an instrument to assess the motivational qualities of music in exercise and sport: The brunel music rating inventory. *Journal of sports sciences*, 17(9):713–724, 1999.
12. Paul Krebs and Dustin T Duncan. Health app use among us mobile phone owners: a national survey. *JMIR mHealth and uHealth*, 3(4), 2015.
13. Andrew M Lane, Paul A Davis, and Tracey J Devonport. Effects of music interventions on emotional states and running performance. *Journal of sports science & medicine*, 10(2):400, 2011.
14. Elias Leight. How did pop music get so slow?, 2017.
15. Cameron Lister, Joshua H West, Ben Cannon, Tyler Sax, and David Brodegard. Just a fad? gamification in health and fitness apps. *JMIR serious games*, 2(2), 2014.
16. Robert Macdougall. The affective quality of auditory rhythm in its relation to objective forms. *Psychological review*, 10(1):15, 1903.
17. Matthias Mauch, Robert M MacCallum, Mark Levy, and Armand M Leroi. The evolution of popular music: Usa 1960–2010. *Royal Society open science*, 2(5):150081, 2015.
18. Frédéric Patin. Beat detection algorithms. *GameDev. net*, <http://archive.gamedev.net/archive/reference/programming/features/beatdetection>, 21, 2003.
19. Hairol Romero Sandí and M Sc Elvin Rojas Ramírez. La gamificación como participante en el desarrollo del b-learning: Su percepción en la universidad nacional, sede regional brunca. 2013.
20. Herbert Schildt. *Java: The Complete Reference*. McGraw-Hill Education Group, 2014.
21. Neil Smyth. *Android Studio 3.0 Development Essentials-Android 8 Edition*. Payload Media, Inc., 2017.

22. G Tenenbaum, R Lidor, N Lavyan, K Morrow, S Tonnel, A Gershgoren, J Meis, and M Johnson. The effect of music type on running perseverance and coping with effort sensations. *Psychology of sport and exercise*, 5(2):89–109, 2004.
23. Peter C Terry, Costas I Karageorghis, Alessandra Mecozzi Saha, and Shaun D'Auria. Effects of synchronous music on treadmill running among elite triathletes. *Journal of Science and Medicine in Sport*, 15(1):52–57, 2012.
24. Bijan Kumar Tripathy. A study on step distance and its relation with some morphometric features in adult male. *The Anthropologist*, 6(2):137–139, 2004.
25. Edith Van Dyck, Bart Moens, Jeska Buhmann, Michiel Demey, Esther Coorevits, Simone Dalla Bella, and Marc Leman. Spontaneous entrainment of running cadence to music tempo. *Sports medicine-open*, 1(1):15, 2015.
26. Wikipedia contributors. Nightcore — Wikipedia, the free encyclopedia, 2018. [Online; accessed 19-June-2018].
27. Emma Winston. Nightcore and the virtues of virtuality. *Brief Encounters*, 1(1), 2017.

Sistema de Seguimiento, Detección y Localización para Vehículos Autónomos

André Sales Mendes, Juan Francisco De Paz, Gabriel Villarrubia González,
Juan Manuel Corchado Rodríguez

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{andre, gvg, fcofds, corchado}@usal.es
<http://diaweb.usal.es/>

Resumen La robótica es una de las áreas más estudiadas dentro del campo científico, apareciendo constantemente nuevas formas de interacción con los robots. Una de ellas es el reconocimiento de gestos y el seguimiento automático del usuario. De igual modo la localización de interiores es un campo en el que se realizan continuamente estudios e investigaciones, con el objetivo de conocer la posición en tiempo real de un objeto o usuario en el interior de un edificio.

Se realiza una propuesta de un robot que sea capaz de seguir automáticamente a los usuarios haciendo uso de técnicas de visión artificial, teniendo la capacidad de ser localizado desde una plataforma web, haciendo uso de las tecnologías ya instaladas en él, como pueden ser las redes Wi-Fi o las balizas Bluetooth.

La propuesta, esta se aplica al caso de estudio de un carro de la compra de supermercado, que, aparte de permitir realizar un seguimiento del usuario facilitando el proceso de la compra a personas con movilidad reducida, es capaz de ser localizado dentro del supermercado. Además permite indicar al usuario en todo el momento los productos que tiene en la cesta y obtener la lista de la compra que se haya realizado anteriormente con una aplicación para ello.

Tanto el vehículo de seguimiento como el sistema de localización han superado pruebas de testeo y funcionalidad en las cuales se puede ver como el robot sigue al usuario y como, a su vez, este es localizado dentro del edificio desde la plataforma web.

Keywords: Seguimiento de Usuarios; Robots; Localización de interiores

1. Introducción

La robótica es una de las áreas más estudiadas dentro del campo científico, dentro de la cual constantemente aparecen nuevas técnicas de interacción del ser humano con ellos. El reconocimiento de gestos permite interactuar de forma remota sobre estos sin que sean comandados mediante una interacción física con un controlador.

Un tipo de interacción con los robots es el seguimiento automático de los usuarios, el cual tiene una amplia aplicación en diferentes entornos como puede ser la carga de productos, transporte de máquinas en entorno hospitalario o industrial, transporte de materiales peligrosos al contacto humano, etc.

La primera característica para que un robot sea capaz de realizar el seguimiento del usuario es que este debe ser capaz de obtener la posición relativa del usuario de modo que pueda desplazarse para seguir su trayectoria.

Para obtener la posición relativa en la cual se encuentra el usuario existen varios métodos, estos se pueden clasificar por métodos intrusivos y métodos no intrusivos. Los intrusivos son los que el usuario tiene que llevar encima algún dispositivo para que el robot le permita localizar. Los métodos no intrusivos son los cuales funcionan sin que el usuario tenga un dispositivo especial. Para esto la técnica más utilizada es la visión artificial.

En cuanto a los sistemas de localización de interiores se realizan constantemente estudios e investigaciones con el objetivo de conocer con exactitud la posición en tiempo real de un objeto o usuario, en este caso del robot, permitiendo ofrecer una variedad de servicios como puede ser la realización de rutas automáticas, guiar a los usuarios o gestionar flotas de robots dentro de un edificio.

Los sistemas de localización de interiores están en constante desarrollo debido a que el sistema de posicionamiento global, conocido el nombre de GPS, no está disponible en los interiores de los edificios debido a cuestiones físicas de la propia tecnología.

1.1. Objetivos

El trabajo tiene como objetivo el diseño de un vehículo o robot que sea capaz de realizar un seguimiento de los usuarios proporcionando una mayor movilidad al reducir la interacción gestual del usuario con el vehículo. Además, se tiene como objetivo la introducción de un sistema de localización en interiores dando a conocer en todo momento donde se encuentran los robots y a su vez los usuarios.

Las principales tareas que se persiguen con el trabajo son:

- Realizar un estudio del estado del arte de los diferentes vehículos de seguimiento de usuarios existentes hasta el momento, de las diferentes técnicas y algoritmos de visión artificial, realizar comparativas de las diferentes técnicas y tecnologías de localización de interiores.
- Elección del hardware para la construcción del robot que cumpla con los objetivos anteriores.
- Desarrollo e implementación del algoritmo de seguimiento teniendo en cuenta la tecnología y las técnicas elegidas.
- Definición de la arquitectura para el sistema de localización de interiores, que sea distribuida y que haga uso de tecnologías de bajo coste permitiendo obtener una precisión adecuada al problema.
- Creación de una plataforma web para la gestión de todo el sistema de localización que permita visualizar la posición en la que se encuentran los robots en tiempo real y los históricos de las rutas que estos hayan realizado.

- Aplicación de la propuesta, tanto del robot que sigue usuarios, como la localización de interiores a un caso de estudio real.

2. Estado del arte

En este apartado se tiene como objetivo realizar una recolección general de los diferentes sistemas de vehículos existentes que permitan el realizar el seguimiento de usuarios, técnicas de visión artificial para la detección de objetos y usuarios y los diferentes algoritmos para la extracción de características en imágenes. También se revisarán las técnicas de localización que permitan realizar el posicionamiento en interiores.

2.1. Vehículos de seguimiento autónomo

La localización del movimiento humano es una de las tecnologías más importantes y fundamentales en las aplicaciones prácticas de interacción del robot con los humanos.

En 2002 han desarrollado un robot que es capaz de seguir la trayectoria de un usuario el cual lleva encima un anillo de leds que emiten luz y que pueden ser fácilmente detectables por la máquinas que está situada en el robot [1] .

En 2010 se desarrolló un sistema para el seguimiento de personas para espacios que tengan condiciones de visibilidad reducida, como pueden ser espacios con elevada concentración de humo de incendio [2]. Durante su estudio realizan pruebas de detección con sonares láser y sensores de ultrasonidos. Tras los estudios que han realizado han comprobado que los ultrasonidos ofrecen mejores resultados en esas situaciones de luminosidad, para la localización del usuario le dotan de un anillo emisor colocado en una de las piernas, mientras que en el robot instalan múltiples receptores de ultrasonidos, permitiendo de esa manera realizar una triangulación con la diferencia de los tiempos de llegada. Los sensores en la plataforma y el anillo se pueden ver en la Figura 1.

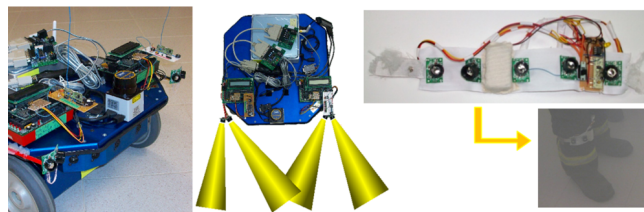


Figura 1: sensores de ultrasonido y anillo emisor

En 2012 desarrollan un robot que pueda realizar un seguimiento de una persona, manteniendo una cierta distancia al usuario objetivo, basando el funcionamiento del sistema en el uso de sensores de infrarrojos [3]. El motivo por el cual

han usado estos sensores es que los valores de las medidas son prácticamente inmutables a los cambios de radiación infrarroja del medio ambiente pero, en cambio, si que son bastante sensibles a los cambios de la radiación inducida por el movimiento humano.

Para la detección de movimiento usan un conjunto de sensores infrarrojos en la parte superior de la plataforma robótica. Mediante el reparto del área frontal del robot en 8 zonas, y haciendo uso de los sensores que el sistema es capaz de posicionar al usuario en una de esas zonas en cada instante y ver como se desplaza entre ellas. Este sistema se caracteriza por ser capaz de ofrecer un tiempo de respuesta bastante bajo, frente a sistemas que usen técnicas de visión artificial debido a que los costes computacionales de sus cálculos son menores. En contrapartida este sistema no tiene la capacidad de identificar a diferentes usuarios.

En 2014 desarrollan un robot que permite la iteración con el usuario mediante gestos, en su desarrollo hacen uso se una cámara con sensor de profundidad, la Kinect de Microsoft, que permite obtener la posición de los usuarios [4].

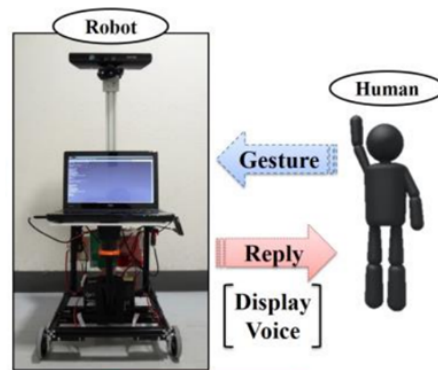


Figura 2: robot de seguimiento con Kinect

Mediante diferentes gestos pueden controlar el robot, indicándoles, que siga al usuario, se mueva, Figura 2.

2.2. Visión artificial

La visión artificial se puede definir como un campo de la inteligencia artificial que permite la obtención, procesamiento y análisis de cualquier tipo de información especial en imágenes digitales. Los sistemas de visión artificial están basados en la visión humana, pero a la falta de entender cómo llevan a cabo los procesos del cerebro, los sistemas artificiales han resultado imprácticos y ha llevado a que la visión artificial se basara en métodos y algoritmos matemáticos.

Las principales aplicaciones de la visión artificial en robótica actualmente son la identificación e inspección de objetos, determinación de la posición de objeto en el espacio, establecimiento de relaciones entre objetos, etc.

Imagen digital: en visión artificial la captación de imágenes es realizada digitalmente. Estas permiten realizar un tratamiento de estas haciendo uso de diferentes técnicas para el descubrimiento y el resaltado de la información que esta contenga. La imagen digital puede ser clasificada en vectorial o en mapa de bits. La imagen vectorial está descrita por formulas, su característica principal es que es escalable sin la pérdida de calidad, mientras que la imagen basada en mapas de bits está descrita mediante pixeles. En cada uno de los pixeles se guarda la información de ese punto.

Reconocimiento objetos: para llevarlo a cabo es necesario realizar un procesamiento de las imágenes. donde se elimina la mayor parte de la información para realizar el reconocimiento, extrayendo únicamente las características para realizar una distinción de los otros elementos de la imagen. El proceso de reconocimiento consiste en la interpretación de los parámetros seleccionados en el procesado, esto se realiza mediante una comparación de los modelos extraídos de la imagen que se está analizando con la del modelo obtenido anteriormente, es un proceso de elevado coste computacional.

El **reconocimiento basado en marcas de referencias** hace uso de un objeto que se encuentra en el campo de visión de la imagen, proporcionando un punto de referencia y unas dimensiones que facilitan el proceso de reconocimiento. Este método ha sido presentado en 1999 por Kato, para encontrar la posición de la cámara en relación a un objeto [5], mientras que Stricker lo usa para determinar las coordenadas de un en tres dimensiones de un objeto cuadrado [6].

El **reconocimiento basado en puntos de interés**, en visión artificial, hace referencia a la localización de puntos relevantes en cuanto a la cantidad de información del entorno. Estos deben ser estables a perturbaciones y transformaciones que puede sufrir la imagen. La característica más importante es que el sistema sea lo más coherente posible detectando en instantes consecutivos el mismo punto característico en caso que se encuentre en el foco de visión. Las fases del reconocimiento son:

- **Extracción:** se basa en la búsqueda de puntos en la imagen con diferente apariencia a las de su alrededor, como puede ser los bordes, esquinas, o zonas más brillantes u oscuras. Los algoritmos más usados en esta fase es el detector de esquinas de Harris [7], el Features From Accelerated Segment Test (FAST) que obtiene los pixeles con valores máximos o mínimos en relación a los pixeles vecinos [8].
- **Descripción:** se calcula el vector que describe las características de los puntos que se han extraído en la fase anterior para que sean comparados posteriormente con otros puntos de interés. Estas descripciones pueden ser divididas en histograma de gradientes o en prueba binaria.

El histograma de gradientes se calcula a partir de la cuantificación de los gradientes de un área local. Algunos de los algoritmos que lo usan son Scale Invariant Feature Transform (SIFT) [9], Speeded Up Robust Features (SURF) [10].

La prueba binaria se basa en la realización de comparaciones de la intensidad de los dos píxeles obteniendo como resultado un binario indicando cuál de ellos es más brillante. Los algoritmos más conocidos que usan pruebas binarias son Binary Robust Independent Elementary Features (BRIEF) [11], Binary Robust Invariant Scalable Keypoints (BRISK) [12] y Oriented FAST and Rotated BRIEF (ORB) [13].

- **Búsqueda de coincidencias:** se compara el vector de características de la imagen que se está actualizando con los que se hayan almacenado anteriormente y clasificado anteriormente. Esta búsqueda es imposible que sea realizada en tiempo real debido a las dimensiones de los algoritmos como puede ser el SIFT[9], como alternativa a ello se usan algoritmos como el del vecino más cercano [14].

Un **descriptor de características** debe tener la capacidad de buscar correspondencias entre las imágenes a analizar y las analizadas previamente. Para que se considere un descriptor de características este debe ser reproducible, robusto y rápido. Con ser reproducible se indica que debe ser capaz de encontrar los mismos puntos de interés en diferentes iteraciones y diferentes condiciones de visibilidad, siendo fiable y de alta precisión. Con el termino robusto, indica la capacidad de identificar el mismo punto entre imágenes consecutivas obtenidas con una cámara, aunque existan cambios de iluminación o ruido. Por fin el descriptor debe ser rápido, es decir, que el sistema debe ser capaz de extraer los puntos y compáralos con una base de datos en el menor tiempo posible y si puede ser, en tiempo real.

Los algoritmos más usados para la descripción de características son el SIFT[9] y el SURF [10].

El algoritmo **SIFT** [9] fue propuesto por David Lowe en 1999, las características más importantes es que el algoritmo es invariante a cambios de escala, rotaciones, cambios de iluminación lineales, borrosidad y ruido.

El algoritmo **SURF** [10] es uno de los más utilizados para la extracción de puntos de interés en imágenes [15]. La extracción se realiza detectando en primer lugar los puntos de interés y su localización dentro de la imagen.

Tracking: un problema, que puede ser definido como un problema de la estimación de la posición y la trayectoria que están realizando los humanos o van a realizar en momentos cercanamente futuros. El tracking consiste en encontrar correspondencia de puntos en dos frames seguidos. Para la realización del tracking existen dos técnicas:

- **Teniendo en cuenta un objeto de referencia:** en cada instante el tracking es realizado calculando a partir de la posición que tiene el objeto que se pretende realizar el tracking en relación a la referencia que se conoce la posición.

- Teniendo en cuenta la posición inicial del objeto: las posiciones de tracking que se obtienen son relativas a la posición inicial del objeto.

Flujo óptico: hace referencia al movimiento aparente de los patrones de intensidad en una imagen. El movimiento de los objetos implica un movimiento en los patrones de intensidad en el plano de la imagen, el flujo óptico puede relacionarse directamente con el movimiento de los objetos de escena [16]. Las técnicas para su estimación se clasifican en la comparación de regiones, las basadas en fase y las basadas en energía. En todas ellas para la estimación se parte de los niveles de gris que permanecen constantes ante los movimientos espaciales en un instante dado.

Cámaras: son el elemento esencial en visión artificial, encargados de realizar la captura de una escena y almacenar una imagen de modo que pueda ser tratada por los métodos descritos anteriormente.

Visión estereoscópica es el proceso que se realiza de manera natural cuando un observador mira simultáneamente dos imágenes de un mismo objeto que han sido captadas desde distintas posiciones. Cada ojo ve una imagen y el resultado de ese proceso es la percepción de la profundidad o las tres dimensiones. [17]. Para lograr percibir las 3 dimensiones se hace uso las sombras, tamaño relativo o mediante la perspectiva. El ojo humano para la observación de un objeto realiza movimientos de acomodación y convergencia que se producen de manera simultanea para garantizar la comodidad en la observación.

Los pasos a seguir de visión espectroscópica son la adquisición de las imágenes, el modelado de la cámara, la extracción de características y obtener la correspondencia entre las imágenes y por ultimo determinar la distancia de profundidad.

Kinect de Microsoft: inicialmente ha sido pensado como controlador de juego, pero gracias a los componentes que lo integran: sensor de profundidad, cámara RGB, arreglo de micrófonos y sensor de infrarrojos, es capaz de capturar el exoesqueleto humano, reconocerlo y posicionarlo en el plano. De esta manera permite a los usuarios interactuar con la consola sin necesidad de tocar ningún controlador de juego físicamente. Aunque el dispositivo haya sido desarrollado para el mundo de los videojuegos, ha tendido un gran éxito fuera de ese mundo, como puede ser escáneres 3d [18] o creación de entornos virtuales [19]. Lo que ha llevado a cabo que se use en la multitud de proyectos es la aparición de kits de desarrollo completos como es el SDK Kinect de Windows [20] y OpenNI [21].

Los sensores principales de sensor son la cámara RGB que es capaz de operar con varios formatos de imagen, el sensor de profundidad usa luz estructurada infrarroja en su funcionamiento y el array de micrófonos. Con este conjunto de sensores se puede obtener el conjunto de puntos que forman el esqueleto de los usuarios en un espacio tridimensional. En la figura 3, se puede ver un ejemplo de los puntos que forman un esqueleto obtenido a través de la Kinect.

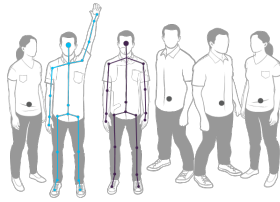


Figura3: Esqueleto obtenido con el sensor Kinect

2.3. Sistemas de localización interiores

El termino de localización hace referencia a la determinación del lugar en que se encuentra una persona o cosa. Las investigaciones sobre sistemas de localización de interiores se realizan debido a la incapacidad que tiene el sistema de posicionamiento global (GPS) para la realización de localizaciones en ese contexto.

Antes de abordar las técnicas y los algoritmos de localización se especifican algunas características de los sistemas de localización que son independientes de la tecnología o algunas técnicas que estos utilicen.

- Posición física y simbólica: la posición física es la proporcionada, por ejemplo, por un sistema GPS, mientras que una simbólica puede ser en Salamanca, en el salón o en la cocina.
- Posición absoluta frente a relativa, en un sistema de posicionamiento absoluto usan un sistema de rejilla de referencia compartida para todos los objetos a ser localizados. Mientras que un sistema relativo usan un sistema de coordenadas propio.
- Privacidad del cálculo de la localización: dependiendo de quien realiza los cálculos, en el cliente o en el servidor, se pueden clasificar como un sistema privativo o un sistema que pueda ser rastreado.
- Exactitud y precisión: un sistema de localización debe ser capaz de situar a un objeto con exactitud y precisión constante. Cuando se indica que un sistema es más preciso que otro se indica que el primer obtiene menos error en las estimaciones.
- Área de localización: hace referencia a la zona que el sistema es capaz de dar cobertura y a su vez ofertar localizaciones.
- Variedad de costos: costos relacionados con varias entidades, como puede ser tiempo de instalación, infraestructura, costos futuros como ampliación de la zona o de mantenimiento.
- Limitaciones: están definidas por el entorno en que se encuentran, en los sistemas basados en GPS pueden existir zonas ubicaciones donde la señal de GPS sea nula o no exista el número de satélites necesarios. En sistemas basados en radiofrecuencia pueden estar limitados por las paredes de los edificios.

De las características anteriores, cuando se trata de un sistema de localización de interiores el tipo de posiciones más importante es saber donde se encuentra el usuario, en el salón, cocina que unas coordenadas, en cuanto al cálculo de posición se suele distribuir entre los diferentes nodos de modo que no exista una carga elevada en algunos de ellos. En cuanto a la precisión y exactitud para ciertos casos basta saber la zona o la habitación donde se encuentra, lo que hace referencia a la área de localización que suele estar establecida en un edificio o un conjunto de estos, sobre los cuales se pretende realizar la localización.

Técnicas de posicionamiento: a continuación se verá un resumen de las técnicas de localización basadas en la realización de características de señales electromagnéticas para la estimación de la posición.

- Basadas en la detección de la proximidad: se basa en la conectividad entre los dispositivos, la posición que ofrece es una posición simbólica relativa a la estación base. La posición que indican este tipo de sistemas es la que tiene la estación base de origen[22].
- Basadas en análisis del entorno: también llamada fingerprint, se basa en la realización de mediciones de las fuerzas de las señales, que se están emitiendo de los diferentes puntos, en un punto determinado. Usando las fuerzas en conjunto con diferentes algoritmos se puede obtener la posición del objeto [23]. Esta técnica consta de dos fases, la primera es una fase de calibración, en la cual se genera una base de conocimiento y una segunda, la fase de operación, donde se calculan las localizaciones.
- Basadas en triangulación: para determinar la posición hace uso de las distancias que obtiene desde puntos de referencia. Para que se lleve a cabo satisfactoriamente se deben tener 3 puntos de referencia. Estas técnicas se basan en:
 - Dirección: llamada ángulo de llegada, se basa en el uso de antenas que son sensitivas a la dirección del receptor de manera que se puede obtener la dirección de la onda de llegada desde el emisor[24].
 - Distancia: para el cálculo de la distancia están basados en el tiempo de transcurso de la onda o en alguna propiedad física que permita la orientación de la distancia.

Las técnicas basadas en tiempo pueden ser tiempo de llegada, diferencia del tiempo de llegada, y tiempo de retorno de la señal.

Las que se basan en propiedades de la señal, suelen estar basadas en la atenuación de la señal, la estimación se lleva a cabo a partir de medidas de potencias recibidas, mediante el cálculo de pérdidas de propagación. Estos cálculos se ven afectados por los efectos del multitrayecto de las ondas reduciendo la precisión.

Tecnologías de posicionamiento: Las tecnologías más usadas para el posicionamiento de interiores suelen ser las que basan su funcionamiento sin cables.

El Bluetooth de baja energía es una de las más usadas debido a que su tecnología se basa en el bajo consumo, es bajo en latencia, tiene altos niveles de seguridad y unos costes reducidos.

El WIFI, esta es la que se encuentra con mayor disponibilidad a nivel de puntos de acceso en la mayoría de edificios modernos, puede ofrecer localización de bajo coste de instalación [25], la desventaja de esta frente a Bluetooth de baja energía es el consumo.

La navegación inercial, en su funcionamiento hace uso de sensores de movimiento del dispositivo, como pueden ser el acelerómetro y el giroscopio. Esta tecnología requiere que se realice una localización inicial, pero el error que se genera es acumulativo y debe ser corregido en intervalos periódicos con una tecnología externa al sistema.

Los ultrasonidos también son usados para localización, con esta se pueden obtener precisiones de centímetros. La técnica usada es la de tiempo de llegada, que se basa en una red de sensores fijos en el techo y el usuario lleva un emisor de ultrasonidos encima.

Los sistemas basados en infrarrojos se utilizan en dispositivos que transmiten datos a través de esta tecnología, una de las características importantes es que la luz infrarroja no atraviesa paredes, ofreciendo una mayor precisión que las ondas de radio frecuencia.

La transmisión de datos mediante iluminación LED se llama LI-Fi. Una particularidad de esta iluminación LED es que la onda de luz puede ser modulada a alta velocidad permitiendo la transmisión de datos. Esta modulación es capturada a través de la cámara de los dispositivos móviles [26], permitiendo conocer la posición con un alto grado de precisión.

La tecnología GPS Pseudolites, tiene una orientación industrial, la cual se basa en realizar un despliegue de una constelación virtual de satélites en interiores, se logran precisiones de centímetros, pero es una tecnología de elevado costo tanto iniciales como para la ampliación de los dispositivos ya que son especiales.

La identificación por radio está altamente extendida. Su principal orientación es la gestión de productos, distribución o presencia, Se utilizan en soluciones de proximidad o presencia.

Algoritmos de posicionamiento basados en el reconocimiento de patrones: son los responsables de encajar el vector de potencias recibidas, que proceden de diferentes estaciones base, con una muestra que se haya obtenido anteriormente. Estos algoritmos se usan en las técnicas de análisis del entorno.

- Vecinos más cercanos (KNN) ha sido usado en los últimos años para el desarrollo de sistemas de localización de interiores. Es una técnica de clasificación de los K vecinos más cercanos del vector de distancias de las potencias RSS que se han obtenido en el punto que se encuentra. Los pesos normalmente en estos sistemas toman el valor de 1 ya que la estimación es un promedio de todas las medidas.
- Bayesianos: en el estudio nombrado Localización de personal en entornos interiores con tecnología RFID [27] usan algoritmos de redes bayesianos que

modelan la posición del usuario en un instante determinado con una variable aleatoria que se obtiene a partir de las probabilidades obtenidas hasta el instante actual.

- **Redes neuronales artificiales:** estas son usadas para la clasificación de conjuntos, las entradas del preceptrón multicapa, es cada una de las potencias recibidas de cada estación base, y la salida es la posición que estima. La desventaja es el tiempo de entrenamiento y que que no se realicen sobreajustes provocando resultados erróneos a la hora de usar el sistema.
- **Máquinas de soporte vectorial:** con este algoritmo se procesan los vectores de las muestras con el fin de obtener un hiperplano que separe linealmente las observaciones permitiendo obtener la localización mas fiable posible.

3. Arquitectura propuesta

Tras la realización del estudio de cómo se encuentra el estado del arte, para la propuesta se deben tener en cuenta las siguientes funcionalidades:

- **Seguimiento del usuario:** el vehículo o robot debe ser capaz de realizar un seguimiento del usuario, tanto por interiores como por exteriores, con la posibilidad de esquivar obstáculos y otros usuarios que se puedan encontrar en su trayectoria.
- **Localización de interiores:** el sistema debe ser capaz de realizar una localización de interiores, de modo que se pueda conocer cuáles han sido las rutas que han realizado los usuarios, para la realización de estudios de esas rutas o para el realizado de guiados de interiores.

La propuesta para la realización del seguimiento del usuario se basa en un robot que tenga la capacidad de realizar el seguimiento haciendo uso de técnicas de visión artificial. El sistema debe ser capaz de identificar a los usuarios para la realización de la sincronización y conocer su posición de modo a seguir su trayectoria.

Debido al tamaño del sistema se propone la realización de un sistema modular. De esa manera los módulos pueden ser usados de forma independiente, todos en conjunto o solamente una combinación finita de ellos.

Módulo de soporte: representa la estructura base sobre los cuales van todos los componentes electrónicos. La estructura se puede ver en la figura 4, los materiales propuestos para su construcción son tubos metálicos de 2cm, de modo que pueda ser usada en varios entornos.

Módulo central: responsable de unir los diferentes módulos del sistema. Para ello se propone usar un Tablet que tenga como sistema operativo Windows ya que es necesario para dependencias que tienen los siguientes módulos.

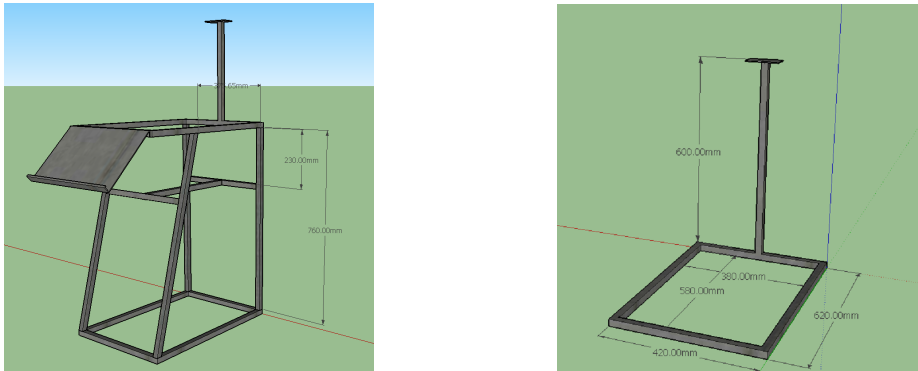


Figura 4: Estructura base del robot

Módulo de seguimiento: Encargado de realizar la sincronización con el usuarios y obtener en cada momento la posición de este. Como sensor para la captura de imágenes se propone usar el sensor Kinect de Microsoft ya que mediante sus librerías permite posicionar al usuario fácilmente en un espacio tridimensional, abstrayendo toda la capa de implementación de bajo nivel. Para la sincronización del usuario se propone realizarlo mediante un gesto, como puede ser mantener levantados los brazos por un instante predefinido de tiempo.

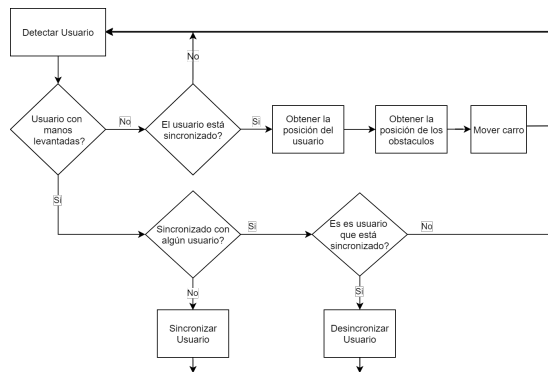


Figura 5: Algoritmo de seguimiento

El algoritmo de seguimiento es el que se puede ver en figura 5. El paso inicial es la identificación de los usuarios, seguidamente la comprobación de que se está realizando el gesto de sincronización o desincronización, en caso de que lo esté realizando y no esté sincronizado con ningún usuario se sincroniza con ese usuario. En el caso que sea el propio usuario sincronizado el que realiza el gesto se desincroniza, en otro caso se descarta. En el caso que el usuario no esté

realizando ningún gesto y coincida con el usuario sincronizado el sistema obtiene la posición del usuario, verifica que no existen obstáculos en la ruta que debe seguir el robot, y en el caso que sea segura la trayectoria se mueve.

Módulo de detección de obstáculos: debe ser capaz de indicar que se encuentran obstáculos en la trayectoria que sigue el robot o en los laterales. Dentro del campo de visión de la Kinect se propone usar el mismo sensor que se usa en la identificación de los usuarios, ya que ofrece una nube de puntos con la distancia de profundidad a cada uno de ellos. Para la detección de los obstáculos laterales se propone usar los sensores hc-sr04 ya que su funcionamiento de basa en propiedades de los ultrasonidos y tienen un rango de 2 a 250cm.

Módulo actuador: es el encargado de dar movimiento al robot. Debe conectarse con el módulo central para recibir las ordenes de movimiento y al mismo tiempo usar los motores de corriente continua para dar movimiento al robot. Como microcontrolador se propone usar el Arduino nano ya que cumple con requisitos. Para la regulación de la intensidad y la dirección de giro de los motores se propone usar los drivers de motores IBT_2.

Módulo localización: responsable de la realización del posicionamiento en interiores. Se propone que la arquitectura haga uso de las redes WI-FI en conjunto de balizas Bluetooth, ya que las redes WIFI se encuentran instaladas en la mayoría de los edificios y que las balizas Bluetooth son de bajo coste. Otra ventaja de estas tecnologías es que no se necesita de ningún adaptador especial para la obtención de estas señales.

Como técnica se propone el fingerprint, ya que es el que mayor precisión ofrece. Esto se debe a la calibración previa del entorno donde se mide las potencias recibidas en cada uno de los puntos por donde se mueve el usuario.. Consta de dos fases:

- Fase de calibración: fase en la cual se genera una base de conocimiento con los valores de las fuerzas de las señales que emiten las estaciones base en las diferentes posiciones siendo capturadas por el dispositivo.
- Fase de operación: el dispositivo que pretende realizar la localización envía al servidor los valores de potencia capturados en ese momento. El servidor los usa junto a los obtenidos en la fase de calibración para la obtención de la localización.

Para que no sea necesario realizar la fase de calibración en cada uno de los robots, se ha decantado por realizar una arquitectura distribuida, es decir, en la cual la base de datos y los servicios de obtención de la localización se encuentren en un servidor remoto y todos los robots realizan las peticiones a ese servidor.

La arquitectura del sistema de localización es la que se puede ver en la figura 6 , está constituida por un servidor que contiene la base de datos y el algoritmo de clasificación. El servidor debe contener los servicios de configuración de mapas

mediante una interfaz web que pueda ser accedida desde un navegador. Por último, debe permitir a los robots móviles realizar la calibración y la consulta de predicciones de la posición en la que se encuentra.

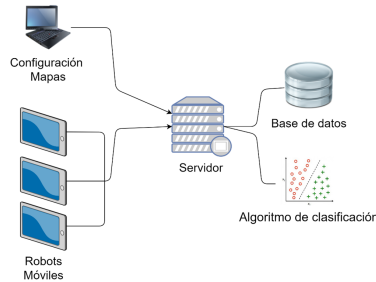


Figura 6: Arquitectura sistema de localización

4. Caso de estudio

La propuesta presentada ha sido aplicada al caso de estudio de un carrito de supermercado, el cual se puede considerar como caso óptimo sobre el cual se puede aplicar toda la propuesta.

El seguimiento del usuario permite que el usuario se desplace por el supermercado sin tener que realizar la tarea de empujar el carro de la compra, esto puede parecer una tarea sencilla, pero si se piensa en el caso de un usuario que esté en una silla de ruedas o esté usando las muletas esta tarea sencilla puede complicarse.

La localización de interiores en un supermercado permite guiar a los usuarios durante el proceso de compra. También permitirá realizar recomendaciones dependiendo de la posición que se encuentre o realizar estudios de cuáles son las rutas más comunes entre los usuarios para la aplicación de técnicas de marketing.

4.1. Carro de la compra

En cuanto a **hardware** se puede decir que la estructura es bastante parecida a la propuesta. En cuanto a la tablet usada en el módulo central es una Tablet Surface de Microsoft con un procesador i5 y 4GB de ram. Como añadido a la propuesta se han incorporado sensores RFID que permiten saber cuáles son los productos que se encuentran dentro de la cesta de compra, en total se han incorporado 3 sensores, uno en el interior de la cesta, para detectar los productos que se añaden, uno en el lateral para ver los productos que se quitan de la cesta, y uno en la parte frontal para ver la información sobre los productos.

La apariencia final que tiene el carro de la compra con todos los componentes es la que se puede ver en la figura 7, aparte de los módulos descritos se puede ver también un adaptador USB e WIFI que es usado para la obtención de la potencia de las señales, ya que el interno es usado para la realización de la comunicación con el servidor.



Figura 7: carro de la compra con los componentes electrónicos

La **aplicación que se encuentra ejecutando en la tablet** tiene como funcionalidad principal ver la información de los productos de la cesta de la compra, figura ?? en ventana de la izquierda, realizar la sincronización y desincronización, ver la lista de la compra y realizar el calibrado del sistema de localización.

El estado de la sincronización de los usuarios se ha representado con los diferentes colores. Si el usuario se encuentra desincronizado color es rojo, si es el usuario que está seleccionado entonces verde, azul en el caso del usuario sincronizado y por fin blanco de desincronizado. La ventana de sincronización se puede ver en la figura ??.



Figura 8: Aplicación carro de la compra

Una vez que se había desarrollado el carro de la compra, durante las pruebas en el supermercado, apareció la **necesidad de la existencia de una lista de compra en la pantalla de la aplicación que se ejecuta en la tablet**

del carro, con los productos que el usuario necesitaba comprar. Para ello se ha desarrollado una aplicación que permitiera realizar la lista de la compra en casa y que cuando el usuario esté utilizando el carro de la compra pueda usar esta lista.

La aplicación desarrollada ha sido realizada para la plataforma Android. Es una simple aplicación que permite el control de una lista mediante comandos de voz, al mismo tiempo que permite enviar al carrito de la compra la lista una vez que esta se haya realizado.

Debido a que se han añadido los sensores RFID para la identificación de los productos, la aplicación que se ejecuta en el carro debería obtener la información desde una fuente de datos. Para ello se ha creado una **plataforma web de gestión**, figura 9, donde se pueden registrar los productos por categorías y al mismo tiempo ver las ventas que se han realizado haciendo uso del carro de la compra pudiendo de este modo realizar estadísticas sobre los productos más vendidos.

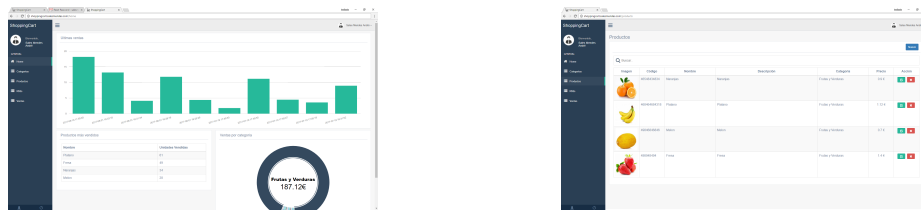


Figura 9: Plataforma web de gestión de productos

4.2. Sistema de localización

Para el **sistema de localización**, aunque la tablet ya contuviera incorporado un receptor WIFI, se ha añadido uno externo de manera que se pudiera realizar al mismo tiempo escaneos y que estuviera conectado al servidor remoto tanto para el sistema de localización como para la obtención de la información sobre los productos.

En la plataforma web se pueden, figura 10, dar de alta modificar e eliminar los dispositivos a localizar, edificios con sus diferentes plantas, ver la localización en tiempo real de carros de la compra y por fin ver un histórico de los recorridos que han realizado los usuarios.

Aparte de la plataforma web se ha creado una **API** donde se conecta la aplicación que se ejecuta en el carro de la compra para la realización del calibrado del sistema y para la obtención de la localización.

Las balizas Bluetooth que se han utilizado son las IBKS 105, figura 11, de AccentSystems, que usan el protocolo Ibeacon o Edystone.

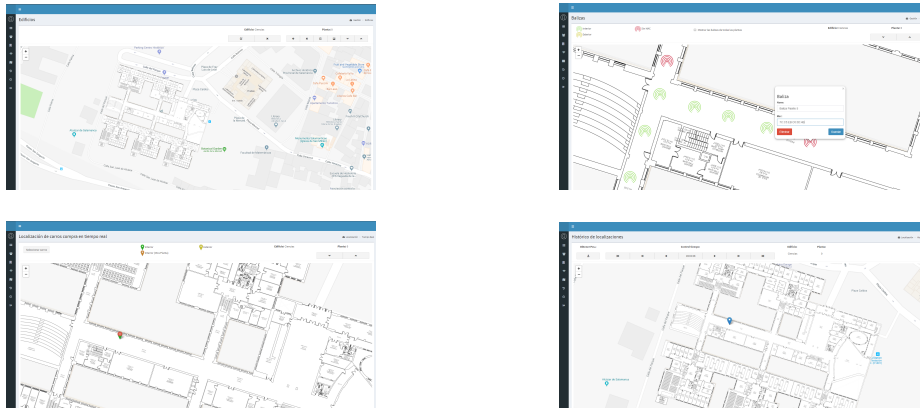


Figura 10: Plataforma web localización



Figura 11: Beacon ibks 105

5. Resultados

Para la comprobación de los resultados se realizó un proceso heurístico a través de diferentes circuitos de prueba con diversos objetivos y posteriormente una prueba en un entorno de aplicación real.

Las pruebas de laboratorio han consistido en una batería de test en las cuales diferentes usuarios realizan el proceso de sincronización, e desincronización en conjunto por cada una de las 3 rutas que se han definido.

Los test se propusieron como un proceso de desarrollo iterativo con un total de 20 usuarios, de manera tras ellos se obtuvieron modificaciones sobre los aspectos estudiados.

De esta manera, en el primer test el objetivo fue validar un correcto funcionamiento del carro de la compra, en un espacio amplio, y con elevada iluminación, para ello se ha usado la ruta con el color amarillo de la figura 12. Con esta prueba se llegó a la conclusión que el robot tenía la capacidad de realizar el se-

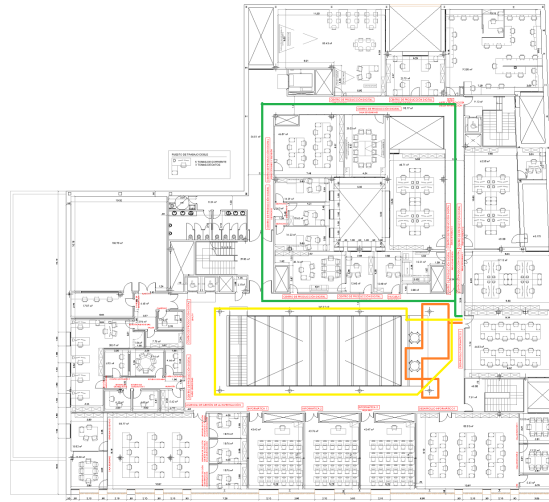


Figura 12: rutas de prueba

guimiento, pero en este punto se han detectado dificultades con algunos usuarios para realizar la sincronización mediante gestos, con esto se ha llevado a cabo la implantación del nuevo método que usa el botón de la interfaz gráfica para ello.

En el segundo, el circuito diseñado tenía como objetivo evaluar la validación del desplazamiento entre pasillos, para la simulación de los desplazamientos en los supermercados entre las estanterías, en la figura 12 se encuentra representada de color verde. Una vez obtenidos los resultados, se detectó que el robot debería mejorar en cuanto a la posición de las ruedas, debido a que en algunas situaciones colisionaban con alguna esquina o al pasar por las puertas. Como no era posible modificar fácilmente la estructura se han aumentado los márgenes laterales, logrando que el carro haga los giros con una mayor amplitud, pero evitando las colisiones.

El último circuito de prueba realizado el objetivo era la evasión de los objetos y personas que se pudieran interponer en la ruta del carro, para ello se ha usado la ruta de color naranja de la figura 12. En esta última prueba no se han detectado anomalías en el funcionamiento debido a los ajustes realizados anteriormente.

Las pruebas de entorno real han sido realizadas en un supermercado real figura 13, una vez realizadas las de laboratorio, estas se pueden clasificar como pruebas de validación del prototipo.

Durante estas pruebas se han obtenido una gran adhesión por parte de los clientes, especialmente personas mayores, se ha detectado el siguiente problema:

- La interferencia del refresco del sistema de iluminación compuesto por fluorescentes, estos tenían una frecuencia de 50hz y la cámara de Kinect 30hz, esto lleva a cabo que el sistema descincronizara al usuario automáticamente.



Figura 13: pruebas entorno real

te. La solución ha pasado por actualizar el sistema de iluminación a nuevas tecnologías.

Para la validación del sistema de localización de interiores, las pruebas han sido realizadas en el edificio I+D+I de la Universidad de Salamanca. En cuanto a las redes WI-FI se han usado las que se encuentran preinstaladas en el edificio mientras que para el Bluetooth se han instalado en total de 56 Beacons.

La configuración de la frecuencia de escaneo usada durante las pruebas ha sido de aproximadamente 4.6 escaneos por segundo para el Wi-Fi, mientras que para los escaneos Bluetooth han sido 1.6 veces por segundo. La configuración para frecuencia de emisión de las balizas Bluetooth se ha establecido en 300 milisegundos.

Cada vez que se finaliza un escaneo Wi-Fi se procede a almacenar en el servidor los valores obtenidos para el Wi-Fi y de igual modo para las balizas Bluetooth. Durante la fase de calibración se han realizado un total 1596 medidas.

Para la determinación de eficiencia se han obtenido un total de 10824 mediciones de redes Wi-Fi y Bluetooth asociando a cada una de ellas las coordenadas en el plano correspondiente.

Para la predicción de las posiciones se han usado diferentes algoritmos que permiten realizar la clasificación, en la tabla 1, se puede ver el error en metros obtenidos con el cada uno de los diferentes algoritmos.

CLASIFICADOR	ERROR
REDES BAYESIANAS	2.398849
RANDOMFOREST	2.549873
J48	3.727805
RANDOMTREE	3.972994

Tabla 1: Clasificadores redes WI-FI

Entre los algoritmos probados, las redes bayesianas, el j48, el random forest y el random tree, el que mejores resultados ofrece en la predicción de las posiciones es el de redes bayesianas que ofrece resultados con un error inferior a 2.4 metros.

Con el error que se ha logrado permite localizar el robot y a su vez el usuario en una zona determinada, como puede ser una sala o un pasillo.

6. Conclusiones y líneas de trabajo futuro

Con este trabajo se ha llevado a cabo una investigación y desarrollo de un vehículo de seguimiento autónomo de usuarios, y una localización de interiores aplicada a un caso de estudio real. El resultado es un prototipo en su fase final que permite al usuario interactuar con el robot sin la existencia de contacto físico.

Para el desarrollo de la propuesta, como se ha dicho con anterioridad, se han estudiado diferentes algoritmos de detección de usuarios y características en imágenes. Sin embargo a la hora de implementación se han hecho uso de librerías que los implementan internamente, logrando una capa de abstracción reduciendo la complejidad del trabajo, además estas librerías tienen como ventaja añadida estar optimizadas para el hardware empleado.

El uso de la tecnología infrarroja de la nube de puntos permite omitir los sensores de distancia frontales ya que de esta nube de puntos se infieren las distancias a los diferentes objetos y usuarios que se puedan encontrar en la trayectoria del robot. Estas medidas inferidas son mucho más precisas a las equivalentes que se puede obtener con otras tecnologías.

El haber hecho uso de las tecnologías Wi-Fi y Bluetooth ha permitido la creación de un sistema económico ya que hace uso de las tecnologías existentes ya instaladas en los edificios.

Debido a que el caso de estudio fue aplicado a un caso real, el carro de la compra del supermercado ha permitido ser expandido con la visualización de estadísticas sobre las ventas realizadas y la aplicación de realizar la compra.

El sistema desarrollado además de las funcionalidades aquí descritas permite muchos más usos e implementaciones, algunos de ellos se reflejarán dentro del apartado de líneas de trabajo futuras, otras que no tienen relación con el caso de estudio, como por ejemplo el transporte de maquinaria en hospitales, robots asistentes de carga, transporte de materiales peligrosos evitando el contacto directo de los usuarios con estos.

Líneas de trabajo futuras

- Añadir al robot un sensor magnético, ya que este valor tiene ligeras alteraciones entre posiciones cercanas, de este modo se añade otra característica para la estimación de las posiciones.
- Añadir al robot sensores de tipo encoder o acelerómetro para la detección de la velocidad en cada instante, de manera que puedan ser combinados con las técnicas desarrolladas de fingerprint con el WIFI y Bluetooth para un incremento de la precisión

- Generación de mapas de calor para que se permita realizar un estudio de manera más sencilla de las zonas más transitadas dentro del supermercado.
- En el caso de estudio incorporar un sistema de recomendación, que tenga en cuenta la localización del usuario y de esta manera pueda recomendar productos al usuario en el supermercado que se encuentren en la zona que esté el usuario.
- Incorporar al sistema de localización de interiores la funcionalidad de guiado. En el caso de estudio permitiría al usuario viajar dentro del supermercado optimizando los tiempos de compra de los usuarios.
- Exportación del algoritmo de reconocimiento de usuarios a dispositivos más económicos, como puede ser cualquier tipo de cámara que haga uso de librerías como por ejemplo opencv [28] .

Referencias

1. Akihisa Ohya, T.M.: Intelligent Escort Robot Moving together with Human - Interaction in Accompanying Behavior. FIRA Robot Congress (2002)
2. Sales, J., Marín, R., Cervera, E., Rodríguez, S., Pérez, J.: Multi-Sensor Person Following in Low-Visibility Scenarios. *Sensors* **10**(12) (dec 2010) 10953–10966
3. Feng, G., Guo, X., Wang, G.: Infrared motion sensing system for human-following robots. *Sensors and Actuators A: Physical* **185** (oct 2012) 1–7
4. Tatsuya Fujii, Jae Hoon Lee, Member, IAENG, S.O.: Gesture Recognition System for Human-Robot Interaction and Its Application to Robotic Service Task. *Proceedings of the International MultiConference of Engineers and Computer Scientist* **1** (2014)
5. Kato, H., Billingham, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, IEEE Comput. Soc 85–94
6. Jun Park, Bolan Jiang, Neumann, U.: Vision-based pose computation: robust and accurate augmented reality tracking. In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, IEEE Comput. Soc 3–12
7. Harris, C., Stephens, M.: A COMBINED CORNER AND EDGE DETECTOR
8. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, IEEE (2005) 1508–1515 Vol. 2
9. Flores, P., Braun, J.: Algoritmo SIFT: fundamento teórico. (2011)
10. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features
11. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features
12. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary Robust Invariant Scalable Keypoints
13. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF
14. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y., Mount, D.M., Wu, A.Y.: An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. *Journal of the ACM* **45**(6) (1994) 573–582

15. Tena, L.G., Sossa, H., Alvarado, A., Vergara, O.: Reconocimiento de objetos en una plataforma robótica móvil Introducción. *CULCyT* (55) (2015)
16. Hervás Ortega Director, M., González Morcillo, C.: Framework para el Desarrollo de Interfaces Naturales de Usuario basadas en Visión por Computador y Realidad Aumentada (2015)
17. Cárdenas Quiroga, E.A., Yolanda, L., Martín, M., Caycedo, A.U.: Stereoscopy, methods and applications in multiple fields of knowledge. *Stéréoscopie, des méthodes et des applications dans différents domaines de la connaissance Estereoscopia, métodos e aplicações em diferentes áreas do conhecimento*
18. Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, Hao Yan: Scanning 3D Full Human Bodies Using Kinects. *IEEE Transactions on Visualization and Computer Graphics* **18**(4) (apr 2012) 643–650
19. Seamon, B., DeFranco, M., Thigpen, M.: Use of the Xbox Kinect virtual gaming system to improve gait, postural control and cognitive awareness in an individual with Progressive Supranuclear Palsy. *Disability and Rehabilitation* **39**(7) (mar 2017) 721–726
20. : Open-source SDK for 3D sensors - OpenNI
21. : Kinect for Windows SDK 1.8 | Kinect for Developers
22. Xin Hu, Lianglun Cheng, Guangchi Zhang: A Zigbee-based localization algorithm for indoor environments. In: *Proceedings of 2011 International Conference on Computer Science and Network Technology*, IEEE (dec 2011) 1776–1781
23. Wang, T., Wang, T., Gao, H., Li, Y.: Research on Indoor Location Method Based on WLAN Signal Location Fingerprints. Springer, Cham (mar 2018) 828–840
24. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* **37**(6) (nov 2007) 1067–1080
25. Singh, V., Aggarwal, G., Ujwal, B.V.S.: Ensemble based real-time indoor localization using stray WiFi signal. In: *2018 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE (jan 2018) 1–5
26. Xie, B., Gong, S., Tan, G.: LiPro: light-based indoor positioning with rotating handheld devices. *Wireless Networks* **24**(1) (jan 2018) 49–59
27. Seco, F., Koutsou, K., Ramos, F., Jiménez, A.R.: Localización personal en entornos interiores con tecnología RFID. *Revista Iberoamericana de Automática e Informática Industrial RIAI* **10**(3) (jul 2013) 313–324
28. : OpenCV library

Sistema de Recomendación de Música Sensible al Contexto

F. Andrés Yepes Vélez, Vivian F. López Batista, María N. Moreno

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{andresyv,vivian,mmg}@usal.es
<http://diaweb.usal.es/>

Resumen Los sistemas de recomendación han adquirido gran importancia en los últimos años con el éxito de las plataformas de *streaming*. Ante un catálogo que se hace cada día más grande, es necesario implementar soluciones que ayuden a los usuarios a encontrar los ítems que realmente le interesen. Existen dos grandes tipos de técnicas para recomendaciones: filtrado basado en contenido y filtrado colaborativo siendo estos últimos los más usados. Recientemente han adquirido gran interés los sistemas de recomendación sensibles al contexto, los cuales tienen en cuenta el contexto en el que se encuentra el usuario para poder realizar una recomendación más precisa. En este artículo, aplicaremos los algoritmos más populares de filtrado colaborativo para recomendación de música y veremos como la incorporación de datos contextuales afectan a la precisión de los mismos. Además, se aplicarán técnicas para obtener ratings de forma implícita, lo cual ha sido necesario debido a la escasez de datasets con información contextual y ratings explícitos de los usuarios. Finalmente se concluye que la inclusión de la dimensión de contexto al proceso de recomendación supone una mejora en la fiabilidad de los sistemas evaluados.

1. Introducción

Los sistemas de recomendación se han convertido en una parte crucial de los servicios de *streaming* de música. Ante un catálogo que es cada vez más grande, las diferentes plataformas deben buscar soluciones que ayuden a los usuarios a navegar a través de él y a encontrar los ítems que realmente son de su interés. Se han desarrollado diferentes soluciones para generar recomendaciones como las técnicas de filtrado basado en contenido [1] que realizan recomendaciones en base a las características intrínsecas de los ítems y unos perfiles de usuario; las técnicas de filtrado colaborativo [2,3] que realizan recomendaciones basándose en los *ratings* que han dado los usuarios a los ítems y los sistemas de recomendación híbridos, que combinan diferentes enfoques con el objetivo de mejorar la precisión del sistema quedándose con lo mejor de uno y de otro. Los sistemas de recomendación de filtrado colaborativo son los más usados actualmente y los que mejores resultados arrojan.

Estos sistemas son bastante precisos pero fallan en cuanto al contexto del usuario. Por ejemplo, supongamos que una familia se dispone a ver una película en un servicio de *streaming* como Netflix¹ y el sistema recomienda películas clasificadas para mayores de 18 años. Suponiendo que en dicha familia hay menores de edad, ¿podemos considerar correcta la recomendación? Este problema se soluciona aplicando información de contexto al proceso de recomendación y es lo que llamamos un **Sistema de Recomendación sensible al Contexto**.

El objetivo principal de este proyecto es el de utilizar las técnicas de filtrado colaborativo más populares e incorporar información contextual del usuario al proceso de recomendación obteniendo así un sistema de recomendación sensible al contexto. La idea, es la de generar recomendaciones teniendo en cuenta el contexto en el que se encuentra el usuario ya que las preferencias de un usuario pueden variar dependiendo su estado de ánimo, día de la semana, época del año, su ubicación y demás. Una vez implementado el sistema, se someterá a evaluación y sus resultados serán comparados con los de otros enfoques para analizar cómo afecta la incorporación del contexto al rendimiento del sistema

2. Estado del arte

2.1. Sistemas de recomendación

Los sistemas de recomendación (SR) son herramientas que, mediante técnicas matemáticas o más recientemente mediante técnicas de machine learning (ML) [4], ofrecen/recomiendan contenido o productos (ítems) a los usuarios que pueda ser de su interés. Estos sistemas nacen con la necesidad de filtrar productos de un catálogo que se hace más grande cada día en diferentes tipos de servicios.

Existen muchos tipos de sistemas de recomendación y diferentes autores pueden diferir en la forma de clasificarlos. Para dar una visión general de los sistemas de recomendación vamos a basarnos en la taxonomía clásica usada en [5,2]:

- **Sistemas de recomendación basados en contenido:** generan recomendaciones basándose en dos fuentes de información: las características asociadas a los ítems y los *ratings* que los usuarios han dado a dichos ítems. Recomiendan ítems cuyas características son similares a las de otros ítems que han sido valorados positivamente en el pasado.
- **Sistemas de recomendación de filtrado colaborativo:** generan recomendaciones usando solo la información existente de los perfiles de los diferentes usuarios. En su forma más simple, recomiendan al usuario activo los ítems que otros usuarios con gustos similares han valorado de forma positiva en el pasado. La similitud de unos usuarios con otros es calculada en base a la similitud de los *ratings* dados por ambos usuarios a los mismos ítems en el pasado.
- **Sistemas de recomendación híbridos:** combinan diferentes tipos de técnicas como las mencionadas anteriormente con el objetivo de intentar cubrir

¹ <http://www.netflix.com/>

las desventajas de un tipo de sistema con las ventajas de otro. Por ejemplo, RS de filtrado colaborativo no pueden generar recomendaciones ante la presencia de ítems nuevos ya que no pueden recomendar ítems que no poseen ninguna valoración de ningún usuario anteriores. En un RS basado en contenido no tendríamos este problema puesto que solo se tendrían en cuenta las características del ítem para encontrar otros ítems que sean similares a él.

- **Sistemas de recomendación sensibles al contexto:** añaden una dimensión más al problema de la recomendación: el contexto. Esto es debido a que las preferencias de un usuario pueden variar dependiendo del contexto en el que se encuentren, ya sea el día, el tiempo, su ubicación, su estado de ánimo, etc.

La salida de los sistemas de recomendación suele ser una lista de ítems que son recomendados al usuario activo, en cuyo caso decimos que se trata de un *sistema de recomendación de ítems* (o recomendaciones top-N) o una predicción de *ratings* a un conjunto de ítems que han sido pasados como entrada, en cuyo caso decimos que es un *sistema de predicción de ratings*.

2.2. Filtrado colaborativo

Los sistemas de recomendación de filtrado colaborativo (Figura 1) se basan en la suposición de que, si dos usuarios han compartido los mismos intereses en el pasado, lo harán también en un futuro. De esta forma si un usuario A y un usuario B tienen gustos musicales similares y el usuario A recientemente ha escuchado canciones de Nirvana que no han sido escuchadas por el usuario B, la idea es recomendar estas canciones al usuario B.

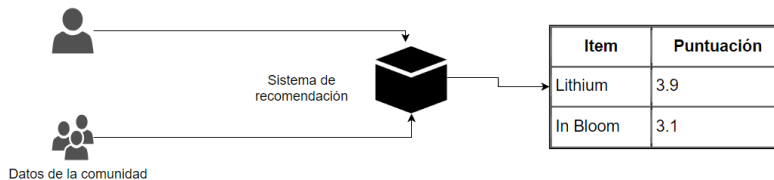


Figura 1: Filtrado colaborativo

Existen dos tipos de sistemas de recomendación de filtrado colaborativo [5]:

- **Filtrado colaborativo basado en usuarios:** en este tipo de sistemas las recomendaciones son generadas teniendo en cuenta las preferencias de los usuarios que componen el “vecindario” del usuario activo. La idea es obtener un grupo de usuarios similares al usuario activo en base a sus preferencias (un vecindario) y generar recomendaciones basándose en los ratings que han

sido dados por usuarios del vecindario en ítems que el usuario activo no ha valorado y se recomiendan.

- **Filtrado colaborativo basado en ítems:** similar al anterior, pero en este caso se tienen en cuenta los ítems en lugar de los usuarios. Se parte de los ítems que el usuario ha valorado positivamente en el pasado y se buscan ítems que sean similares a estos y que no hayan sido valorados por el usuario activo y se recomiendan.

Como mencionan en [5,2], también podemos clasificar estos sistemas en dos clases generales:

- **Enfoques basados en memoria:** también conocidos como enfoques de vecinos más cercanos, utilizan directamente los *ratings* almacenados en el sistema para predecir los *ratings* que dará el usuario a un ítem nuevo. Involucran cálculos de similitud en memoria para encontrar los “vecinos más cercanos” ya sea entre usuarios o ítems.
- **Enfoques basados en modelos:** usan los *ratings* para entrenar un modelo predictivo. Transforman los usuarios y los ítems a el mismo espacio de variables latentes e intenta encontrar una relación entre los *ratings* y las relaciones usuario-ítem, caracterizándolos con variables que son inferidas directamente del *feedback*.

Los enfoques basados en modelos han demostrado ser superiores a los enfoques basados en memoria ya que en los últimos, los datos deben ser cargados en memoria y deben realizarse los cálculos para realizar las recomendaciones en tiempo de recomendación. Sin embargo, los enfoques basados en modelos carecen de un principio que ha empezado a tomar gran importancia de cara a la satisfacción del usuario final: la *serendipia*.

La *serendipia* extiende el concepto de novedad ayudando al usuario a descubrir ítems que son de su agrado y que de otra forma no habría descubierto puesto que no se adapta a su perfil habitual de gustos. Los sistemas basados en modelos son bastante buenos a la hora de caracterizar las preferencias de un usuario, por ejemplo, un sistema basado en modelo sería capaz de inferir que un usuario en concreto tiene preferencia por música rock y recomendarle música de este mismo tipo que el usuario tal vez no haya escuchado, sin embargo, este sistema podría no ser capaz de recomendar canciones que no sean de rock pero que para el usuario pueden resultar de su agrado.

La ventaja de estos sistemas es que son fáciles de implementar y muy precisos. Sin embargo, tienen sus limitaciones:

- **Dispersión:** este problema se debe a que generalmente en los datos disponibles para generar el modelo de recomendación, se dispone de una gran cantidad de ítems de los cuales sólo una pequeña porción ha sido valorada.
- **Escalabilidad:** este tipo de sistemas presentan un problema de escalabilidad ya que a medida que crece el número de usuarios y el número de ítems, aumenta también el tiempo de computación.

- **Primera Valoración (early rater)**: problema que se produce cuando un ítem nuevo es introducido al sistema y no existe ninguna valoración sobre él.
- **Arranque en frío (Cold-start)**: problema ante la presencia de usuarios nuevos que, al no haber realizado ninguna valoración, no es posible generar recomendaciones.
- **Oveja negra (Gray Sheep)**: usuarios cuyas preferencias no coinciden con las de ningún grupo

2.3. Contexto

Aplicando información del contexto del usuario al sistema de recomendación evitamos, por ejemplo, recomendar a un usuario que está haciendo ejercicio, música lenta o con poco ritmo o usando el ejemplo anterior, el sistema recomendaría películas clasificadas como aptas para todo público.

La fiabilidad de las recomendaciones puede verse altamente afectada por la información contextual del usuario. Esta información puede ser *estática*, si es inmutable con el tiempo o *dinámica*, si cambia con el paso del tiempo. Un ejemplo de información contextual estática sería, por ejemplo, la fecha de nacimiento del usuario y un ejemplo de información contextual dinámica su localización, las condiciones meteorológicas y la actividad que esté realizando.

Tipos de contexto En [6], Villegas et al. clasifican el contexto en cinco clases generales: contexto individual, contexto de ubicación, contexto temporal, contexto de actividad y contexto relacional:

- **Contexto individual** se corresponde con la información observada de entidades independientes (usuarios o ítems) que pueden tener características en común. Este tipo de contexto se subdivide a su vez en contexto natural, contexto humano, contexto artificial y contexto de grupo de entidades. El contexto natural representa las características de entidades vivas y no vivas que ocurren de forma natural, es decir, sin intervención humana, por ejemplo, la información meteorológica. El contexto humano describe el comportamiento y preferencias del usuario, por ejemplo, el método de pago preferido por el usuario. El contexto artificial describe las entidades resultantes de acciones humanas o procesos técnicos, por ejemplo, configuraciones de hardware y software usadas en plataformas de comercio electrónico. Por último, la información contextual de grupos de entidades se refiere a grupos de sujetos independientes que comparten características comunes y que pueden relacionarse entre sí, por ejemplo, las preferencias de los usuarios en la red social del usuario activo.
- **Contexto de ubicación**: este tipo de contexto se refiere al lugar asociado a la actividad del usuario, por ejemplo, la ciudad en la que vive. Al igual que el anterior, este tipo de contexto puede subdividirse en dos tipos distintos: físico, por ejemplo, las coordenadas de la ubicación del usuario o la dirección del sitio en el que se encuentra y virtual, como por ejemplo la dirección IP del ordenador desde la que está conectado el usuario.

- **Contexto temporal:** se refiere a información como la parte del día (mañana, medio día, tarde, noche y media noche), la fecha, día de la semana y temporada del año. El contexto temporal se puede subdividir en definido, si representa un marco de tiempo con un punto específico de inicio y fin e indefinido si representa eventos recurrentes que suceden cuando otro evento sucede, por lo cual no tienen un tiempo definido como tal. Por ejemplo, la sesión de un usuario en un comercio electrónico.
- **Contexto de actividad:** se refiere a las tareas realizadas por los usuarios en el momento de la recomendación. Por ejemplo, hacer ejercicio es una tarea que el usuario realiza en un tiempo específico.
- **Contexto relacional:** se refiere a las relaciones entre entidades que se producen debido a las circunstancias que las rodean. El contexto relacional puede ser social, por ejemplo, asociaciones o afiliaciones y funcional, por ejemplo, el uso que una entidad hace de otra.

Obtención de información contextual La información contextual puede ser obtenida de diferentes formas [7]:

- **Explícitamente:** se pregunta directamente al usuario o se obtiene información mediante otros medios.
- **Implícitamente:** de los datos en entorno, como cambios en la localización del usuario aprovechando los sensores de un teléfono móvil. Otro ejemplo sería obtener información contextual temporal en un sistema transaccional observando el *timestamp* en que se realiza una transacción. A diferencia del explícito, aquí no interviene el usuario de forma directa. Los datos son inferidos por el sistema a través de los datos de interacción del usuario con el sistema.
- **Inferencia:** se puede inferir también el contexto aplicando técnicas de minería de datos. Por ejemplo, una compañía de streaming de series como Netflix no puede saber de forma directa qué tipo de persona o personas están usando el servicio. Sin embargo, sí que se puede inferir con una precisión razonable observando el tipo de series o películas que ha visto usando diferentes métodos de minería de datos. El problema de esto es que para que el sistema sea capaz de inferir, tiene que haber una gran cantidad de datos disponibles para crear un modelo predictivo² lo suficientemente preciso para inferir el contexto del usuario.

Integración del contexto en los SR Hasta ahora hemos considerado las recomendaciones como un problema de 2 dimensiones, es decir, preferencias de usuario y representaciones de ítems. Con la inclusión del contexto como una nueva dimensión (Figura 2), podemos construir recomendaciones basadas en el contexto como un problema de 3 dimensiones [7]:

En un sistema de recomendación tradicional, el proceso de recomendación comienza con la especificación inicial del conjunto de *ratings* que, como hemos

² Un clasificador, por ejemplo.

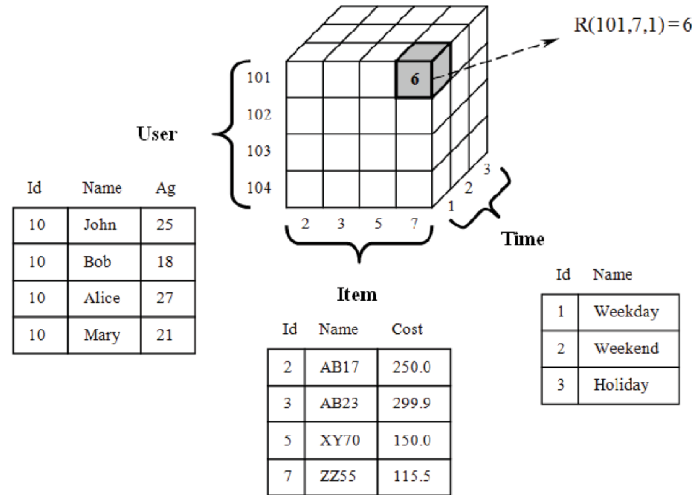


Figura 2: Recomendación sensible al contexto

mencionado antes, puede ser proporcionado explícitamente por los usuarios o inferido implícitamente por el sistema. Una vez estos *ratings* son especificados, el sistema de recomendación intenta estimar la función de *rating* R para los pares (usuario, ítem) que no han sido calificados aún por los usuarios.

$$R : User \times Item \rightarrow Rating \quad (1)$$

Este enfoque busca mejorar el modelado y predicción de los gustos del usuario incorporando información adicional sobre el contexto del usuario como categorías adicionales de datos explícitos. En este caso, la función de *rating* sería como muestra la ecuación 2:

$$R : User \times Item \times Context \rightarrow Rating \quad (2)$$

Para la integración, existen tres paradigmas para integrar la información contextual en el proceso de recomendación (Figura 3). Según la forma en que se realiza esta incorporación, podemos hablar de hasta 3 formas distintas de hacerlo [5,7,8]:

- **Pre-filtrado contextual:** en este enfoque la información contextual es utilizada para filtrar los datos antes de ser usados en la recomendación. De esta forma, primero se escoge un subconjunto de datos del conjunto de datos total el cual coincide con el contexto que queremos filtrar. Luego este subconjunto es pasado a un sistema de recomendación tradicional en donde se predicen los *ratings* usando las dos dimensiones tradicionales de los sistemas de recomendación $User \times Item$.
- **Post-filtrado contextual:** en este enfoque, los datos contextuales son ignorados en el principio de la recomendación. Se realiza la predicción de los

ratings usando cualquier sistema de recomendación tradicional de dos dimensiones sobre todos los datos. Una vez se obtienen los resultados, se obtiene un subconjunto de ítems del resultado en el cual la información de contexto coincide con el contexto que se está evaluando. Este subconjunto de datos es el producto final.

- **Modelado contextual:** en este enfoque el contexto es usado directamente como parte de la estimación de *ratings*.

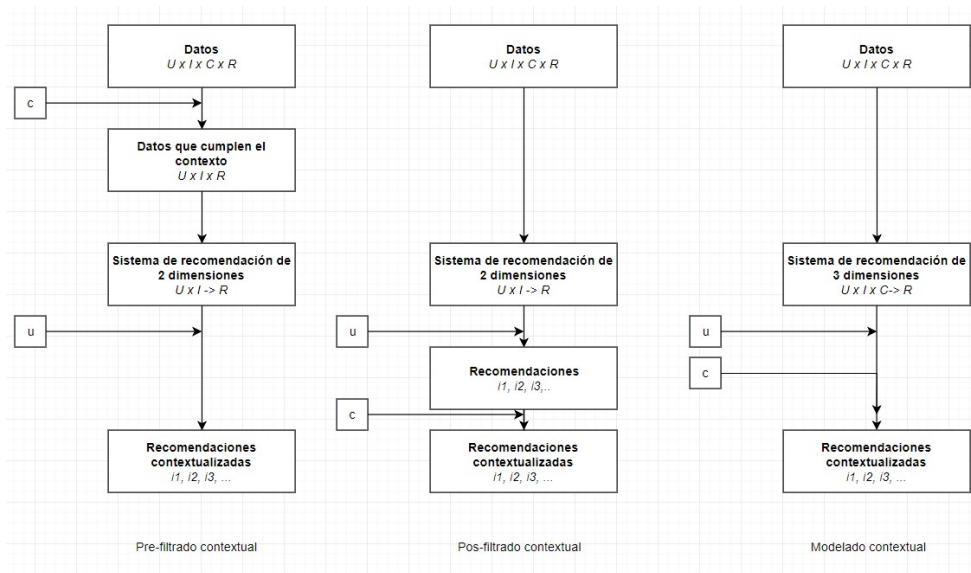


Figura 3: Filtrados contextuales

2.4. Trabajos relacionados

En esta sección se hablará brevemente sobre diferentes investigaciones recientes en las que se ha aplicado información contextual al proceso de recomendación.

En [9] Viktoratos et al. proponen un sistema híbrido en el que las recomendaciones son realizadas mediante reglas de asociación y los parámetros contextuales son introducidos en la precondición con el objetivo de aliviar el problema de arranque en frío.

Wang et al. [10] utilizan un modelo probabilístico que integra información contextual sobre las actividades diarias del usuario, obtenidas mediante un dispositivo móvil.

Lee et al. [11] presentan una arquitectura muy similar al anterior pero en este caso tienen en cuenta lo que llaman “micro actividades” que ocurren en un intervalo muy pequeño de tiempo.

Ostuni et al. [12] presentan un sistema de recomendación de películas que explota la información contextual espacio-temporal para mejorar un sistema de recomendación basado en contenido.

Otros contextos en los que se ha puesto especial interés es en el que el usuario está conduciendo, así, en [13] realizan un estudio sobre la influencia del tipo de música que prefiere escuchar un usuario en diferentes escenarios de conducción simulados estudiando también la carga cognitiva del conductor en cada uno de ellos y en [14], desarrollan un framework en Java en el que de forma interactiva, se va a ajustando el modelo de recomendación según las preferencias del conductor que concluye en confirmar que el tipo de música que se escucha mientras se conduce afecta a la carga cognitiva de manera distinta en casos donde la música es rápida y en otros casos donde la música es lenta.

Otro enfoque bastante interesante es el presentado por Pichl et al. [15] en el cual la información contextual es extraída a partir del nombre que dan los usuarios a listas de reproducción creadas por ellos, obteniendo un rendimiento un 33 % superior al de los enfoques tradicionales.

En [16] Dao et al. crean un sistema para recomendación basada en la localización, en el cual extraen la información del contexto de los usuarios a partir de sus dispositivos móviles y hallan similitudes entre los contextos (filtrado colaborativo) para realizar las recomendaciones pertinentes obteniendo finalmente una mejora clara en la efectividad de otros enfoques que usan como base.

Lifetrak [17] es un sistema que genera una lista de reproducción a partir de la biblioteca de música del usuario teniendo en cuenta factores relativos al contexto físico de éste como localización, hora, día, tráfico, nivel de ruido, temperatura y tiempo atmosférico. El sistema sigue un enfoque de post-filtrado contextual en el que se generan recomendaciones a partir de una biblioteca etiquetada previamente por el usuario con etiquetas que representan situaciones contextuales en las que son apropiadas.

El humor o los sentimientos de los usuarios también son factores contextuales muy utilizados en la recomendación de música. Además, el contexto emocional se puede utilizar para establecer un puente entre ítems de música e ítems de otros dominios permitiendo, de esta forma, hacer recomendaciones de medios cruzados [18]. Como ejemplo, podemos hablar del trabajo de Cai et al. [19] en el que se propone un sistema que recomienda música mientras el usuario lee páginas web. Las canciones, descritas por su letra y revisiones on-line, y los documentos web se representan como texto. De esta forma, se clasifican en una escala emocional basándose en las palabras que contienen y posteriormente establecen su similitud.

Hay ocasiones en las que la información contextual no se puede recoger como un conjunto de factores estáticos debido a que es información dinámica que puede inferirse a partir de la interacción de los usuarios con el sistema. Esto es usual en el dominio de la recomendación de música en el que el contexto puede derivarse de la secuencia de canciones que escucha el usuario. En [20] se presenta un sistema de recomendación de música que captura esos cambios contextuales del usuario y adapta dinámicamente las preferencias. Para ello, se aplica una técnica de obtención de patrones secuenciales, cada uno de los cuales representa un contexto diferente.

En [21] se utilizan modelos ocultos de Markov para capturar los cambios en las preferencias de los usuarios. La secuencia de ítems obtenida del usuario se utiliza para obtener el modelo, cuyas variables ocultas representan el contexto del usuario. La predicción del contexto se utiliza para generar las recomendaciones. Con esta propuesta, que se valida con datos de listas de reproducción de música, se consiguen mejorar los resultados de exactitud y diversidad proporcionados por otros métodos examinados.

Recientemente, muchos sistemas de recomendación híbridos explotan las redes sociales y otras fuentes web con el fin de reunir información que puede ser útil en el proceso de recomendación [22,23]. En el trabajo descrito anteriormente [20] se utiliza el etiquetado social para derivar los temas latentes asociados a las canciones a partir de las etiquetas más frecuentes asociadas a ellas. En [24] se propone un sistema de recomendación que también utiliza etiquetas de medios sociales para establecer la similitud entre las canciones. Además, se aprovecha la información de las etiquetas para capturar las preferencias del usuario. Los resultados revelan que el etiquetado social es un medio válido para predecir preferencias musicales.

Además de las etiquetas, en los sistemas de recomendación se pueden utilizar otras características de las redes sociales. En [25] se hace uso de las relaciones sociales implícitas, para ello proponen un modelo, *Explicit and Implicit Social Relation* (EISR), y un algoritmo, *Possibility of Friendship Between Non-Friends* (PFNF), para extraer la relación implícita (relación social escondida) en los grafos no dirigidos de redes sociales mediante la explotación de las técnicas de predicción de enlace. Tanto las amistades explícitas como implícitas se incorporan en el modelo propuesto. La evaluación de este modelo con un conjunto de datos de *last.fm*³ obtiene buenos resultados.

Por otra parte, en [26] se examina el papel de las emociones en los sistemas de recomendación sensibles al contexto, incluyendo la definición y obtención de características emocionales, así como la incorporación de dichas características en los algoritmos de recomendación. La recomendación de música es un campo muy propicio para explotar las emociones [27]. En [28] se utiliza el estado de ánimo como contexto para hacer recomendaciones de música. En [23] se proponen métodos sensibles a la emoción con el mismo propósito. Se utilizan microblogs para extraer las emociones del usuario con diferentes grados de granularidad y durante diferentes ventanas de tiempo.

En el contexto del etiquetado social y el análisis del sentimiento, los métodos de clasificación multietiqueta han resultado ser de gran utilidad. Estos algoritmos se han aplicado en la recomendación de etiquetas en algunos sistemas como Bibsonomy [29]. En él se ayuda a los usuarios de la red social en el etiquetado de los recursos que manejan, los cuales pueden tener múltiples etiquetas. Para realizar el aprendizaje automático de las mismas y asignarles su grado de relevancia, utiliza un conjunto de recursos previamente etiquetado por los usuarios de la red. Por otra parte, los métodos de clasificación multietiqueta *Multi-Label k-Nearest Neighbors* (MLkNN) y *Binary Relevance k-Nearest Neigh-*

³ <https://www.last.fm>

bors (BRkNN), se han usado en aplicaciones de clasificación multietiqueta de emociones en música [30]. Otros clasificadores multietiqueta como Calibrated Label Ranking (CLR) y *Random k-Labelsets* (RAkEL) han demostrado ser adecuados para la clasificación automatizada de grandes colecciones de música con múltiples emociones, aplicándose en sistemas de recomendación de música [31] basados en las emociones del usuario.

Utilizando pre-filtrado contextual, en [32] crean microperfiles para filtrado colaborativo sensible al tiempo. Se probaron varias divisiones de tiempo para obtener los microperfiles que mejor representan al usuario en cada periodo particular. Este método fue validado con el dataset de *last.fm* para recomendación de canciones. En [33] realizan un estudio comparativo de los enfoques de pre-filtrado y post-filtrado colaborativo considerando la época del año como dimensión de tiempo.

Los trabajos que proponen métodos que aprovechan el tiempo conjuntamente con otras variables contextuales son más numerosos. En [34] proponen un enfoque multidimensional para trabajar conjuntamente con información de los usuarios, ítems y del contexto. Los *ratings* son estimados usando un enfoque basado en reducción que aplica filtrado colaborativo en segmentos contextuales. Para validar el método, utilizan un sistema de recomendación de películas en el cual el tiempo y la empresa son utilizados como información contextual.

3. Metodología

El objetivo principal de esta investigación es el de observar como afecta a la fiabilidad de un sistema de recomendación de música la introducción de parámetros contextuales en el proceso de recomendación. El gran problema con el que nos encontramos inicialmente fue la cantidad escasa de *datasets* disponibles para la implementación. Se ha optado por utilizar el ya bien conocido *dataset* de *last.fm* el cual posee información sobre usuarios y su historial de reproducción, sin embargo, no posee información de *ratings* ni de reproducciones de forma explícita, problema que ha sido abordado aplicando algunas técnicas que se explicarán con más detalle en este apartado. Dicho dataset puede ser utilizado para recomendaciones de artistas o canciones, siendo este último el enfoque seguido en este proyecto. Se han implementado 3 sistemas de recomendación distintos:

- Filtrado colaborativo básico, en el que no se tiene en cuenta información del contexto
- Filtrado colaborativo personalizado con información sobre la temporada del año
- Filtrado colaborativo personalizado con información sobre la hora del día (mañana, medio día, tarde, noche y media noche).

3.1. Arquitecturas utilizadas

Se han utilizado dos arquitecturas distintas, una para el enfoque básico y otra para el enfoque con información contextual. Para el enfoque básico (Figura 4)

contamos con un módulo de generación de *ratings* implícitos, que a partir de los *logs* de los usuarios extraídos de *last.fm*, genera *ratings* que pueden ser usados para el proceso de recomendación y por otro lado, un módulo de recomendación que es un RS básico de dos dimensiones.

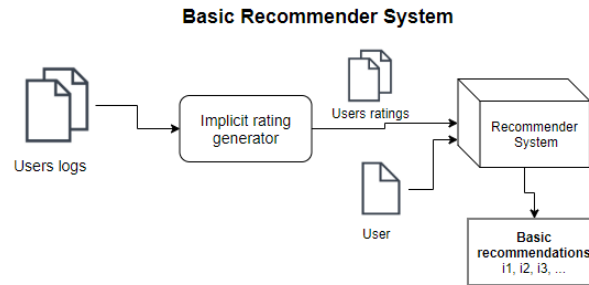


Figura 4: Arquitectura RS básico

Para el enfoque sensible al contexto, se ha seguido una arquitectura de pre-filtrado contextual y es similar al anterior pero incorpora un par de módulos extra (Figura 5): un módulo extractor de contexto, que extrae la información de contexto temporal usando como referencia el *timestamp* de los *logs* y un módulo de filtrado contextual, que se encarga de filtrar los datos de forma que sólo conserva los que coinciden con el contexto actual del usuario.

3.2. Preprocesamiento de datos

El *dataset* utilizado es uno creado por Oscar Celma⁴, el cual contiene registros del tipo $\langle user, timestamp, artist, song \rangle$ obtenidos directamente de la API de *last.fm* que se corresponde con el historial de reproducción de cerca de mil usuarios de la plataforma.

Ante la ausencia de *ratings* explícitos, ha sido necesario crear el sistema a partir de *ratings* implícitos para lo cual, es necesario aplicar primero ciertas transformaciones al *dataset*:

Obtener parámetros contextuales dado que información contextual temporal que utilizamos en los enfoques que utilizamos la temporada el año y la parte del día no poseen unos límites claros, se ha optado por seguir la metodología mencionada en [35] en el que se aplica lógica borrosa y cada concepto contextual se considera como un conjunto borroso. Así, el contexto de cada *log* puede ser representado como un conjunto de conjuntos borrosos en el que cada subconjunto se corresponde con un concepto contextual.

⁴ <http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html>

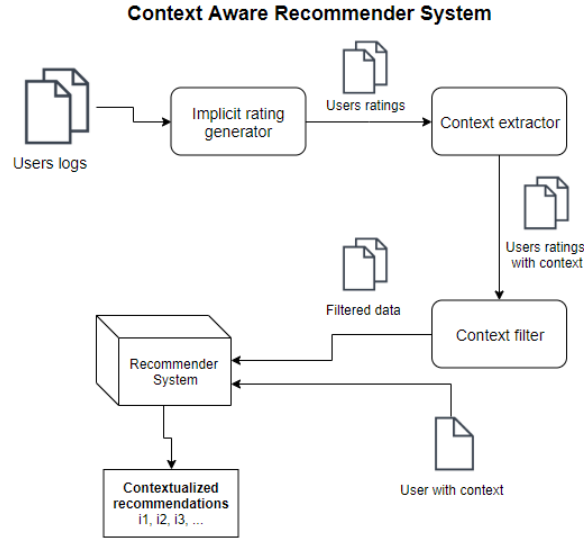


Figura 5: Arquitectura RS contextual

Asumiendo que existe un universo de conceptos contextuales:

$$C = \{C_1, C_2, C_3, \dots, C_k, \dots, C_n\}$$

un contexto puede ser descrito por un conjunto de pares compuestos por un concepto contextual C_k y un grado de pertenencia m_k que indica como de fuerte el contexto es descrito por C_k .

$$l.ctx = \{(C_k, m_k) | C_k \in C \wedge m_k = f_{c_k}(l)\} \quad (3)$$

donde f_{c_k} es una función de pertenencia que mide la pertenencia de un *log* a un concepto C_k y devuelve un valor entre 0 y 1.

De esta forma, Para el enfoque de la temporada del año usamos el número del día respecto a los días totales del año y se establecen los conceptos contextuales *spring*, *summer*, *autumn* y *winter* representados con los números 1, 2, 3 y 4 respectivamente. Con una función de pertenencia trapezoidal (Figura 6), medimos la pertenencia de cada log al contexto temporal y añadimos una columna, *season*, con el valor que representa al concepto contextual con mayor grado de pertenencia (Tabla 1).

Obtener número de reproducciones En el enfoque base es bastante sencillo, basta con agrupar por los campos *user-id* y *track-id* y añadir una columna adicional, *plays*, que es el sumatorio de ocurrencias de este par de atributos (Tabla 2).

En cuanto a los otros dos enfoques, el agrupamiento de los datos no sólo depende de los campos anteriores, sino también del contexto obtenido en el paso

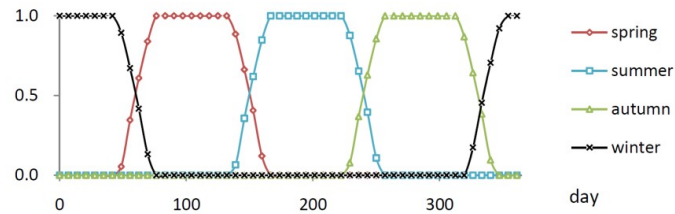


Figura 6: *Lógica borrosa: Season*

userid	trackid	season
user_000001	dc394163...	1
user_000001	340d9a0b...	1
user_000001	0b04407b...	2
user_000001	4e78efc4...	4
...

Tabla 1: Dataset con contexto

userid	trackid	plays
user_000001	dc394163...	35896
user_000001	340d9a0b...	22545
user_000001	0b04407b...	19563
user_000001	4e78efc4...	15236
...

Tabla 2: Dataset con *plays*

anterior. Para obtener el número de reproducciones se agrupa el *dataset* por *user-id*, *track-id* y *season* y se crea una columna adicional, *plays*, sumatorio del número de ocurrencias de estos 3 atributos (Tabla 3)

userid	trackid	season	plays
user_000001	dc394163...	1	25896
user_000001	340d9a0b...	1	22547
user_000001	0b04407b...	2	18563
user_000001	4e78efc4...	4	15247
...

Tabla 3: Dataset con contexto y *plays*

Para el enfoque de la hora del día, el procedimiento es similar al anterior sólo que en vez de usar el número del día usamos la hora del día y establecemos los conceptos contextuales *midnight*, *noon*, *morning*, *afternoon* y *evening* representados por los valores 0, 1, 2, 3 y 4 respectivamente. Aplicamos de nuevo una función de pertenencia trapezoidal al *log* y añadimos una columna, *timeday*, con el valor que representa al concepto contextual con mayor grado de pertenencia. Finalmente se agrupa el *dataset* por *user-id*, *track-id* y *timeday* y se crea una columna adicional, *plays*, sumatorio del número de ocurrencias de estos 3 atributos.

3.3. Cálculo implícito de *ratings*

Una vez calculado el número de reproducciones totales y los específicos de cada contexto, procedemos a calcular los *ratings* implícitos de cada usuario a cada canción. Para el enfoque base se calcula el *rating* implícito de cada par *userid-trackid* con el método descrito en [36]. Para los otros dos enfoques se ha modificado la ecuación para tener en cuenta la información del contexto, de tal forma que para un usuario u , una canción i y un contexto c , obtenemos la frecuencia de reproducción con la ecuación 4:

$$freq_{u,i,c} = \frac{count(u, i, c)}{\sum_{i'} count(u, i', c)} \quad (4)$$

y finalmente siendo $freq_{k,c}(u)$ la frecuencia de las k canciones más escuchadas por el usuario u , en el contexto c , se obtiene el *rating* r_{uic} con la ecuación 5:

$$r_{uic} = 4 \cdot \left(1 - \sum_{k'=1}^{k-1} freq_{k',c}(u) \right) \quad (5)$$

3.4. Modelos de recomendación

Se han implementado 3 modelos de recomendación, cada uno para los enfoques ya mencionados. Los 3 modelos han sido implementados mediante la librería

*Surpriselib*⁵ para Python. Para el enfoque base cogemos el dataset correspondiente y nos quedamos sólo con las columnas *user-id*, *track-id* y *score* (el *rating* implícito) como se muestra en la Tabla 4.

userid	trackid	score
user_000001	dc394163...	4
user_000001	340d9a0b...	3,9
user_000001	0b04407b...	3,5
user_000001	4e78efc4...	2,9
...

Tabla 4: Dataset recomendación base

Antes de ajustar el modelo con el *dataset*, debemos adaptarlo a la entrada del módulo de recomendación. Siendo N el número de usuarios y M el número de canciones del *dataset*, creamos una matriz de dimensión $N \times M$ en la que cada celda se corresponde con un *rating* dado por un usuario a una canción (Tabla 5). Esta matriz es pasada como parámetro al modelo creado con *Surpriselib* para la evaluación del sistema.

	track1	track2	track3
user1	4	3,9	3,5
user2	1	2,1	1,2
user3	3	2,5	1
user4	4	3,9	1,2
...

Tabla 5: Entrada Módulo de recomendación

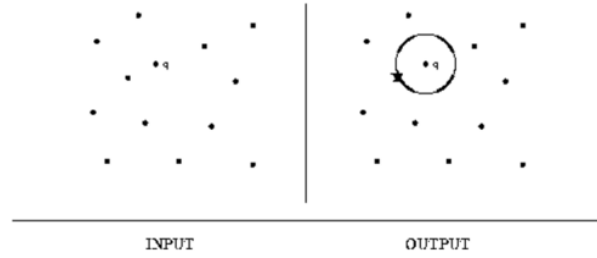
En cuanto a los otros dos enfoques, debemos tener en cuenta también el contexto en el que se encuentra el usuario, de forma que obtenemos los *ratings* de cada usuario, para cada ítem en cada uno de los contextos establecidos como se muestra en la Tabla 6.

Algoritmos de recomendación Como algoritmo para la recomendación se ha utilizado *k-Nearest Neighbors* (k-NN), una variante del Nearest Neighbor search en la que al igual que en el segundo, se trata de un problema de optimización en el que buscamos un punto que es más cercano (similar) a un punto dado (Figura 7), solo que en este caso encontramos los k puntos más cercanos. Dado un conjunto de puntos S en un espacio M y un punto de consulta $q \in M$, se trata de encontrar los k puntos más cercanos a q .

⁵ <http://surpriselib.com/>

		track1	track2	track3
user1	c1	4	3,9	3,5
	c2	1	3	2,2
	c3	4	3,1	2,8
	c4	3,7	1,9	3,1
user2	c1	4	3,9	3,5
	c2	3	2	2,3
	c3	2	0	3,8
	c4	1	1,6	0
...

Tabla 6: Entrada Contexto Módulo de recomendación

Figura 7: *Nearest Neighbor Search*

Se ha experimentado con diferentes configuraciones del algoritmo, utilizando diferentes medidas de similitud de las cuales la que mejor resultados nos ha dado ha sido *Pearson baseline*, una variante del *coeficiente de correlación de Pearson* que viene definida en [37] con la ecuación 6.

$$Pearson_bsl = \hat{\rho}(i, j) = \frac{\sum_{u \in U_{i,j}} (r_{ui} - b_{ui})(r_{uj} - b_{uj})}{\sqrt{\sum_{u \in U_{i,j}} (r_{ui} - b_{ui})^2 \sum_{u \in U_{i,j}} (r_{uj} - b_{uj})^2}} \quad (6)$$

donde r_{ui} es el *rating* dado por el usuario u al ítem i y b_{ui} es un factor residual de un predictor de línea base [37], que compensa las desviaciones específicas del usuario y el ítem. En cuanto a los *ratings*, se ha aplicado la técnica de normalización *z-score* que ha ayudado a aumentar más aún la fiabilidad del sistema. Para un enfoque basado en usuarios, definimos una medida $Z(r_{uj})$ con la ecuación 7:

$$Z(r_{uj}) = \frac{r_{uj} - \bar{r}_u}{\sigma_u} \quad (7)$$

donde \bar{r}_u es la media de los *ratings* dados por el usuario u y σ la desviación estándar. La predicción de *ratings* vendría definida por la ecuación 8:

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \frac{\sum_{v \in N_j(u)} \frac{w_{uv}(r_{uj} - \bar{r}_v)}{\sigma_v}}{\sum_{v \in N_j(u)} |w_{uv}|} \quad (8)$$

donde $N_j(u)$ es el conjunto de ítems que ha valorado el usuario u y w_{uv} es el peso de la similitud entre u y su vecino v .

También se ha implementado una variante utilizando el método de factorización de matrices SVD [37] con el objetivo de comparar un enfoque basado en memoria con uno basado en modelos.

4. Validación y resultados

Para la validación del sistema se ha aplicado *Cross-validation*, una técnica que divide el conjunto de datos en n particiones distintas (*folds*) y cada una de ellas es usada como conjunto de entrenamiento y de prueba al menos una vez durante la ejecución del algoritmo. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre las diferentes particiones [38]. Como medida de evaluación se ha utilizado *Mean Absolute Error* (MAE) y *Root Mean Square Error* (RMSE) siendo más relevante el RMSE puesto que es el que más penaliza los errores grandes y en este tipo de sistemas es lo más apropiado.

Siendo R_{train} y R_{test} el conjunto de *ratings* de entrenamiento y prueba respectivamente utilizados para aprender la función f , MAE viene definido con la ecuación 9:

$$MAE(f) = \frac{1}{|R_{test}|} \sum_{r_{ui} \in R_{test}} |f(u, i) - r_{ui}| \quad (9)$$

y el RMSE con la ecuación 10:

$$RMSE(f) = \sqrt{\frac{1}{|R_{test}|} \sum_{r_{ui} \in R_{test}} (f(u, i) - r_{ui})^2} \quad (10)$$

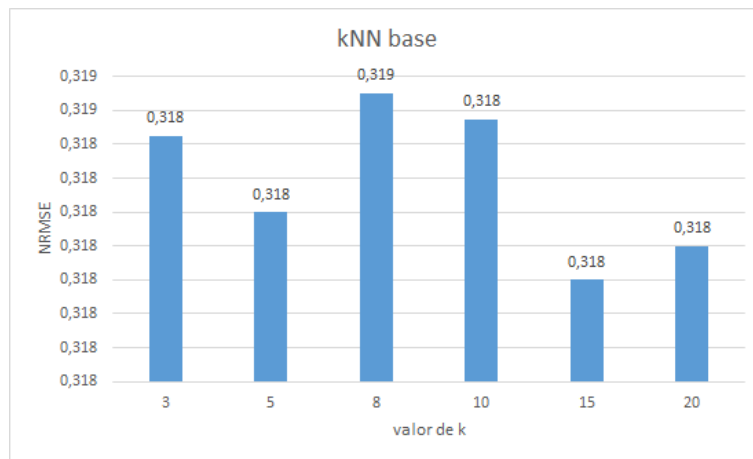
Existen otras derivadas de estas como el *Normalized RMSE* (NRMSE) y el *Normalized MAE* (NMAE) que son versiones en las que la medida es normalizada por el rango de los *ratings* y que utilizaremos para representar los resultados en gráficas.

4.1. Enfoque basado en memoria: k-NN

Para este enfoque se han realizado pruebas con diferentes valores de k : 3, 5, 8, 10, 15 y 20. Como medida de similitud se ha utilizado la ya mencionada *Pearson baseline*. Los resultados del *Cross-validation* se pueden observar en la tabla 7 y en la Figura 8.

k	RMSE	MAE
3	1,274	1,089
5	1,273	1,089
8	1,274	1,090
10	1,274	1,089
15	1,272	1,088
20	1,272	1,088

Tabla 7: Cross-validation z-score, Pearson baseline

Figura8: *Cross-validation enfoque base*

Como una primera observación vemos que el valor más óptimo de k ha sido 15 ya que ha sido el valor con menor error medio. Ahora vamos a aplicar contexto a este enfoque y compararemos los resultados del *Cross-validation* para analizar el efecto que produce la introducción de parámetros contextuales al sistema. La Figura 9 muestra los resultados de la validación del sistema base con $k = 15$ y los del sistema sensible al contexto utilizando parámetros contextuales del día. También, en la Figura 10 se compara el mismo enfoque base con el enfoque sensible al contexto utilizando esta vez la estación del año.

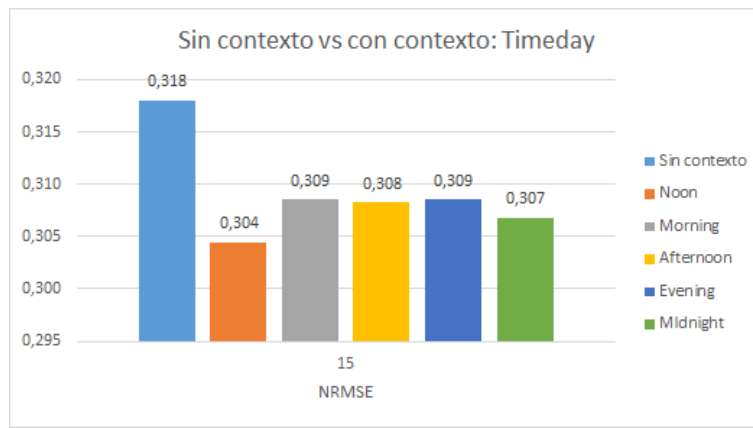


Figura 9: *Cross-validation contexto: timeday*

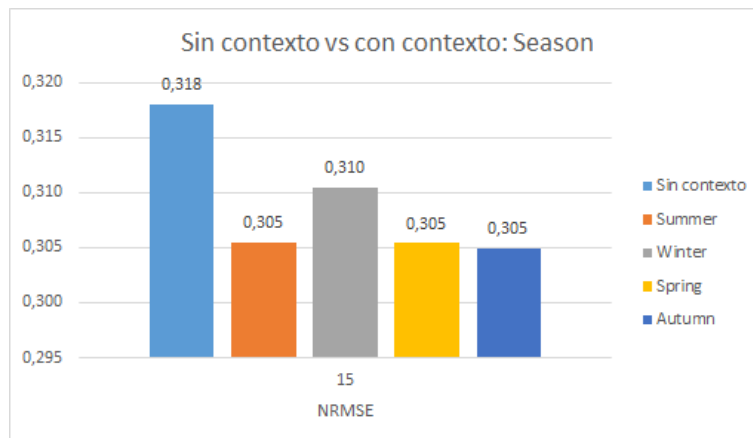


Figura 10: *Cross-validation contexto: season*

Vemos como la introducción del contexto ha supuesto una ligera mejora en la precisión del sistema teniendo en cuenta el contexto en el que se encuentra el usuario. Otro factor interesante a observar es el hecho de que hay contextos en los que el error es menor que en otros y esto se debe, en parte, a la reducción que hacemos cuando se aplica el filtrado contextual (lo cual hace que para algunos contextos haya mayor información disponible que en otros) y al comportamiento del propio algoritmo que usamos para la recomendación.

4.2. Enfoque basado en memoria: SVD

En cuanto a SVD, en la figura 11 vemos como la precisión del sistema es mayor que en el enfoque basado en memoria en el enfoque base. En cuanto a la introducción de parámetros contextuales, el comportamiento es similar, la introducción de contexto ha supuesto una ligera mejora en la fiabilidad del sistema (Figura 12, Figura 13) y volvemos a observar el mismo comportamiento de algunos contextos en los que el error parece ser menor.

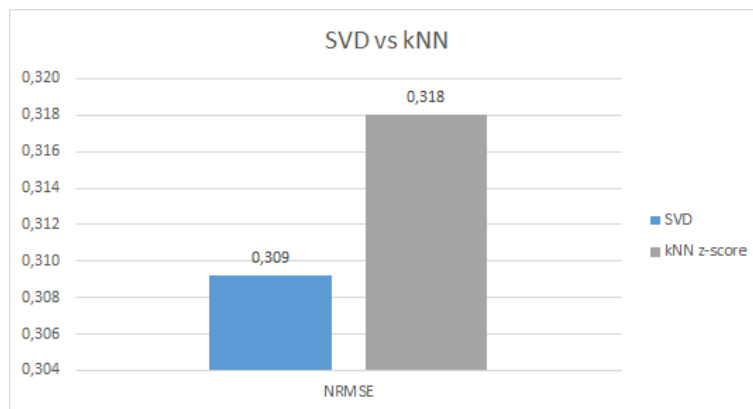
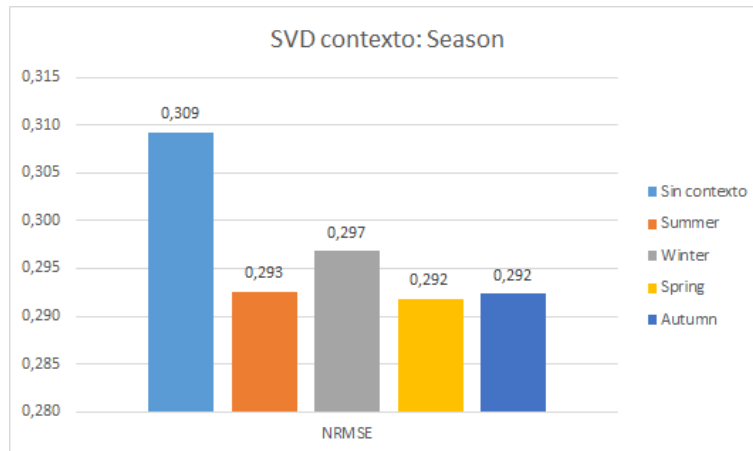
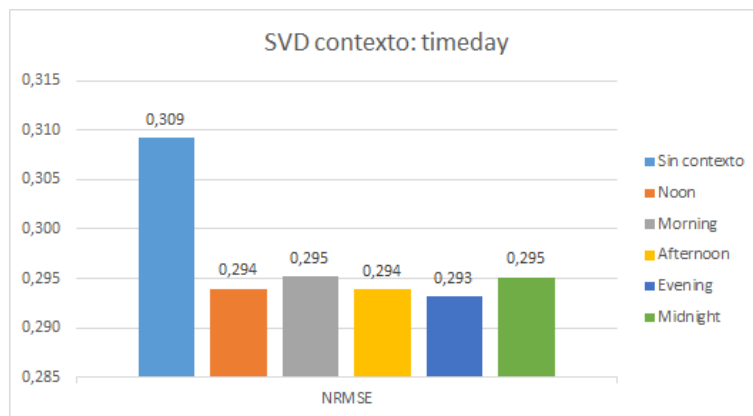


Figura 11: *SVD vs kNN*

Como hemos podido observar en los resultados obtenidos, las diferencias de error no son muy grandes, pero la tendencia es claramente a mejorar cuando se introducen parámetros de contexto al proceso de recomendación.

Figura 12: *SVD contexto: Season*Figura 13: *SVD contexto: Timeday*

5. Conclusiones

En este proyecto de investigación, se han llevado a cabo un estudio experimental con los sistemas de recomendación de filtrado colaborativo más populares, utilizando diferentes algoritmos (tanto basados en memoria como en modelos) y ajustando factores como la medida de similitud utilizada y la normalización de *ratings*. De esto podemos concluir que aplicar normalización a los *ratings* ha supuesto una mejora en la precisión, siendo la normalización *z-score* la que mejores resultados ha arrojado. También, en la comparativa de enfoques basados en memoria con basados en modelos, hemos visto como los enfoques basados en modelo tienen una mayor precisión que los basados en memoria, aunque se pierde el factor de la serendipia, que hace que el sistema sea capaz de recomendar ítems al usuario que este normalmente no escucharía pero que es posible que sean de su agrado.

Además, se han introducido factores contextuales para evaluar el rendimiento del sistema cuando se tienen en cuenta, lo cual era el principal objetivo de la investigación. De esto podemos concluir que la introducción de dimensiones contextuales ha supuesto una mejora en la fiabilidad de los sistemas evaluados que, aunque no es muy grande, presentan una clara tendencia a aumentar. Se han utilizado sólo variables contextuales temporales debido a la escasez de conjuntos de datos para realizar estos experimentos, pero lo ideal sería introducir gran variedad de contextos y evaluar el rendimiento con cada uno de ellos.

5.1. Líneas de trabajo futuras

Como líneas de trabajo futuras, podemos implementar un sistema de recomendación comercial con el objetivo de obtener más datos con los cuales trabajar y así introducir muchos más contextos para realizar pruebas. Algo bastante interesante para esta tarea sería el de implementarlo como aplicación para dispositivos móviles, la cual nos permita extraer información contextual de los diferentes sensores del dispositivo. Además, cada vez es más normal ver a personas con *wearables* como pulseras de actividad, que registran información contextual de la persona que podría ser usada para la recomendación. Aquí habría que ver alguna forma de integrar la aplicación con un sistema de streaming y por supuesto, que los usuarios estén dispuestos a compartir su información, lo cual es un tema que cada vez preocupa más a la gente.

Referencias

1. Lops, P., de Gemmis, M., Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends. In: Recommender Systems Handbook. Springer US, Boston, MA (2011) 73–105
2. Desrosiers, C., Karypis, G.: A Comprehensive Survey of Neighborhood-based Recommendation Methods. In: Recommender Systems Handbook. Springer US, Boston, MA (2011) 107–144

3. Sarwar, B.M., Konstan, J.A., Borchers, A., Herlocker, J., Miller, B., Riedl, J.: Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System
4. Portugal, I., Alencar, P., Cowan, D.: The use of machine learning algorithms in recommender systems: A systematic review. Volume 97. Pergamon (may 2018)
5. Gorakala, S.K.: Building recommendation engines : understand your data and user preferences to make intelligent, accurate, and profitable decisions
6. Villegas, N.M., Müller, H.A.: Managing Dynamic Context to Optimize Smart Interactions and Services. Springer, Berlin, Heidelberg (2010) 289–318
7. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-Aware Recommender Systems. *AI Magazine* **32**(3) (oct 2011) 67
8. Zheng, N., Li, Q.: A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications* **38**(4) (apr 2011) 4575–4587
9. Viktoratos, I., Tsadiras, A., Bassiliades, N.: Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems. *Expert Systems with Applications* **101** (jul 2018) 78–90
10. Wang, X., Rosenblum, D., Wang, Y.: Context-Aware Mobile Music Recommendation for Daily Activities. (2012)
11. Lee, W.P., Chen, C.T., Huang, J.Y., Liang, J.Y.: A smartphone-based activity-aware system for music streaming recommendation. *Knowledge-Based Systems* **131** (sep 2017) 70–82
12. Ostuni, V.C., Di Noia, T., Mirizzi, R., Romito, D., Sciascio, E.D.: Cinemappy: a Context-aware Mobile App for Movie Recommendations boosted by DBpedia
13. Helmholtz, P., Vetter, S., Robra-Bissantz, S.: AmbiTune: Bringing Context-Awareness to Music Playlists while Driving. Springer, Cham (2014) 393–397
14. Zheng, Y., Mobasher, B., Burke, R.: CARSKit: A Java-Based Context-Aware Recommendation Engine. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), IEEE (nov 2015) 1668–1671
15. Pichl, M., Zangerle, E., Specht, G.: Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name?
16. Dao, T.H., Jeong, S.R., Ahn, H.: A novel recommendation model of location-based advertising: Context-Aware Collaborative Filtering using GA approach. *Expert Systems with Applications* **39**(3) (feb 2012) 3731–3739
17. Reddy, S., Mascia, J.: Lifetrak. In: Proceedings of the 1st ACM international workshop on Human-centered multimedia - HCM '06, New York, New York, USA, ACM Press (2006) 25
18. Berkovsky, S., Kuflik, T., Ricci, F.: Cross-Domain Mediation in Collaborative Filtering. In: User Modeling 2007. Springer Berlin Heidelberg, Berlin, Heidelberg (2007) 355–359
19. Cai, R., Zhang, C., Wang, C., Zhang, L., Ma, W.Y.: MusicSense. In: Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07, New York, New York, USA, ACM Press (2007) 553
20. Hariri, N., Mobasher, B., Burke, R.: Context-aware music recommendation based on latent topic sequential patterns. In: Proceedings of the sixth ACM conference on Recommender systems - RecSys '12, New York, New York, USA, ACM Press (2012) 131
21. Hosseinzadeh Aghdam, M., Hariri, N., Mobasher, B., Burke, R.: Adapting Recommendations to Contextual Changes Using Hierarchical Hidden Markov Models. In: Proceedings of the 9th ACM Conference on Recommender Systems - RecSys '15, New York, New York, USA, ACM Press (2015) 241–244

22. Hyung, Z., Lee, K., Lee, K.: Music recommendation using text analysis on song requests to radio stations. *Expert Systems with Applications* **41**(5) (apr 2014) 2608–2618
23. Deng, S., Wang, D., Li, X., Xu, G.: Exploring user emotion in microblogs for music recommendation. *Expert Systems with Applications* **42**(23) (dec 2015) 9284–9293
24. Su, J.H., Chang, W.Y., Tseng, V.S.: Personalized Music Recommendation by Mining Social Media Tags. *Procedia Computer Science* **22** (jan 2013) 303–312
25. Reafee, W., Salim, N., Khan, A.: The Power of Implicit Social Relation in Rating Prediction of Social Recommender Systems. *PLOS ONE* **11**(5) (may 2016) e0154848
26. Zheng, Y., Mobasher, B., Burke, R.: Emotions in Context-Aware Recommender Systems. Springer, Cham (2016) 311–326
27. Shan, M.K., Kuo, F.F., Chiang, M.F., Lee, S.Y.: Emotion-based music recommendation by affinity discovery from film music. *Expert Systems with Applications* **36**(4) (may 2009) 7666–7674
28. Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K.H., Schwaiger, R.: InCarMusic: Context-Aware Music Recommendations in a Car. Springer, Berlin, Heidelberg (2011) 89–100
29. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel Text Classification for Automated Tag Suggestion
30. Wiczorkowska, A., Synak, P., Raś, Z.W.: Multi-Label Classification of Emotions in Music. In: *Intelligent Information Processing and Web Mining*. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 307–315
31. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing* **2011**(1) (dec 2011) 4
32. Baltrunas, L., Amatriain, X.: Towards Time-Dependant Recommendation based on Implicit Feedback
33. Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A.: Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In: *Proceedings of the third ACM conference on Recommender systems - RecSys '09*, New York, New York, USA, ACM Press (2009) 265
34. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems* **23**(1) (jan 2005) 103–145
35. Lee, D., Park, S.E., Kahng, M., Lee, S., Lee, S.g.: Exploiting Contextual Information from Event Logs for Personalized Recommendation. Springer, Berlin, Heidelberg (2010) 121–139
36. Pacula, M.: A Matrix Factorization Algorithm for Music Recommendation using Implicit User Feedback
37. Koren, Y., Bell, R.: *Advances in Collaborative Filtering*
38. : Validación cruzada - Wikipedia, la enciclopedia libre

Sistema de Recomendación de Canciones Basado en Aspectos Emocionales

Carlos J. Gomes González, María N. Moreno García y Ana B. Gil

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{cgomes,mmg,abg}@usal.es
<http://diaweb.usal.es/>

Resumen La música en streaming ha abierto la posibilidad de acceder a cantidades inmensas de canciones, hecho que ha llevado a que los usuarios necesiten la ayuda de sistemas de recomendación personalizados para encontrar aquellas canciones que mejor se puedan adaptar a sus preferencias y gustos. Se han propuesto distintos sistemas de recomendación durante los últimos años, el que se va a tratar en este trabajo hace uso de características acústicas extraídas directamente por el análisis automático de canciones, las cuales permiten la clasificación en aspectos emocionales. Estos aspectos emocionales junto a información obtenida de las relaciones sociales entre los usuarios se utilizarán para generar recomendaciones e intentar mejorar los resultados obtenidos por los enfoques clásicos de recomendación que usan filtrado colaborativo basado en memoria. Además, se ha implementado una aplicación web que permite la interacción con los distintos usuarios de forma que se obtienen datos reales de prueba que son utilizados por el módulo de recomendación.

Keywords: Sistema de recomendación, Características acústicas, Clasificación de emociones, Filtrado colaborativo, Recomendación de canciones, Recomendaciones a grupos.

1. Introducción

Los avances en la tecnología han permitido que la música y las personas se encuentren cada vez más conectadas. En la actualidad existe la posibilidad de acceder a millones de canciones gracias a la conocida como música en streaming, sobre todo con servicios como Spotify¹. De esta forma podemos tener a nuestro alcance la música que nos gusta, pero también nos abre la posibilidad de explorar y descubrir nuevas canciones que podrán adaptarse a nuestras preferencias, a través de lo que se conoce como sistemas de recomendación.

Los sistemas de recomendación aplicados al ámbito de la música permiten a los usuarios descubrir de manera personalizada nuevos contenidos musicales según las preferencias de los mismos. Uno de los enfoques más utilizados en

¹ <https://www.spotify.com/>

la actualidad en los sistemas de recomendación utiliza las valoraciones que los usuarios hacen sobre los ítems (canciones).

En este trabajo se propone un sistema de recomendación de canciones que haga uso de aspectos emocionales, usando características tanto de las canciones como de los usuarios y que además, tenga en cuenta valoraciones dadas por los usuarios hacia esas canciones para poder aplicar la técnica de filtrado colaborativo (CF). Este sistema puede valerse de la extracción de características acústicas de las canciones para generar las recomendaciones, también puede utilizar emociones que se encuentran asociadas a la intensidad de algunas de esas características o fundamentarse en las relaciones sociales entre usuarios.

1.1. Objetivos

El principal objetivo de este trabajo es construir un sistema de recomendación de canciones usando CF que tenga en cuenta aspectos emocionales que puedan ayudar a que la recomendación sea más precisa y se adapte más a los gustos de los distintos usuarios. Además, comprobar su eficacia utilizando un conjunto de datos de usuarios reales que han hecho uso de una aplicación web implementada.

2. Estado del Arte

2.1. Música y Emociones

La generación de sentimientos o aspectos emocionales se encuentra estrechamente ligado a la música. En un estudio para comprobar la relación entre diferentes emociones y características acústicas [8], se clasificaron un conjunto de canciones en sentimientos de miedo, enfado, felicidad, tristeza y ternura usando características acústicas relacionadas con el timbre, ritmo y tonalidad.

En otro experimento [3] se buscaba comprobar el efecto del tempo (asociado a las pulsaciones por minuto (BPM)) en las emociones. Se presentaron a cada individuo extractos musicales, los cuales tenían que valorar siguiendo la escala de valencia y activación del SAM (*Self Assesment Manikin*) [10]. Con estos valores se clasificó el extracto según una emoción correspondiente con el Modelo Circumplejo de Russell [14]. El término valencia se asocia como la atracción (valencia positiva) o la aversión (valencia negativa) a un evento o situación. En cambio el valor de activación hace referencia al nivel de alteración por una emoción fuerte y que provoca que el pulso se acelere. La correlación entre BPM y la activación fue significativa, se obtuvo que un incremento de BPM implica un incremento en la activación, en cambio la correlación entre BPM y valencia no obtuvo resultados significativos.

En la conferencia anual de recuperación y evaluación de información musical (MIREX) de 2007 [7] se incluyó una tarea de clasificación de canciones por estados de ánimo. Se distribuyeron las canciones en conjuntos de emociones y para la clasificación se ignoraron las letras de las canciones y se redujeron a 30 segundos cada una de ellas. Se aplicaron varios algoritmos de clasificación, los

sistemas que usaban el algoritmo SMO [13] obtuvieron una exactitud del 58 %, LibSVM² 61 % y K-nearest neighbor 47 %.

2.2. Sistemas de Recomendación

Las principales metodologías de recomendación [4] y que son las más extendidas se clasifican en dos:

- **Filtrado Colaborativo (*Collaborative Filtering*)**: consiste en recomendar a un usuario concreto artículos que gustan a usuarios con preferencias similares. Existen dos enfoques de filtrado colaborativo, el basado en memoria (usan toda la base de datos en el momento de la recomendación) y el basado en modelos (se crea un modelo a partir de la base de datos).
- **Basadas en contenido (*Content-based filtering*)**: se recomiendan artículos similares a los que gustaron al usuario en el pasado sin tener en cuenta otros usuarios para la recomendación.

El CF se aplica construyendo una matriz M , con m ítems o elementos y n usuarios. Esta matriz contiene en su interior las valoraciones (valoraciones explícitas, número de reproducciones, etc.) que cada usuario ha asignado a ese ítem.

El filtrado colaborativo basado en memoria puede presentar dos enfoques:

- **Basado en usuarios (*User-Based*)**: en este caso, para un usuario concreto o usuario activo se buscan usuarios similares en función de las valoraciones que estos hayan dado a los ítems y se recomiendan ítems que esos usuarios similares hayan valorado con buenas puntuaciones.
- **Basado en ítems (*Ítem-Based*)**: en este enfoque se selecciona un ítem concreto y se buscan otros ítems similares usando las valoraciones que los usuarios le hayan dado. Posteriormente se tiene en cuenta para la predicción las valoraciones del usuario activo y se le recomiendan ítems similares a los que él valoró positivamente en el pasado.

Un avance importante para los sistemas de recomendación fue la llegada de la Web 2.0 [6], donde los usuarios no solo consumen productos, sino que además generan opiniones o valoraciones sobre sus experiencias con los servicios consumidos.

2.3. Trabajos Relacionados

En un estudio [9] se realizó una evaluación durante un mes a un total de 54 individuos utilizando distintos métodos de recomendación. Se probaron recomendaciones basadas en los géneros de las canciones, usando filtrado colaborativo y filtrado colaborativo con un filtro de emociones. Los resultados consiguieron mejores resultados aplicando el filtro de las emociones.

² <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Otro estudio [1] utilizó un sistema híbrido de recomendación de artistas teniendo en cuenta aquellos artistas que el usuario activo de la aplicación escucha frecuentemente o selecciona directamente en la interfaz.

En la actualidad existen numerosas páginas que ofrecen sus servicios para la clasificación de canciones ya sea por géneros o por emociones, como es el caso de los servicios ofrecidos por 'Musicoverly'³, 'AllMusic'⁴ o 'Rockola'⁵ y que también utilizan el modelo bidimensional de valencia y activación para conseguir la clasificación.

Un estudio muy interesante [16] consigue la detección de estados emocionales según información inferida del contexto de un usuario en un momento determinado.

3. Sistema de Recomendación

3.1. Arquitectura

En la Figura 1 se encuentra la arquitectura cliente-servidor de la aplicación web, la cuál, a través de peticiones HTTP y haciendo uso del servicio de Spotify mediante su API, se encarga del almacenamiento de canciones, la extracción de características acústicas y la posterior clasificación por emociones de cada una de ellas. Además cuenta con un módulo de recomendación que se encarga de obtener los datos almacenados para poder realizar predicciones personalizadas a los usuarios.

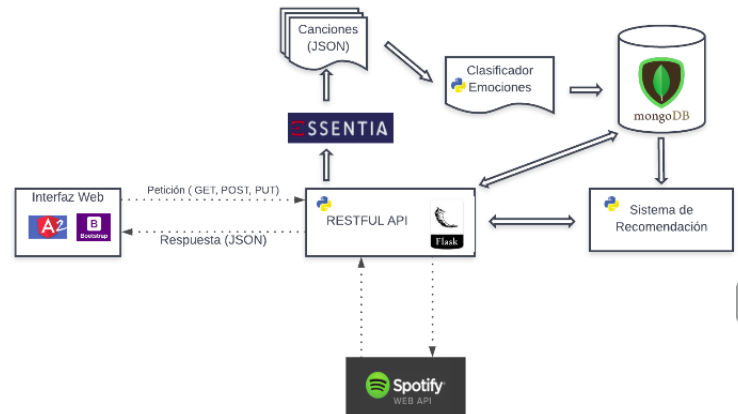


Figura 1: Arquitectura de la Aplicación Web

³ <https://http://b2b.musicoverly.com/>

⁴ <https://www.allmusic.com/>

⁵ <http://www.rockola.fm/>

3.2. MoodSically

La idea original de MoodSically [5] era crear un administrador de contenido musical y personal para la detección automática de emociones, utilizando características acústicas extraídas de canciones subidas por los usuarios. Se ha mejorado esta idea integrando el análisis automático de canciones con el servicio de música en streaming Spotify e introduciendo un modulo que implementa un sistema de recomendaciones personalizadas que hace uso de esos aspectos emocionales y/o características acústicas. Además, la aplicación web tiene la funcionalidad de actuar como red social mediante la opción de seguimiento de usuarios. El análisis y extracción de características acústicas se realiza gracias al uso de las bibliotecas Essentia [2] y Gaia⁶ dentro de la arquitectura del servidor,

Se almacenan descriptores de bajo nivel como el modo, capacidad de baile, pulsaciones por minuto (BPM), o el volumen; y descriptores de alto nivel como el género musical, el timbre, la tonalidad o una clasificación de probabilidad de emociones (tristeza, felicidad, fiesta o relajada).

3.3. Clasificador de emociones

La clasificación de las canciones por emociones se realiza usando valores de valencia y activación que se obtienen a partir de valores de probabilidad de tristeza, felicidad, fiesta y relajación [5] extraídos como descriptores de alto nivel. La valencia (Ecuación 1) se calcula restando la probabilidad de que la canción sea triste (PSad) a la probabilidad de que sea feliz (PHappy). La Activación (Ecuación 2) se obtiene restando la probabilidad de que la canción sea relajada (PRelax) a la probabilidad de que sea una canción de fiesta (PParty).

$$Valencia = PSad - PHappy \quad (1)$$

$$Activacion = PRelax - PParty \quad (2)$$

Una vez que el ángulo (Ecuación 3) y el valor de la distancia (Ecuación 4) se han calculado, se obtienen las coordenadas polares.

$$\theta = \frac{Valencia}{Activacion} \quad (3)$$

$$r = \sqrt{Valencia^2 + Activacion^2} \quad (4)$$

Las canciones almacenadas se encuentran clasificadas por alguna de las 12 emociones contempladas (feliz, exaltado, emocionado, activo, enojado, frustrado, triste, deprimido, aburrido, calmado, relajado y sereno) siguiendo el Circumplejo de Russell de 12 segmentos [17] (Figura 2)

⁶ <https://github.com/MTG/gaia>

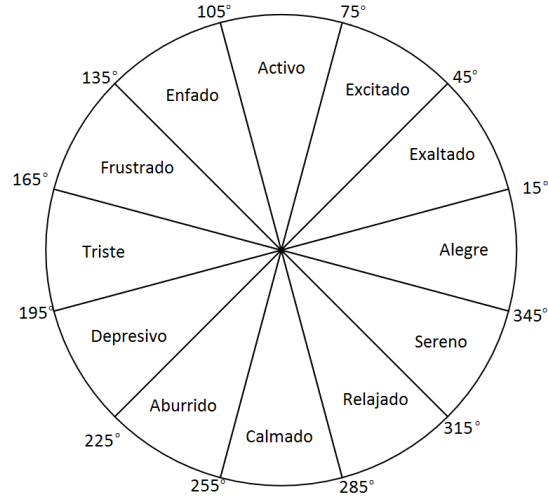


Figura 2: Circumplejo de Russell de 12 Segmentos

3.4. Metodología de Recomendación de Canciones

Para realizar la recomendación de canciones se utilizó el enfoque de filtrado colaborativo (CF) [4] junto al filtrado basado en contenido, añadiendo atributos tanto de las canciones como de los usuarios para la obtención de similitudes.

Para obtener las similitudes tanto entre ítems como entre usuarios se utilizó la medida de distancia del coseno. También se aplicó el algoritmo k-nearest neighbors (k-NN) o de vecinos más cercanos. El objetivo es encontrar los k usuarios (vecinos) con el mayor coeficiente de similitud con el usuario activo (usando tanto las valoraciones explícitas, como los atributos adicionales), creando un subconjunto de usuarios para la aplicación del filtrado colaborativo (Ecuación 5).

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}} \quad (5)$$

donde $P_{a,i}$ es la predicción de la valoración que el usuario activo a le asignaría ítem i , $w_{a,u}$ es la similitud entre el usuario activo y el usuario u , K es el subconjunto de k-usuarios similares, $r_{u,j}$ es la valoración del vecino u al ítem j y \bar{r} es la media de valoraciones de un usuario.

El enfoque de CF basado en ítems utiliza la matriz traspuesta (ítems x usuarios) para las predicciones. Este enfoque sigue la ecuación 6, donde se tienen en cuenta las valoraciones realizadas por el propio usuario activo sobre k canciones similares a la que se va a realizar la predicción.

$$P_{a,i} = \frac{\sum_{j \in K} r_{a,j} \times w_{i,j}}{\sum_{j \in K} w_{i,j}} \quad (6)$$

donde $P_{a,i}$ es la predicción para el usuario activo a sobre el ítem i , K es el conjunto de ítems similares al ítem i y $w_{i,j}$ es la similitud entre los ítems i y j .

3.5. Recomendación a Grupos

La aplicación web generada actúa como red social mediante la opción de seguimiento de usuarios. Varios estudios han demostrado la influencia de las redes sociales en las recomendaciones ([11],[15]). La técnica utilizada para la recomendación de grupos consiste en calcular las predicciones de ratings para el usuario activo y posteriormente los ratings para cada uno de los usuarios seguidos por este usuario activo. Se utiliza la estrategia de Maximización de la satisfacción promedio (Ecuación 7), donde el rating del grupo se calcula como la media de los ratings individuales.

$$R_i = \frac{1}{n} \sum_{u \in G} r_{u,i} \quad (7)$$

donde R_i es el rating del grupo, u es cada uno de los usuarios del grupo G para los que se van a tener en cuenta las predicciones, $r_{u,i}$ es la predicción para el usuario u sobre el ítem i y n es el número de usuarios del grupo G .

4. Conjunto de Datos

El conjunto de datos utilizado contiene información de usuarios que han interactuado con la aplicación y que han valorado canciones que ellos mismos han escuchado recientemente.

Conjunto de Datos	
	Cantidad
Usuarios de la Aplicación	20
Canciones almacenadas	1191
Relaciones usuario-canción	1306
Relaciones usuario-usuario	28

Tabla 1: Información sobre el conjunto de datos utilizado

Debido a limitaciones establecidas por el servicio de streaming, solo es posible extraer 30 segundos de canción para el análisis acústico, hecho que podría provocar problemas en la clasificación de emociones. Para contrastar esto se han analizado las diferencias entre la clasificación de las primeras 10 canciones (Tabla 2) del conjunto de datos utilizando usando 30 segundos y la clasificación utilizando la totalidad de la canción. En la mayoría de los casos la clasificación fue la misma, las diferencias ocurren por variaciones en los valores de los atributos tanto de alto como de bajo nivel que se usan para obtener valores de valencia

y activación. Estas discrepancias se reflejan como una clasificación en emociones adyacentes utilizando el Circumplejo de Russel de 12 segmentos (Figura 2). Adicionalmente, en la conferencia anual de recuperación y evaluación de información musical (MIREX) de 2007 [7], las canciones también fueron reducidas a 30 segundos, obteniendo buenos resultados.

Canciones (Titulo-Artista)	Emoción(30 sec)	Emoción
Future Starts Slow-The Kills	Excitado	Excitado
Nellie the elephant-Toy Dolls	Excitado	Exaltado
Wait-M83	Aburrido	Calmado
El Tiburón (The Shark)-Henry Mendez	Excitado	Excitado
Cien Gaviotas-Duncan Dhu	Exaltado	Exaltado
El pensamiento circular-Ivan Ferreiro	Calmado	Calmado
Estadio Azteca-Andrés Calamaro	Deprimido	Deprimido
Morning Glory-Oasis	Exaltado	Exaltado
Love Illumination-Franz Ferdinand	Excitado	Exaltado
Caminito a Motel-Taburete	Deprimido	Aburrido

Tabla 2: Comparación de clasificación de emociones 10 primeras canciones

En lo que respecta a las emociones generadas automáticamente y asociadas a las canciones del conjunto de datos, podemos observar en la Figura 3 que existe gran variedad en la clasificación, destacando que no se ha conseguido clasificar ninguna canción asociada a las emociones de enfado o frustración.

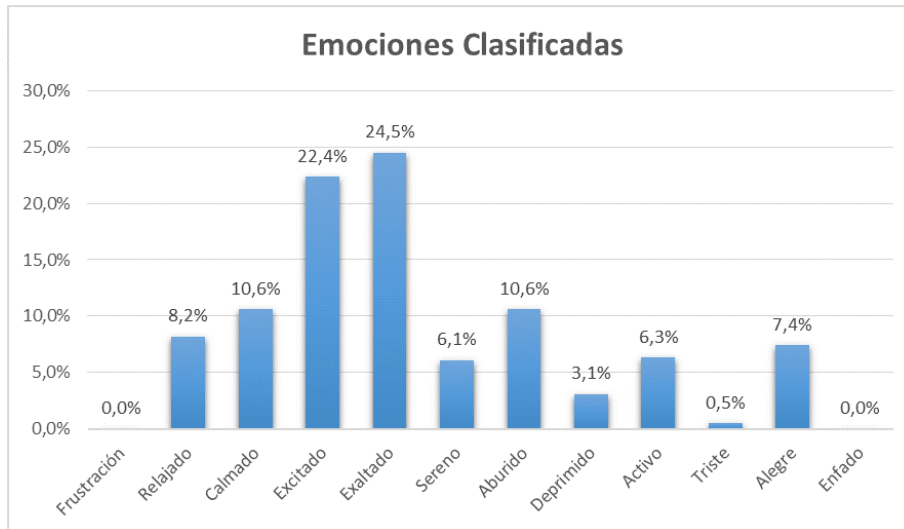


Figura 3: Frecuencia de aparición de emociones en el conjunto de datos

En la aplicación se almacena una puntuación que indica el nivel de conformidad de los usuarios con la emoción generada por el sistema. Si el usuario está totalmente de acuerdo con la clasificación lo valorará con un 5, utilizando un sistema de valoración por intervalos del 1 al 5 (representación por estrellas) (Figura 4), mientras que si está en total desacuerdo lo valorará con un 1.

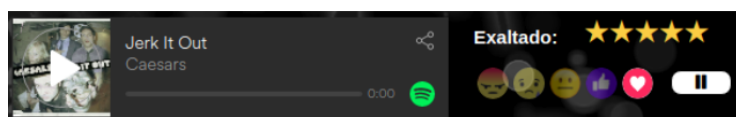


Figura 4: Sistema de Valoración y Evaluación de Canciones

Las evaluaciones aportadas en cuanto a la clasificación de las emociones nos deja el resultado de la Figura 5, donde se puede comprobar que en general, existe una buena asociación entre la emoción y la canción según la perspectiva de los usuarios, con puntuaciones de 3, 4 o 5 y un 74,9% del total de evaluaciones.

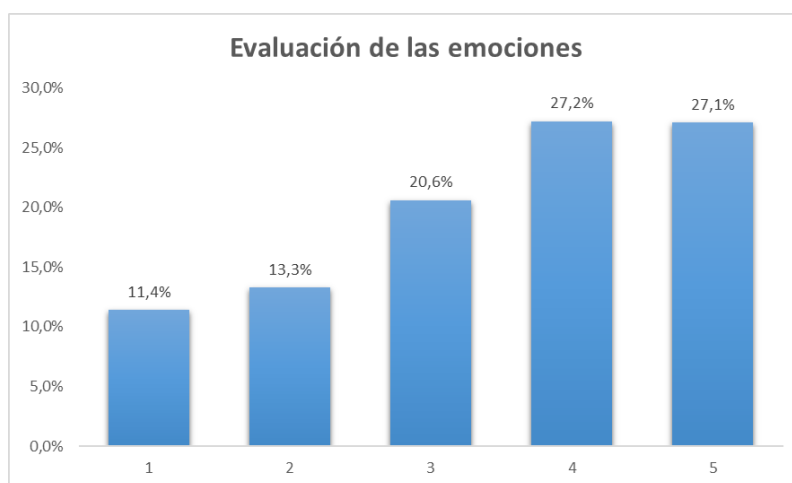


Figura 5: Evaluaciones hacia las emociones generadas automáticamente

En la Figura 4 también se puede observar un mecanismo de valoración mediante emoticonos. Este sistema permite establecer una asociación de gustos entre el usuario y alguna de las últimas 50 canciones que ha escuchado a través de Spotify. Esta valoración se usará como el rating necesario para aplicar en las Formulas (5 y 6) de filtrado colaborativo.

Como información adicional que va a ser utilizada por el sistema de recomendación, para cada usuario se almacenan como atributos el número de canciones

asociadas a cada emoción que este ha escuchado. Además, de las canciones se almacenan descriptores tanto de bajo como de alto nivel (bpm, disonancia, emoción generada por el sistema, probabilidad de emociones [feliz, fiesta, relajado y triste], timbre, tonal y volumen).

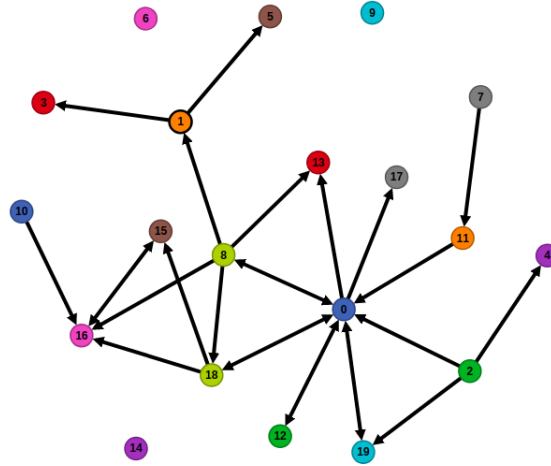


Figura 6: Relaciones de seguimiento entre usuarios de la aplicación

En la Figura 6 están representadas las relaciones entre usuarios que han usado la aplicación, en la que cada nodo representa a un usuario y cada eje dirigido simboliza que un usuario ha seguido a otro. En total existen 28 relaciones de seguimiento en este conjunto de datos.

5. Resultados y Discusión

En este apartado se presentan los resultados de validación del sistema de recomendación, aplicando distintos experimentos de recomendaciones individuales o a grupos. Para analizar estos resultados se utilizó la métrica de evaluación de Root Mean Squared Error (RMSE - Ecuación 8) [12]. Esta medida sirve para cuantificar el error producido entre la predicción hecha de la valoración y la valoración real.

$$RMSE = \sqrt{\frac{1}{N} \sum (x_i - \hat{x}_i)^2} \quad (8)$$

El error utilizando esta métrica y para este sistema de valoraciones puede variar en el rango [0,4]. También se puede utilizar el valor normalizado de este error, el cual se encuentra en el rango [0,1] y que se obtiene dividiendo el error por la diferencia entre el valor máximo de valoración (5) y el valor mínimo (1).

Para la validación de los resultados se ha utilizado la técnica de *cross-validation* o validación cruzada. Se ha utilizado el enfoque que consiste en realizar 10 subconjuntos del conjunto de datos (*10-fold cross validation*).

5.1. Experimentos Base

Para el experimento base se han utilizado los dos enfoques CF basado en memoria, tanto el basado en usuarios como en ítems o canciones, modificando los valores de vecindad (k). El error es más bajo para el enfoque basado en usuarios (Tablas 3 y 4), La variación en el número de vecinos no es muy significativa para el error.

Enfoque	$k=1$	$k=5$	$k=10$	$k=15$
Basado en Usuarios	0.909415	0.905285	0.904633	0.904643

Tabla 3: RMSE para el filtrado colaborativo basado en usuarios

Enfoque	$k=10$	$k=50$	$k=100$	$k=200$
Basado en Ítems	1.077381	1.019513	1.016611	1.016611

Tabla 4: RMSE para el filtrado colaborativo basado en ítems

5.2. Aplicación de aspectos emocionales

Para el enfoque de CF basado en usuarios, además de tener en cuenta las valoraciones de los usuarios para encontrar usuarios similares, se tuvieron en cuenta las emociones de canciones escuchadas por esos usuarios. Se calcula el porcentaje de cada una de las 12 emociones generadas automáticamente del total de canciones escuchadas por cada uno de los usuarios y se añade una columna nueva por cada emoción.

En las siguientes pruebas se analizó el error para distintos experimentos teniendo en cuenta el enfoque basado en ítems. A continuación se describen los experimentos realizados, manteniendo en todos ellos el experimento base:

- **Experimento 1:** Se añaden valores de probabilidad de que una canción sea de alguna de estas cuatro emociones: Feliz, Fiesta, Relajado, Triste. Estas probabilidades se obtienen como descriptores de alto nivel en el análisis acústico.
- **Experimento 2:** al igual que en el experimento anterior se tienen en cuenta las probabilidades de las emociones, pero se añade un nuevo atributo que consiste en la media aritmética de evaluaciones por parte de los usuarios sobre las emociones de cada una de las canciones.
- **Experimento 3:** partiendo del experimento 2, pero teniendo en cuenta únicamente la media de las evaluaciones sobre las emociones.
- **Experimento 4:** Se tienen en cuenta algunas de las características acústicas que más influyen en la determinación de aspectos emocionales (volumen, disonancia, timbre, bpm y tono).

Enfoque	k=1	k=5	k=10	k=15
Basado en Usuarios	0.909414	0.899356	0.889894	0.880654

Tabla 5: RMSE usando el enfoque basado en usuarios y porcentaje de emociones

Experimento	k=10	k=50	k=100	k=200
Experimento 1	0.825084	0.919130	0.940295	0.960806
Experimento 2	0.839491	0.898071	0.913458	0.939962
Experimento 3	0.954245	0.915110	0.942264	0.942907
Experimento 4	0.715693	0.799224	0.846094	0.892229

Tabla 6: RMSE experimentos utilizando enfoque basado en ítems

Los resultados del error a la vista de la Tabla 5 han mejorado muy poco con respecto al error del enfoque base, posiblemente debido al número tan pequeño de usuarios (20) y el número tan grande de canciones (1191) que se están teniendo en cuenta.

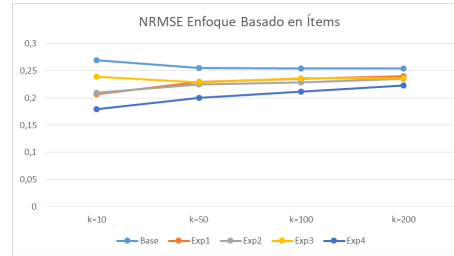


Figura 7: NRMSE utilizando distintos experimentos y enfoque basado en ítems

El error de los experimentos (Tabla 6) mejora los resultados obtenidos por el enfoque base (Tabla 4). En la Figura 7 podemos apreciar los valores normalizados de error. El experimento en el que se han obtenido mejores resultados es el 4, usando como atributos el conjunto de características acústicas extraídas como descriptores de alto nivel, se ha logrado una mejoría de hasta casi un 8 % en el caso de k=10.

5.3. Implementación de recomendaciones a grupos

Se pueden realizar predicciones usando el conjunto de usuarios que un usuario activo sigue. En este caso, la recomendación se realizaría para el usuario activo y para todos los usuarios que este ha elegido seguir.

Enfoque	k=1	k=5	k=10	k=15
Basado en Usuarios	0.909414	0.899356	0.889894	0.880654
Grupos	0.935997	0.897672	0.851857	0.825483
Grupos con Emociones	0.925370	0.789532	0.655712	0.500502

Tabla 7: RMSE para el filtrado colaborativo basado en usuarios

La recomendación individual se realiza usando tres enfoques que se explican a continuación:

- **Basado en Usuarios:** experimento base donde se utiliza el enfoque de filtrado colaborativo basado en usuarios, además, se utiliza como atributos el porcentaje de cada una de las emociones asociadas a las canciones. La recomendación es individual y no se tienen en cuenta las relaciones entre usuarios.
- **Grupos:** se realiza la recomendación a grupos usando como recomendación individual el filtrado colaborativo basado en usuarios, teniendo en cuenta únicamente las valoraciones explícitas de los usuarios. En la recomendación a grupos se utiliza el método de maximización de la satisfacción promedio descrito en el apartado 3.5.
- **Grupos con Emociones:** se realiza la recomendación a grupos usando como recomendación individual el filtrado colaborativo basado en usuarios teniendo en cuenta el porcentaje de emociones asociadas a las canciones.

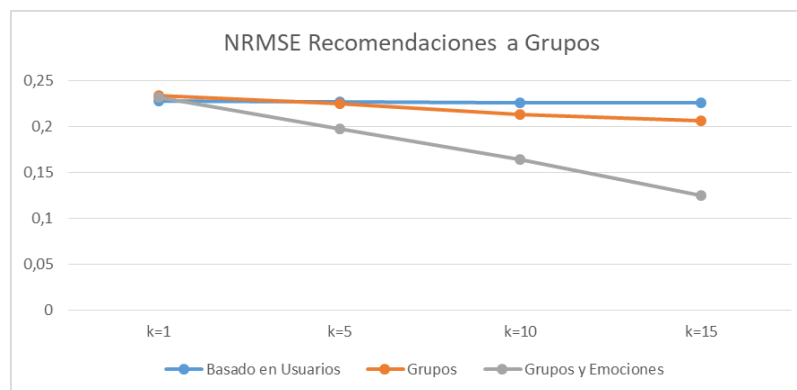


Figura 8: NRMSE utilizando recomendaciones a grupos

A la vista de los resultados (Figura 8), el error normalizado disminuye hasta en un 10% usando el enfoque de grupos con emociones y k=15, siendo este un resultado muy significativo.

5.4. Encuesta de validación de recomendaciones

Algunos de los usuarios de la aplicación han aportado sus opiniones sobre aquellas canciones que el sistema ha devuelto como posibles recomendaciones. Estas opiniones se realizan a través de un sistema binario mediante botones, en donde uno de ellos indica si le gusta la canción recomendada y el otro justo lo contrario.

Para evaluar los resultados obtenidos se ha utilizado una medida de evaluación (Ecuación 9). Los valores que puede tomar esta medida se encuentra en el rango $[-1,1]$, en el que los resultados positivos y más cercanos a 1 reflejan buenas recomendaciones.

$$E = \frac{(R^+ - R^-)}{n} \quad (9)$$

donde el valor de exactitud E es la diferencia de las canciones valoradas positivamente R^+ y las canciones valoradas negativamente R^- y n es el número de canciones recomendadas.

Los datos generales de la encuesta realizada a los distintos usuarios los podemos encontrar en la Tabla 8.

	Cantidad
Usuarios que han aportado valoraciones	10
Valoraciones totales	188
Valoraciones enfoque basado en usuarios	96
Valoraciones enfoque basado en ítems	92

Tabla 8: Datos de valoraciones a las recomendaciones

Se podían valorar recomendaciones usando el enfoque colaborativo basado en usuarios con el porcentaje de emociones para la búsqueda de usuarios similares (Sección 5.2) con $k=10$. También se podían evaluar recomendaciones usando el Experimento 2 (Sección 5.2) y $k=100$.

	Exactitud
Enfoque basado en usuarios	0.315
Enfoque basado en ítems	0.304
Ambos enfoques	0.308

Tabla 9: Evaluación para los distintos enfoques

A la vista de los resultados de la Tabla 9 ambos enfoques obtuvieron resultados similares en base a los datos de la encuesta, con una cierta mejoría en el enfoque basado en usuarios. Los resultados son aceptables dentro del rango de

valores establecidos, considerando el problema del tamaño del conjunto de datos y el elevado valor de dispersión de las valoraciones.

6. Conclusiones

En conclusión, se ha implementado una aplicación web que permite de forma automática la extracción de características acústicas de las canciones y la clasificación por emociones de las mismas, todo ello usando información de las canciones escuchadas por los usuarios y aportada por la API de Spotify. Además, se ha implementado un módulo de recomendación haciendo uso de los dos enfoques de filtrado colaborativo básicos basados en memoria (Basado en usuario y en ítems). Estos enfoques fueron mejorados usando métodos híbridos y el uso de atributos tanto de características acústicas como de emociones o relaciones entre usuarios.

A la vista de los resultados podemos concluir que el enfoque basado en ítems consigue mejores resultados frente al enfoque basado en usuarios, sobre todo con la utilización de atributos de características musicales como el bpm, tono, volumen o disonancia y valores pequeños del número de vecinos.

Resultan interesantes los buenos resultados obtenidos en las recomendaciones a grupos de usuarios utilizando las probabilidades de las emociones, con una mejoría de casi el 10 % en uno de los casos de estudio. A pesar de que estas recomendaciones exigen el cálculo de predicciones a distintos usuarios, lo que puede resultar poco eficiente computacionalmente, en el caso de pocos usuarios es viable y mejora los resultados de error con respecto a experimentos anteriores.

En cuanto a la encuesta, podemos concluir que los resultados para ambos enfoques de filtrado colaborativo son aceptables, logrando así recomendar canciones afines a los usuarios activos que han usado la aplicación web.

En general se puede decir que la relación entre las emociones y la música es innegable, estos aspectos permiten mejorar las recomendaciones personalizadas a usuarios. Además, las relaciones entre esos usuarios puede aportar información de gran utilidad para mejorar aun más esas recomendaciones.

6.1. Líneas de trabajo futuras

Como líneas de trabajo futuras se puede utilizar la información de las evaluaciones sobre las emociones para la búsqueda de grupos de usuarios, ya que esto es un aspecto subjetivo (Un usuario puede creer que una canción es aburrida cuando para otro es todo lo contrario). También se puede mejorar el algoritmo de clasificación de emociones utilizando otros aspectos para la recomendación, pero sobre todo se puede aplicar a recomendaciones basadas en contexto, teniendo en cuenta el estado anímico del usuario en momentos determinados.

Referencias

1. Ivana Andjelkovic, Denis Parra, and John O'Donovan. Moodplay: Interactive mood-based music discovery and recommendation. In *Conference: User Modeling, Adaptation and Personalization, At Halifax*. 07 2016.

2. Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, Jose Zapata, and Xavier Serra. Essentia: An open-source library for sound and music analysis. In *Proceedings - 21st ACM International Conference on Multimedia*. 10 2013.
3. Jonathan Azael Caballero-Meneses and Marina Menez. Influencia del tempo de la música en las emociones. *Revista Colombiana de psicología*, pages 37–44, 2010.
4. Oscar Celma. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. 01 2010.
5. Jorge García Vicente et al. Moodsically. personal music management tool with automatic classification of emotions. In *International Symposium on Distributed Computing and Artificial Intelligence 15th International Conference (DCAI 2018)*. Springer Berlin Heidelberg. (in press).
6. Simon Fong, Chen Wei, and Richard Khoury. Web 2.0 recommendation service by multi-collaborative filtering trust network algorithm. In *Information Systems Frontiers*, volume 15, pages 533–551. 08 2013.
7. Xiao Hu, J Downie, Cyril Laurier, Mert Bay, and Andreas F. Ehmann. The 2007 mirex audio mood classification task: Lessons learned. In *ISMIR 2008 - 9th International Conference on Music Information Retrieval*, pages 462–467. 01 2008.
8. C Laurier, Olivier Lartillot, Tuomas Eerola, and Petri Toiviainen. Exploring relationships between audio features and emotion in music. In *Frontiers in Human Neuroscience - FRONT HUM NEUROSCI*, pages 260–264. 01 2009.
9. Mortensen M., Gurrin C., and Johansen D. Real-world mood-based music recommendation. In *Information Retrieval Technology. AIRS 2008*, pages 514–519. 2008.
10. Margaret M. Bradley and Peter J. Lang. Measuring emotion: The self-assessment manikin and the semantic differential. In *Journal of behavior therapy and experimental psychiatry*, volume 25, pages 49–59. 04 1994.
11. Diego Sánchez Moreno, Javier Pérez Marcos, Ana B. Gil González, Vivian López Batista, and María N. Moreno-García. Social influence-based similarity measures for user-user collaborative filtering applied to music recommendation. (in press).
12. Denis Parra and Shaghayegh Sahebi. Recommender systems: Sources of knowledge and evaluation metrics. In *Studies in Computational Intelligence*, volume 452, pages 149–175. 01 2013.
13. J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.
14. James Russell. A circumplex model of affect. In *Journal of Personality and Social Psychology*, volume 39, pages 1161–1178. 12 1980.
15. Khaled Sellami, Mohamed Ahmed-Nacer, and Pierre F. Tiako. From social network to semantic social network in recommender system. *CoRR*, abs/1407.3392, 2014.
16. Marko Tkalcic, Andrej Kosir, and Jurij Tasic. Affective recommender systems: The role of emotions in recommender systems. In *CEUR Workshop Proceedings*, volume 811. 01 2011.
17. M. Yik, Russell, J. A., and J. H Steiger. A 12-point circumplex structure of core affect. *Emotion*, pages 705–731, 2011.

Sistema de Monitorización y Detección de Patrones Anómalos Mediante el Uso de Sensores Médicos

Christian Bonal Martín, Juan Manuel Corchado Rodríguez, Gabriel Villarrubia González, Juan Francisco de Paz Santana

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{cbonalm, corchado, gvg, fcofds}@usal.es
<http://diaweb.usal.es/>

Resumen Gracias al gran avance tecnológico que está sufriendo la sociedad hoy en día, se han creado diferentes sensores médicos para la monitorización de los usuarios que han permitido mejorar las tareas de seguimiento de sus constantes vitales en su vida diaria, reduciendo la asistencia de éstos a centros de salud para realizar tareas de medición y control rutinario. El principal objetivo de este proyecto se basa en la creación de una arquitectura de teleasistencia de bajo coste que permita a personas que padecen algún tipo de enfermedad crónica la medición de diferentes constantes vitales para así poder realizar un seguimiento diario de su enfermedad. Una vez creado, se probará la efectividad de este prototipo mediante su uso en 10 pacientes, de una edad comprendida entre los 18 y 58 años, como parte de un proceso de prueba antes de dedicarse a pacientes crónicos.

Keywords: eHealth, Cooking Hacks, Bajo coste, Monitorización

1. Introducción

Hoy en día, si se mantuviesen las tendencias demográficas actuales, la población de España descenderá ininterrumpidamente a partir del año 2017, perdiendo aproximadamente un 1,2 % de su población en 15 años (552.245 habitantes). La causa principal se debe a un crecimiento vegetativo negativo, ya que el número de defunciones aumentará en consecuencia del envejecimiento de la población, mientras que el número de nacimientos disminuirá [7].

En el año 2016, la población de edad avanzada en España suponía un 18,7 % (8,7 millones de habitantes), [8] cifra ligeramente inferior al conjunto de la UE (19,2 %), siendo Galicia, el Principado de Asturias y Castilla y León (todas ellas con algo menos del 24 %) las regiones más envejecidas [9]. Esta tendencia incremental se viene dando desde el año 2008 debido principalmente a un aumento de la esperanza de vida de las personas.

Según el Instituto Nacional de Estadística (INE) en *Península Ibérica 2017 en cifras*,[9] “ En 2015, España era el país de la UE con mayor esperanza de

vida para las mujeres: 85,7 años (...) En el caso de los hombres, ese mismo año, los valores calculados para España (80,1 años) también superaban a la media de la UE 28 (77,9 años)". Estos valores, siguiendo las proyecciones realizadas por el INE [7], aumentarán en el futuro, llegando a alcanzar los 83,2 años en los varones y los 87,7 en las mujeres en 2031. Esto, unido a un descenso del número de nacimientos (concretamente un 22% menos que en los pasados 15 años), produce que se lleve a cabo un desplazamiento en la pirámide de población de España, siendo el grueso de la población de España cada vez más mayor. Esto, unido a que las personas ancianas constituyen un grupo de riesgo para algunas enfermedades debido a su edad, siendo en muchas ocasiones muy dependientes, hace que se antoje imprescindible el poder realizar instrumentos que les faciliten su vida. Surge por ello la idea de desarrollar este sistema de monitorización, con el objetivo de ofrecer al hogar del mayor un instrumento sencillo y manejable que les ahorre desplazamientos a sus respectivos centros médicos para cuestiones tan simples como el chequeo de constantes vitales.

El planteamiento propuesto para el desarrollo de este trabajo se basa en el estudio del uso de diferentes sensores médicos que nos permitan realizar una monitorización remota, analizando la posibilidad de crear ciertas reglas que nos permitan detectar diversos patrones anómalos con el fin de crear un sistema de alertas utilizando diferentes técnicas de Inteligencia Artificial. Por lo tanto, la investigación se centrará en obtener diversas características sobre distintas enfermedades con el fin de crear un motor de reglas para el sistema de alertas, analizando posteriormente diversos algoritmos de clasificación para así obtener el más adecuado para este caso de estudio en cuestión.

Este trabajo se trata de una continuación del Trabajo de Fin de Grado presentado por el estudiante Christian Bonal Martín, titulado *Sistema de teleasistencia para pacientes crónicos*. En él se presentó un sistema formado por un conjunto de sensores de bajo coste que permiten medir diferentes constantes vitales, así como un sistema web que permite la monitorización remota del paciente por parte del médico.

2. Estado del arte

2.1. Patrones anómalos

Se entiende por patrones anómalos aquellas desviaciones que se producen en las medidas sobre un modelo de persona sana y que suelen estar asociados normalmente con enfermedades. Existen diferentes patrones para cada medida, destacando en este caso los patrones anormales en los electrocardiogramas (ECG), en la medición de la saturación de oxígeno y en la medición de la tensión arterial.

Electrocardiograma: El electrocardiograma (ECG) es un gráfico en el que se registran las diferencias en el potencial eléctrico generado por el corazón, recogidas a través de unos electrodos situados en la piel. Esta señal está se determina

por los potenciales de acción generados por millones de células individuales y su secuencia de activación [13].

El patrón básico del ECG comprende tres ondas, además de dos intervalos de tiempo entre las ondas, los cuales son: [6]

- **Onda P:** Onda de deflexión que representa la despolarización auricular.
- **Onda compleja QRS:** Representa la despolarización ventricular, formado por tres ondas:
 - **Onda Q:** Representa la despolarización del tabique interventricular.
 - **Onda R:** Refleja la despolarización de la masa principal de los ventrículos.
 - **Onda S:** Hace referencia a la despolarización final de los ventrículos.
- **Onda T:** Representa la repolarización ventricular.
- **Intervalo PR:** Tiempo existente entre la primera deflexión de la onda P y la primera desviación del complejo QRS.
- **Segmento ST:** Tiempo existente entre el final del complejo QRS y el inicio de la onda T.

En el electrocardiograma se pueden producir numerosos patrones anómalos que pueden derivar en diversas enfermedades. Algunos de ellos son:

- **Ritmo sinusal:** Se corresponde con el ritmo más común en una persona adulta, con una frecuencia cardíaca entre 60 y 100 latidos/min. Sus características principales se basan en la existencia de una onda compleja QRS estrecha, así como una onda P positiva en la derivación -2 [13].
- **Taquicardia:** Se corresponde con aquellas arritmias que producen una frecuencia cardíaca superior a 100 latidos/min [10]. Es importante distinguir entre dos tipos: taquicardia sinusal, que es un ritmo sinusal con una frecuencia cardíaca generalmente situada entre 100 y 170 latidos/min; y taquicardia auricular, que se caracteriza por tener un número de pulsaciones muy alto, concretamente entre 170 y 250 latidos/min [18], muy regulares y con un complejo QRS muy estrecho.
- **Aleteo auricular:** Es un tipo de arritmia que sucede debido a la progresión de impulsos eléctricos que entran por las aurículas realizando un trayecto circular [11]. Este aleteo posee una frecuencia cardíaca muy alta, entre 250 y 300 latidos/min. Además, la forma que de su gráfica, muestra una forma de dientes de serrucho entre los complejos QRS.
- **Fibrilación auricular:** Es la arritmia más frecuente en la práctica clínica [14], especialmente en personas mayores. Se caracteriza por ser caótico, irregular, con reconocibles complejos QRS y con ausencia de ondas P. La eficacia de las aurículas se pierde y existe un alto riesgo de que se produzca un trombo intracardiaco si esta arritmia persiste más de 48 horas [13].
- **Bradycardia sinusal:** Es una arritmia que se caracteriza por una secuencia normal de activación, con un impulso que surge en el nodo sinusal y una onda P normal, teniendo una frecuencia inferior a 60 latidos/min [18]. Si la frecuencia cardíaca es muy lenta (inferior a 30 a 40 latidos/min en personas de edad avanzada) puede existir aturdimiento o incluso poder llegar a sufrir un síncope [13].

- **Pausa sinusal:** Se trata de un tipo de arritmia que sucede cuando el nódulo SA no inicia los impulsos eléctricos, produciéndose un bloqueo en la conducción que rodea el nódulo sinusal. [18].

Tensión arterial: Se entiende por tensión arterial (o presión arterial) la fuerza que ejerce la sangre contra las paredes de las arterias. La medición de esta constante vital se refleja con dos presiones: presión sistólica (número más alto), que se produce cada vez que el corazón late para bombear la sangre a las arterias, y presión diastólica (número más bajo), la cual se produce entre latidos, cuando el corazón esta en reposo.

Cabe destacar que la presión arterial sigue un ciclo circadiano ya que no se mantiene constante durante el día, sino que se producen variaciones en sus valores según la fase del día en la que se encuentre el individuo. Habitualmente, la presión es más baja en la fase nocturna, aunque se pueden producir variaciones según el paciente clasificándose los pacientes como: dippers, si su presión nocturna desciende entre un 10 % y un 20 %; non dippers, si su presión arterial desciende menos de un 10 %; dipper extremo, si su presión nocturna desciende más del 20 %, y riser, si su presión nocturna es más elevada que la diurna o no les varía.

Con el fin de determinar el nivel de la presión arterial de una persona nos debemos basar en un promedio de dos a tres lecturas realizadas en al menos dos ocasiones diferentes. En base a ese promedio obtenido, según la American Heart Association, los diferentes patrones que nos podemos encontrar a la hora de realizar la medición de la tensión arterial, se pueden clasificar como se muestra en la Tabla 1 en relación a los valores obtenidos.

	Sistólica	Diastólica
Hipotensión	< 80 mmHg	< 60 mmHg
Normal	< 120 mmHg	< 80 mmHg
Elevada	120-129 mmHg	< 80 mmHg
HTA Etapa 1	130 - 139 mmHg	80 - 89 mmHg
HTA Etapa 2	\geq 140 mmHg	\geq 90 mmHg
Crisis Hipertensiva	\geq 180 mmHg	\geq 120 mmHg

Tabla 1: Clasificación de las tensiones arteriales - American Heart Association (AHA)

Saturación de oxígeno: La saturación de oxígeno consiste en la medida de la cantidad de oxígeno disponible en la sangre. Esta cantidad consta del oxígeno unido a la hemoglobina (que abarca entre el 97 % y el 98 % del contenido total de oxígeno) y el oxígeno disuelto en el plasma. Unos niveles de saturación óptimos garantizan que las células del cuerpo reciban la cantidad adecuada de oxígeno. En la Tabla 2 se muestran diferentes patrones asociados a la medición de esta constante vital junto con las actuaciones que se pueden llevar a cabo en relación al valor de saturación de oxígeno que posea un paciente [19].

% Saturación	Actuación
< 95 %	Paciente en buen estado, no hay que actuar.
95-90 %	Paciente con hipoxia. Tratamiento inmediato y monitorización de la respuesta al mismo.
< 90 %	Enfermo grave con hipoxia severa. Oxigenoterapia y traslado al hospital.
< 80 %	Enfermo crítico. Valorar intubación y ventilación mecánica.

Tabla 2: Actuación según % de Saturación

2.2. Sistemas actuales

En la actualidad existen diversas plataformas de bajo coste que permiten realizar mediciones de diferentes constantes vitales. Algunas de ellas son:

- **Bitalino:** Este sistema [1] está formado por un dispositivo que integra múltiples sensores de medición para la adquisición de datos bioeléctricos y biomecánicos. En él se puede realizar la conexión de diferentes sensores propios de cada usuario, no solamente los propios de Bitalino, puesto que tanto los bloques de control, como los bloques de alimentación y comunicación y el firmware son de propósito general.
- **MySignals:** Se corresponde con una plataforma desarrollada por Libelium [4] para dispositivos médicos y aplicaciones de web eHealth, Android o iOS que posee 15 sensores corporales diferentes que permiten realizar la medición de diferentes constantes vitales como el nivel de glucosa, el ritmo cardiaco o la presión sanguínea, entre otras. Además, posee dos aplicaciones diferentes, una para el sistema operativo Android o iOS, que nos ofrece la posibilidad de visualizar los datos a través de Bluetooth, y la aplicación Cloud, que nos permite ver los datos que han sido guardados anteriormente en una cuenta privada de usuario.

- **Pulsera de medición bajo coste:** Este dispositivo, creado por Venkata Virajit Garbhapu y Sundararaman Gopalan y presentado en el *7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2017)* [12] consiste en una pulsera que recopila la información relacionada con diferentes signos vitales del paciente. Una de las ventajas del diseño propuesto es que los signos vitales pueden ser monitoreados por expertos en cualquier lugar del mundo gracias a la red Wi-Fi.

2.3. Clasificadores de señales de ECG

En la actualidad, existen numerosas publicaciones que han tratado el tema de la clasificación automática de las señales del electrocardiograma, proponiendo diversos métodos para ello. A pesar de ello, existen diversas cuestiones que se deben abordar con el fin de desarrollar clasificadores más robustos y eficientes. Algunos de estos problemas se basan en la elección del enfoque de clasificación que se adoptará, la automatización de la selección de características y la forma en la que se seleccionan los mejores parámetros libres del clasificador, la cual generalmente se suele realizar empíricamente [17].

Uno de los métodos existentes, el cual ha sido definido en el artículo *ECG beat recognition using fuzzy hybrid neural network* [20] se basa en la aplicación de una red neuronal híbrida difusa para el reconocimiento y clasificación de ritmos cardíacos basados en el ECG. Esta red consiste en una subred difusa autoorganizada (utilizando algoritmos como el Gustafson-Kessel para su organización) la cual permanece conectada en cascada con el perceptrón multicapa, el cual es utilizado como el clasificador final.

Por otro lado, en *On-line heart beat recognition using Hermite polynomials and neuro-fuzzy network* [15], se realiza la creación de un enfoque neuro-difuso para el reconocimiento y la clasificación de los diferentes ritmos cardíacos en función de las formas de onda del ECG. Una de las partes más importantes de este clasificador tiene su fundamento en que la señal del complejo QRS se caracteriza por polinomios de Hermite, cuyos coeficientes sirven como las características del proceso, las cuales se aplican a una red neuronal difusa para su reconocimiento.

3. Sistema propuesto

Como se indicó anteriormente en la sección 1, se comenzó la elaboración de este sistema como parte del Trabajo de Fin de Grado del alumno Christian Bonal Martín [16], perteneciente a la Universidad de Salamanca.

Este sistema que se desarrolló anteriormente se divide en tres partes: un sistema web, desde el cual el profesional médico puede acceder a diversas funciones como por ejemplo la realización de un estudio; un aplicativo móvil, el cual ayuda a los conductores de las ambulancias a conocer todas las incidencias que se producen, así como a ahorrar tiempo a la hora de desplazarse a una de ellas, y

un conjunto de scripts que permiten recoger los datos obtenidos en las lecturas a partir de los sensores y su posterior envío al servidor.

Como mejora, en este proyecto se expondrá un motor de reglas que, a partir de los datos obtenidos por los sensores, determine un diagnóstico en base a los patrones anómalos antes estudiados en la sección 2.

3.1. Sistema Web

Esta parte del sistema se basa en un portal web que el profesional médico puede utilizar para la realización de todas las tareas relacionadas con una consulta a un paciente, como la realización de un estudio, así como la consulta de históricos de dicho paciente.

La interfaz de este portal se ha realizado utilizando los lenguajes HTML y CSS junto con el framework Bootstrap, ayudados por la utilización de la librería Highcharts, la cual ayuda en la creación de gráficas estáticas y de gráficas en tiempo real, el framework Firebase, el cual ayuda a realizar una comunicación entre cliente y servidor más sencilla, y la API de Google Maps, con el fin de crear nuevos mapas en la interfaz. Para dotar de funcionalidad a este sistema se ha utilizado el lenguaje PHP y el lenguaje JavaScript.

3.2. Sensores

Esta segunda parte del sistema consiste en una placa y un conjunto de sensores desarrollado por Cooking Hacks [2] que permite la realización de mediciones de diferentes constantes vitales como por ejemplo el pulso cardíaco o la tensión arterial. Los componentes principales del sistema de medición son:

- **Placas:** Forman el núcleo principal del sistema, al cual se conectan todos los sensores. Las placas de las que consta este núcleo son: Raspberry Pi 2 Model B, utilizada como base para la captación y envío de los datos que se reciben desde los sensores; Placa e-Health Sensor Shield V2.0, para la conexión de los diferentes sensores, y Placa Raspberry Pi to Arduino Shields Connection Bridge, que actúa como enlace entre la Raspberry Pi y la placa anterior, con el fin de poder conectar dicha placa, diseñada para Arduino, a la propia Raspberry Pi.
- **Pulsioxímetro:** Es un sensor que permite medir la cantidad de oxígeno disuelto en la sangre mediante el uso de dos longitudes de onda de luz diferentes, de 660 nm (luz roja) y de 940 nm (luz infrarroja) que permiten medir la diferencia real en los espectros de absorción de la hemoglobina oxigenada (HbO₂) y sin oxigenar (Hb).
- **Tensiómetro:** El tensiómetro es un sensor que permite al usuario controlar la presión que ejerce la sangre contra las arterias mientras el corazón bombea.
- **Electrocardiógrafo:** El electrocardiógrafo (ECG) es un sensor formado por tres cables a los que se les conectan electrodos, cada uno con una polaridad, que permite evaluar las funciones eléctricas y musculares del corazón.

3.3. Aplicativo móvil

La tercera parte desarrollada en este sistema consiste en una aplicación iOS mediante la cual los conductores de ambulancias pueden estar al tanto en todo momento de las incidencias que se produzcan. En caso de que se reconozca algún patrón anómalo grave en algún estudio realizado a algún paciente, los conductores de ambulancia recibirán una notificación de alerta en sus dispositivos móviles. Al atender dicha alerta, se indicará la ruta más rápida para llegar al lugar donde se encuentra el paciente con el fin de evitar retrasos que puedan ser perjudiciales para la persona que ha realizado el estudio.

Además, cada usuario de la aplicación podrá visualizar un histórico de todas aquellas incidencias que haya tratado.

3.4. Motor de reglas

Consiste en una aplicación Java que permite, a través de un conjunto de reglas definidas utilizando el framework Drools, realizar el diagnóstico de diferentes enfermedades en relación a los datos obtenidos en las mediciones y en base a las características de los diferentes patrones anómalos que se han estudiado en la sección 2.

Pulsioxímetro: Para el caso del sensor pulsioxímetro, las reglas que se han generado intentan concluir el diagnóstico del paciente en lo referente a los valores de saturación de oxígeno que se obtienen de dicho sensor. Los patrones que se pueden detectar gracias a ellas son los relacionados con las enfermedades de hipoxia, hipoxia severa y crítica y un patrón correcto. Un ejemplo de regla para este sensor se muestra en la Figura 1.

```
rule "Hipoxia rule"
  salience 20
  when
    $pulsioxi : Pulsioximeter ((Integer.valueOf(SP02) < 95 && Integer.valueOf(SP02) >= 90));
    $spo2 : String() from $pulsioxi.getSP02();
  then
    System.out.println("Checking action for Pulsioximeter");
    System.out.println("Hipoxia detected in Pulsioximeter");
    $currentMeasure.setDiagnostic("Hipoxia");
    System.out.println("SP02: " + $spo2 + " - Diagnostic: " + $currentMeasure.getDiagnostic());
  end
```

Figura 1: Ejemplo de regla para el pulsioxímetro

Electrocardiógrafo: Las reglas correspondientes con la interpretación del electrocardiograma (ECG) obtenido a partir del sensor tratan de determinar diversos patrones en la forma y duración de la onda, así como diferentes características del pulso obtenido con el fin de decidir si el paciente posee alguna anomalía o no. Los patrones que se detectan a partir de estas reglas son los especificados en la sección 2 para un ritmo sinusal, una taquicardia sinusal y auricular, un aleteo auricular y una bradicardia sinusal.

Tensiómetro: Finalmente, el último grupo de reglas se centran en detectar patrones anómalos en la medición de la tensión arterial a través del tensiómetro. Los patrones que se pueden detectar, además de un patrón correcto, se centran en las enfermedades de tensión alta, hipertensión fase 1, hipertensión fase 2 y crisis hipertensiva.

4. Caso de estudio

Una vez creado el sistema, se realizaron diversos casos de prueba a 10 pacientes diferentes, con edades comprendidas entre los 18 años y los 58 años para determinar posibles variaciones antes de implantarlo como sistema para pacientes crónicos. Las características de cada paciente se recogen en la Tabla ??.

Identificador	Edad	Género	Estatura	Peso
S0001	26	Varón	175 cm	83 kg
S0002	20	Mujer	168 cm	60 kg
S0003	18	Varón	184 cm	90 kg
S0004	38	Mujer	170 cm	64 kg
S0005	58	Varón	173 cm	86 kg
S0006	49	Varón	168 cm	70 kg
S0007	23	Mujer	160 cm	55 kg
S0008	29	Mujer	165 cm	58 kg
S0009	46	Mujer	170 cm	61 kg
S0010	33	Varón	180 cm	84 kg

Tabla 3: Características de los pacientes de prueba

5. Resultados obtenidos

En esta sección se presentarán los resultados que se han obtenido una vez realizadas las pruebas para los diez pacientes antes descritos. Cabe destacar que a la hora de tomar las medidas los pacientes se encontraban en estado de reposo. Además, se han realizado tres mediciones para cada paciente, obteniendo la media de todos los resultados como resultado final con el fin de evitar posibles errores de los sensores que afectasen al resultado final.

Además, debido a la dificultad de encontrar pacientes que posean algún tipo de enfermedad relacionada con la medición de las constantes vitales en este sistema, se generaron alrededor de 2500 estudios aleatorios para poder probar su correcto funcionamiento para todos los casos en cada sensor. Para el caso de la generación de gráficos de electrocardiograma, se ha utilizado la herramienta ECGSYN [3], la cual permite sintetizar señales de diversos electrocardiogramas que ayuden a analizar los diferentes patrones de cada enfermedad. Posteriormente, a partir de los diagnósticos iniciales obtenidos gracias a las reglas, se realizó un proceso de minería de datos con el que determinar si, utilizando alguno de los algoritmos descritos en la sección 2, se realizan las clasificaciones de las enfermedades para los diferentes sensores de una manera correcta. Para ello se ha utilizado la herramienta Weka [5], aplicando el algoritmo supervisado de clasificación J48 para todos los casos, el cual se trata de una implementación del algoritmo C4.5 antes descrito.

5.1. Pulsioxímetro

En la Tabla 4 se reflejan los datos obtenidos para el sensor pulsioxímetro para los 10 pacientes de prueba. Como se puede comprobar, el prototipo realizado ha logrado clasificar de manera correcta a todos los pacientes en relación a los valores de pulso y de saturación de oxígeno. Además, cabe destacar que ningún paciente posee alguna enfermedad.

Identificador	Valor Pulso	Valor SPO ₂	Diagnóstico
S0001	72	99	Healthy
S0002	79	98	Healthy
S0003	70	97	Healthy
S0004	81	97	Healthy
S0005	74	98	Healthy
S0006	71	97	Healthy
S0007	79	99	Healthy
S0008	82	98	Healthy
S0009	77	99	Healthy
S0010	72	97	Healthy

Tabla 4: Resultados pulsioxímetro

En cuanto a los resultados obtenidos en el proceso de minería de datos para el sensor pulsioxímetro (Figura 2), se puede observar que se ha realizado la clasificación de 2500 instancias, en la que se ha obtenido un 99,92 % de acierto, lo que supone que la mayoría de las instancias se han clasificado correctamente, sin errores. Además, cabe destacar también el valor del índice Kappa, un medidor que se usa para generar la curva ROC y que nos ayuda a comprender la calidad de la clasificación, el cual se sitúa en 0.9991, muy cercano a 1 que es el valor perfecto, cuando existe concordancia total.

```

=== Summary ===

Correctly Classified Instances      2498          99.92 %
Incorrectly Classified Instances     2            0.08 %
Kappa statistic                    0.9991
Mean absolute error                 0.0002
Root mean squared error             0.0141
Relative absolute error              0.0929 %
Root relative squared error         4.3099 %
Total Number of Instances          2500

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
1,000  0,001  0,998  1,000  0,999  0,999  1,000  0,998  Healthy
0,996  0,000  1,000  0,996  0,998  0,998  0,998  0,996  SinusTachycardia
1,000  0,000  0,996  1,000  0,998  0,998  1,000  0,996  AtrialTachycardia
1,000  0,000  1,000  1,000  1,000  1,000  1,000  1,000  AtrialFlutter
0,996  0,000  1,000  0,996  0,998  0,998  0,998  0,996  SinusBradycardia
1,000  0,000  1,000  1,000  1,000  1,000  1,000  1,000  Hipoxia
1,000  0,000  1,000  1,000  1,000  1,000  1,000  1,000  SevereHipoxia
1,000  0,000  1,000  1,000  1,000  1,000  1,000  1,000  Critical
Weighted Avg.  0,999  0,000  0,999  0,999  0,999  1,000  0,998

=== Confusion Matrix ===

 a  b  c  d  e  f  g  h  <-- classified as
572  0  0  0  0  0  0  0  | a = Healthy
  0 229 1  0  0  0  0  0  | b = SinusTachycardia
  0  0 272 0  0  0  0  0  | c = AtrialTachycardia
  0  0  0 262 0  0  0  0  | d = AtrialFlutter
  1  0  0  0 237 0  0  0  | e = SinusBradycardia
  0  0  0  0  0 280 0  0  | f = Hipoxia
  0  0  0  0  0  0 328 0  | g = SevereHipoxia
  0  0  0  0  0  0  0 318  | h = Critical

```

Figura 2: Resultados: Pulsioxímetro

Analizando la matriz de confusión, podemos concluir que solamente se han dado dos casos en los que no se ha realizado la clasificación correctamente. En el primer caso, se detectó como una taquicardia auricular una taquicardia sinusal, mientras que en el segundo caso se detectó como paciente sano un paciente con una bradicardia sinusal. En el resto de casos, se han clasificado correctamente los diagnósticos.

5.2. Tensiómetro

Para el caso del tensiómetro, los valores obtenidos para los 10 pacientes de prueba se muestran en la Tabla 5. A raíz de los resultados obtenidos, se puede determinar que se ha realizado un diagnóstico correcto en relación a los patrones

anteriormente estudiados. En este caso, sólo se han podido obtener muestras de tensión normal y tensión elevada.

Identificador	Sistólica	Diastólica	Diagnóstico
S0001	110	68	Healthy
S0002	120	64	Healthy
S0003	120	71	Healthy
S0004	120	68	Healthy
S0005	128	72	Elevated
S0006	104	63	Healthy
S0007	106	72	Healthy
S0008	117	63	Healthy
S0009	125	78	Elevated
S0010	106	60	Healthy

Tabla 5: Resultados tensiómetro

En cuanto a los resultados obtenidos en el proceso de minería de datos para el sensor tensiómetro, mostrados en la Figura 3, se puede comprobar que se ha realizado la clasificación de 2432 instancias de manera correcta en su totalidad, lo que supone un 100% de acierto. Además, el valor obtenido para el índice Kappa concuerda con el valor del acierto, puesto que 1 significa concordancia total.

```

=== Summary ===
Correctly Classified Instances 2432 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0
Root mean squared error 0
Relative absolute error 0 %
Root relative squared error 0 %
Total Number of Instances 2432

=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 Healthy
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 Elevated
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 HTA1
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 HTA2
Weighted Avg. 1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 HipertensiveCrisis

=== Confusion Matrix ===
 a b c d e <-- classified as
527 0 0 0 0 | a = Healthy
0 470 0 0 0 | b = Elevated
0 0 463 0 0 | c = HTA1
0 0 0 452 0 | d = HTA2
0 0 0 0 510 | e = HipertensiveCrisis

```

Figura 3: Resultados: Tensiómetro

Si atendemos al resto de indicadores, se puede comprobar que todos ofrecen el mayor valor posible, que es uno. En el caso de la matriz de confusión, se pue-

den observar que todas las instancias se han clasificado en la clase correcta, sin errores.

5.3. ECG

Para el sensor ECG, los datos obtenidos para los 10 pacientes de prueba se reflejan en la Tabla 6 y la Tabla 7. En este caso se examinan los diferentes parámetros importantes que forman la onda del electrocardiograma para determinar si el paciente posee un ritmo normal.

Identif	Ptime	Pamplitud	Qtime	Qamplitud	Rtime	Ramplitud
S0001	0.03714	0.24583	0.02147	-0.24833	0.0172	2.21284
S0002	0.05705	0.15706	0.02177	-0.2895	0.01146	2.42337
S0003	0.06858	0.1491	0.01939	-0.16258	0.02576	2.36003
S0004	0.09601	0.11663	0.01869	-0.26649	0.01302	2.11937
S0005	0.03598	0.11962	0.02716	-0.20213	0.01111	2.12475
S0006	0.01484	0.19828	0.01523	-0.20946	0.01073	2.19507
S0007	0.05124	0.13179	0.01438	-0.26533	0.01959	2.29686
S0008	0.06054	0.20703	0.01964	-0.11461	0.00415	2.37246
S0009	0.00906	0.1794	0.01595	-0.24046	0.01746	2.27406
S0010	0.05235	0.15584	0.02642	-0.25646	0.02453	2.36248

Tabla 6: Resultados ECG I

Identif	Tamplitud	PRTime	QRSTime	Diagnóstico
S0001	0.4574	0.12016	0.07672	Healthy
S0002	0.46919	0.18451	0.09152	Healthy
S0003	0.47225	0.14078	0.0751	Healthy
S0004	0.42789	0.18029	0.08625	Healthy
S0005	0.40839	0.18448	0.08226	Healthy
S0006	0.41832	0.19176	0.08842	Healthy
S0007	0.48525	0.13691	0.06991	Healthy
S0008	0.48045	0.19477	0.08273	Healthy
S0009	0.44284	0.15446	0.08775	Healthy
S0010	0.42194	0.19061	0.09193	Healthy

Tabla 7: Resultados ECG II

Al igual que para los anteriores sensores, el prototipo desarrollado ha realizado un diagnóstico correcto, puesto que para todas las muestras recogidas no se ha detectado ninguna anomalía, determinando que el paciente está sano.

En cuanto al proceso de minería de datos, los resultados que se han obtenido para el sensor ECG se muestran en la Figura 4. En ellos se puede observar que se ha realizado la clasificación de 2500 instancias, en la que se ha obtenido un 99,76 % de acierto, lo que supone que la mayoría de las instancias se han clasificado correctamente, sin errores (2494 de 2500). En cuanto al valor del índice Kappa, como en los anteriores casos, es muy elevado, siendo 0.9972, muy cercano a la concordancia total.

```

=== Summary ===
Correctly Classified Instances      2494      99.76 %
Incorrectly Classified Instances     6         0.24 %
Kappa statistic                    0.9972
Mean absolute error                 0.0007
Root mean squared error             0.0262
Relative absolute error             0.2801 %
Root relative squared error         7.4841 %
Total Number of Instances          2500

=== Detailed Accuracy By Class ===

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,997	0,001	0,992	0,997	0,995	0,994	0,998	0,990	Healthy
	1,000	0,000	0,997	1,000	0,999	0,998	1,000	0,997	PwaveProblem
	0,991	0,000	1,000	0,991	0,996	0,995	0,996	0,993	PRTIMEProblem
	1,000	0,000	0,997	1,000	0,999	0,998	1,000	0,997	QRSTIMEProblem
	1,000	0,000	0,997	1,000	0,998	0,998	1,000	0,997	QwaveProblem
	0,994	0,000	1,000	0,994	0,997	0,997	0,997	0,995	RwaveProblem
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	TwaveProblem
Weighted Avg.	0,998	0,000	0,998	0,998	0,998	0,997	0,999	0,996	

```

=== Confusion Matrix ===

```

	a	b	c	d	e	f	g	
367	1	0	0	0	0	0	0	a = Healthy
0	370	0	0	0	0	0	0	b = PwaveProblem
1	0	348	1	1	0	0	0	c = PRTIMEProblem
0	0	0	366	0	0	0	0	d = QRSTIMEProblem
0	0	0	0	331	0	0	0	e = QwaveProblem
2	0	0	0	0	353	0	0	f = RwaveProblem
0	0	0	0	0	0	359	0	g = TwaveProblem

Figura 4: Resultados: ECG

Con el fin de conocer los diagnósticos que se han clasificado incorrectamente, tomamos como referencia la matriz de confusión. En ella se puede comprobar que solamente se han dado seis casos en los que no se ha realizado la clasificación correctamente. En el primer caso, se detectó que el paciente tenía un problema en la onda P, pero en realidad estaba sano. En el segundo caso, hubo dos pacientes que se clasificaron como sanos que en realidad tenían problemas en la onda R. Finalmente, existen tres casos en los que pacientes que tenían un problema en el intervalo PR se catalogaron como otra clase. (sanos, error en el segmento QRS y error en la onda Q). En el resto de casos, el algoritmo clasificador ha incluido en la clase correcta los diagnósticos.

6. Conclusiones y líneas de trabajo futuras

Se ha realizado una investigación acerca de diferentes patrones anómalos relacionados con la medida de algunas constantes vitales como el pulso o la saturación de oxígeno, así como sobre diferentes sistemas propuestos en la actualidad y diversos clasificadores para realizar técnicas de minería de datos, tal y como se indicó en los objetivos, con el fin de desarrollar una mejora del Trabajo de Fin de Grado del alumno Christian Bonal Martín, de la Universidad de Salamanca mediante la creación de un motor de reglas que realice diagnósticos en base a los patrones anómalos antes estudiados.

En el caso de los resultados obtenidos para el pulsioxímetro, se puede comprobar que todos los sujetos poseen un valor de saturación de oxígeno situado entre el 97% y el 99%, lo cual significa que no tienen ninguna anomalía. En el caso del pulso, se puede comprobar que las mujeres poseen un valor de pulsaciones por minuto mayor que los hombres, debido principalmente al tamaño del corazón, que es más pequeño en ellas. En cuanto a los resultados obtenidos para el tensiómetro, se puede observar que la mayoría de los pacientes tienen una tensión dentro de los valores en los que se considera normal. En el caso del sujeto número 5 y del sujeto número 9, la tensión es elevada, pero ninguno de ellos llega al problema de la hipertensión. Finalmente, observando los resultados obtenidos para el electrocardiograma, se puede determinar que ninguno de los pacientes posee alguna anomalía. Esta representación está ligada al valor del pulso arterial, por lo que la correspondencia entre ambos es correcta, ya que se indica en ambos resultados que el paciente está sano para todos los casos.

Teniendo en cuenta el global de los resultados, se puede concluir que tiene una gran dificultad conseguir mediciones de personas enfermas, por lo que para probar por completo dicho motor de reglas resulta necesario simular estudios. Para el caso de la simulación de las señales de electrocardiograma, las cuales son más complejas de simular, se ha utilizado la herramienta ECGSYN, que nos permite generar gráficos de electrocardiograma para diferentes patrones. A pesar de esta dificultad, a la vista de los resultados se puede concluir que el sistema realiza los diagnósticos de manera correcta, clasificando las enfermedades de los pacientes de correctamente.

En cuanto a las líneas de trabajo futuras que se pueden realizar a partir de dicho proyecto, destaca una ampliación del número de sensores que se puedan utilizar para así obtener más datos que nos ayuden a tener una mejor perspectiva acerca de la salud de los pacientes. Otra línea se basa en la inclusión de un servicio de mensajería instantánea para mejorar la comunicación con el profesional médico. Otra opción futura para mejorar la funcionalidad del motor de reglas se basaría en la creación de nuevas reglas que permitan conocer si un paciente tiene una situación anómala en relación a posibles enfermedades que posea y que puedan hacer que sus valores de constantes vitales varíen. Por otro lado, la creación de servicios de gestión y reportes de las incidencias con el objetivo de tener un mejor control de ellas y la creación de un sistema de localización del paciente con el objetivo de que un paciente pudiese realizar sus estudios en el lugar en el que se encuentre son otras dos posibles líneas futuras.

Referencias

1. Bitalino - biomedical equipment | low cost toolkit. <http://bitalino.com/en/>. Accessed: 2018-05-31.
2. e-health sensor platform v2.0 for arduino and raspberry pi. <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical>. Accessed: 2018-06-04.
3. Ecgsyn: A realistic ecg waveform generator. <https://www.physionet.org/physiotools/ecgsyn/>. Accessed: 2018-06-09.
4. Mysignals - ehealth and medical iot. <http://www.my-signals.com/>. Accessed: 2018-05-31.
5. Weka 3: Data mining software in java. <https://www.cs.waikato.ac.nz/ml/weka/>. Accessed: 2018-06-09.
6. Niebauer J. Ashley EA. Chapter 3 - conquering the ecg. In *Cardiology Explained*. Remedica, 2004.
7. Instituto Nacional de Estadística. Proyecciones de población 2016 - 2066. *Notas de prensa INE*, 2016.
8. Instituto Nacional de Estadística. *España en cifras 2017*. INE, 2017.
9. Instituto Nacional de Estadística. *La Península Ibérica en cifras*. INE, 2017.
10. Molly Ware Eric H. Awtry, Cathy Jeon. Chapter 21 - tachyarrhythmias. In Molly G. Ware Eric Awtry, Cathy Jeon, editor, *Blueprints Cardiology*. Second edition, 2006.
11. Cosio Francisco G. Atrial flutter, typical and atypical: A review. *Arrhythmia and Electrophysiology Review*, pages 55–62, 2017.
12. Venkata Virajit Garbhapu and Sundararaman Gopalan. Iot based low cost single sensor node remote health monitoring system. *Procedia Computer Science*, 113:408 – 415, 2017. The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017) / Affiliated Workshops.
13. Ary L. Goldberger, Zachary D. Goldberger, and Alexei Shvilkin. *Goldberger's Clinical Electrocardiography (Ninth Edition)*. Elsevier, ninth edition, 2018.
14. Lip GY. Kakar P, Boos CJ. Management of atrial fibrillation. *Vascular Health and Risk Management*, pages 109–116, 2007.
15. T. H. Linh, S. Osowski, and M. Stodolski. On-line heart beat recognition using hermite polynomials and neuro-fuzzy network. *IEEE Transactions on Instrumentation and Measurement*, 52(4):1224–1231, Aug 2003.
16. Christian Bonal Martín. Sistema de teleasistencia para pacientes crónicos. Master's thesis, Universidad de Salamanca.
17. F. Melgani and Y. Bazi. Classification of electrocardiogram signals with support vector machines and particle swarm optimization. *IEEE Transactions on Information Technology in Biomedicine*, 12(5):667–677, Sept 2008.
18. Rothbaum DA. Noble RJ, Hillis JS. Chapter 33 - electrocardiography. In Hurst JW Walker HK, Hall WD, editor, *Clinical Methods: The History, Physical, and Laboratory Examinations*. Third edition, 1990.
19. Seco González A Noguero Casado MJ. Técnicas en ap: Pulsioximetría. *Fisterra*, 2004.
20. S. Osowski and Tran Hoai Linh. Ecg beat recognition using fuzzy hybrid neural network. *IEEE Transactions on Biomedical Engineering*, 48(11):1265–1271, Nov 2001.

Sistema para la Minería de Opiniones

Daniela Elizabeth Sanjinés Flores, Vivian F. López Batista, Ana B. Gil-González

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{idu001885, vivian, abg}@usal.es
<http://diaweb.usal.es/>

Resumen Las empresas comerciales y de servicios entre otros sectores, requieren que la gran cantidad de opiniones que se generan en la web por los usuarios pueda ser extraída de forma automática, obteniendo la polaridad de la opinión de sus clientes sobre sus productos o servicios para alcanzar sus objetivos. Sin embargo, debido a que las opiniones de la web son subjetivas y no estructuradas, aún existen problemas no solucionados dentro de la minería de opiniones, como son los problemas de ambigüedad y soporte de idiomas, que afectan a la hora de realizar la correcta clasificación de las opiniones. Se propone un sistema de minería de opiniones que determine la polaridad positiva y negativa para el idioma español, tratando de detectar la ironía como problema de ambigüedad. Se diseñó un modelo con dos enfoques, el primero utilizando la herramienta *Natural Language Toolkit* (NLTK) para el preprocesamiento de datos y utilizando unigramas y bigramas como características. El segundo enfoque utilizando la herramienta *TreeTagger* para el preprocesamiento de datos y aplicando el resultado lema/categoría gramatical como características. Se implementó el método de aprendizaje supervisado con el algoritmo Naive Bayes para su múltiple clasificación. La evaluación de los resultados con las métricas de *accuracy*, *recall*, *presicion* y *F-1 score* con puntuaciones por debajo de los 0.70. La importancia de los resultados demuestran que la detección de ironía en Español utilizando las técnicas clásicas de la minería de opiniones no son tan eficientes, aunque se deben mejorar dos problemas fundamentales como son la cantidad de datos, el idioma y la calidad del etiquetado.

Keywords: Minería de Opiniones, Análisis de Sentimiento

1. Introducción

Con el surgimiento de la web 2.0 y su rápido crecimiento, los usuarios pasaron de ser usuarios pasivos a formar parte activa y colaborativa en la web, generando diferentes tipos de contenido, uno de ellos es el contenido de forma textual. Las opiniones de los usuarios son generadas diariamente en grandes cantidades y de distintas formas, al calificar un producto comprado, al calificar algún servicio, al expresar la opinión sobre un tema a través de las redes sociales, realizando

comentarios en foros, criticando temas sobre noticias actuales, expresando opiniones en blogs, entre otros. Toda esta información generada es muy atractiva y de bastante interés, principalmente para las empresas, el sector gubernamental, también para investigadores y comunicadores, y además para los mismos usuarios. Debido a que esta información se encuentra en gran cantidad, fue necesario buscar técnicas y mecanismos para la recuperación, procesamiento y análisis de dicha información. Es aquí donde la Inteligencia Artificial interviene dando una respuesta a esta necesidad a través del Aprendizaje Automático o *Machine Learning* y el Procesamiento del Lenguaje Natural (PLN) que son ampliamente utilizados en lo que se denomina Minería de Opiniones.

La Minería de Opiniones o también llamado Análisis de Sentimiento, permite el procesamiento y la extracción automática de información clasificándola como opinión positiva, negativa o neutral, considerando que la información en muchas ocasiones se encuentra de forma no estructurada y es subjetiva debido a que maneja un lenguaje natural. Por estas razones la minería de opiniones cobró mayor importancia permitiendo analizar una gran cantidad de datos y obteniendo información que es muy útil para las empresas, ya que les permite saber si los usuarios están conformes con un determinado producto o servicio y esto ayuda a mejorar sus productos o a plantear estrategias para retener a clientes.

A pesar de que en los últimos cinco años se realizaron bastantes trabajos en este campo, la mayoría orientado al idioma inglés, y a que actualmente existen varias herramientas comerciales para determinar el análisis de sentimiento, lo cierto es que queda mucho camino por recorrer ya que varios obstáculos prevalecen a la hora de determinar la polaridad correcta de una opinión. Entre los retos pendientes se encuentran problemas de ambigüedad, la variabilidad lingüística, la falsedad, etc., todos estos afectan la clasificación del sentimiento de la opinión, dando en ocasiones una clasificación negativa siendo positiva o viceversa.

Se propone elaborar un sistema de minería de opiniones, que trate de resolver problemas de ambigüedad como la detección de ironía y además que esté dirigido el idioma español utilizando datos de Twitter. Para alcanzar lo propuesto se realiza la revisión literaria con el fin de conocer el estado del arte, se estudian las técnicas utilizadas con mejores resultados y en base a ello se propone un modelo aplicado al idioma español para la minería de opiniones tratando de resolver el problema de detección de ironía.

La estructura del trabajo se encuentra de la siguiente forma: En la sección 2 tenemos el estado del arte de la minería de opiniones, el Procesamiento de Lenguaje Natural y el Aprendizaje Automático. En el punto 3 se describe la metodología propuesta y el trabajo experimental. En la sección 4 se exponen los resultados y la discusión, y finalmente en el punto 5 la conclusiones del trabajo realizado y líneas de trabajo futuras.

2. Estado del Arte

La minería de opiniones es una rama del Procesamiento del Lenguaje Natural (PLN), el que a su vez viene del campo de la Inteligencia Artificial. Por

otra parte la minería de opiniones es también clasificada como subconjunto de la minería de contenido web y ésta a su vez pertenece a la minería de datos. Los métodos principales que se han utilizado en los trabajos de minería de opiniones corresponden al Aprendizaje Supervisado y en base a Lexicón¹[4]. Estos trabajos han utilizado textos cortos y textos largos como datos de entrada, entre las tendencias actuales de la minería de opiniones, se destacan el uso de datos cortos que corresponden a datos de Twitter[8], esto principalmente debido a su facilidad de obtención. Los métodos más utilizados para su extracción son a través de *hashtags* '#' como forma supervisada y etiquetado automático, y de forma manual con anotadores² también es utilizado un método semi-supervisado el cual requiere menos cantidad de datos etiquetados y los no etiquetados van aprendiendo poco a poco[5].

Sobre las técnicas de preprocesamiento de datos que corresponden al PLN para la minería de opiniones, se destacan la *tokenizacion* y el análisis *POS Tagging*. Como es sabido la mayoría de los trabajos utiliza datos en idioma inglés por lo cual entre las herramientas más utilizadas está NLTK de Python, también se menciona *OpenNLP*³ y las herramientas de *Stanford NLP*⁴[9], sin embargo se investigaron herramientas más adaptables para el idioma español resultado de esto fue *TreeTagger*⁵. En cuanto a la extracción de características, es muy utilizado el método de bolsa de palabras a través de la frecuencia de las características, con unigramas y bigramas[5] de acuerdo a los objetivos de la investigación, algunos autores tratan de analizar de forma general la polaridad, por documento otros de forma de frase, de forma de grano y conceptual. Las técnicas de clasificación más utilizadas en minería de opiniones y las que obtienen mejores resultados en general son Naive Bayes y SVM. Estos debido a su facilidad de entendimiento y su eficiencia. Por otra parte las técnicas en base a Redes Neuronales están surgiendo. Las métricas de clasificación varían, pero por lo general se mide el *accuracy*, la *presision*, el *recall*, y *F1 score*[4].

3. Metodología y Experimentación

Se presenta un modelo que permitirá la múltiple clasificación de las opiniones de forma automática en el idioma español, a través de técnicas de Inteligencia Artificial tratando de detectar la ironía como problema de ambigüedad. La metodología aplicada para el presente trabajo se resume en la figura 1 donde se sigue una serie de pasos. Para la experimentación se implementará la metodología de aprendizaje supervisado con dos enfoques, el primer enfoque realizando las tareas de preprocesamiento con NLTK y construyendo una matriz con unigramas

¹ Listado de palabras donde cada una tiene un valor numérico que corresponde al grado de sentimiento ya sea positivo o negativo.

² Personas que leen y clasifican de forma manual el texto según su criterio

³ <https://opennlp.apache.org/docs/1.5.3/manual/opennlp.html>

⁴ <https://nlp.stanford.edu/software/tokenizer.html>

⁵ <http://treetaggerwrapper.readthedocs.io/en/latest>

y bigramas y el segundo realizando el preprocesamiento con *TreeTagger* y construyendo la matriz con unigramas, donde el unigramas corresponde al lema/pos de la palabra que se extrae previamente. A continuación se detalla el trabajo experimental:

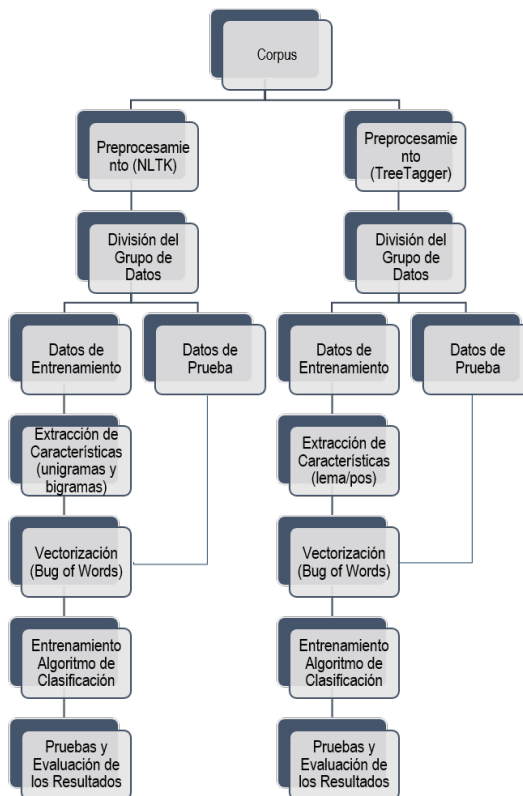


Figura 1: Metodología propuesta. Fuente: Elaboración propia

3.1. Corpus

Los datos utilizados para la experimentación fueron obtenidos de Twitter. La extracción de tuits se realizó de forma supervisada y por grupos para obtener un etiquetado automático. Se realizó la extracción de tres grupos de tuits: positivos, negativos e irónicos a través de Python 3 utilizando la biblioteca *Tweepy*, la cual permite la conexión al *API REST* de Twitter, se utilizó el método *search* que busca los datos recuperando tuits de los últimos 7 días. Al momento de obtener los tuits de cada grupo, se realizó una limpieza eliminando ciertos caracteres y se excluyeron algunos tuits:

1. **Exclusión de Retuits:** Se excluyen los textos que corresponden a retuits ya que genera duplicidad, esto es identificado con 'RT @' dentro del texto.
2. **Exclusión de Tuits con URL:** La mayoría de tuits que contienen URL en el texto corresponden a propagandas. Por este motivo se decidió excluir los tuits que contienen 'https://' en su texto.
3. **Eliminación de hashtag '#':** En muchos casos los usuarios incluyen el hashtag seguido de una palabra como parte de la frase que comparten, de esta forma se vio conveniente no eliminar toda la palabra sino únicamente el símbolo hashtag '#' del texto.
4. **Eliminación de emoticones:** Con el fin de evitar un sobre ajuste se eliminan los emoticones, esto se realizó con una biblioteca llamada *preprocessor*⁶.
5. **Eliminación de espacios:** Se eliminaron todos los espacios innecesarios antes de guardar los tuits.
6. **Eliminación de menciones:** Se eliminaron las menciones a usuarios utilizando la biblioteca llamada *preprocessor*.
7. **Eliminación de tildes:** Ya que el idioma español presenta tildes y estos causan dificultad en el proceso, se utilizó una función que elimina las tildes de los tuits.

Los tuits de los diferentes grupos fueron extraídos de la siguiente forma:

- **Tuits Positivos:** Se realizó la extracción de 15020 tuits positivos. El criterio de búsqueda se realizó con tuits que contengan el emoticón carita feliz ':D' en su texto, asumiendo que estos tuits corresponden a sentimientos positivos. Además al momento de obtener los tuits se etiqueto como 'positivo' y se aplicaron los criterios de exclusión y limpieza de tuits antes mencionados.
- **Tuits Negativos:** Se realizó la extracción de 10064 tuits negativos. El criterio de búsqueda se realizó en base a tuits que contengan en su texto el emoticón de carita triste ':(', asumiendo que estos tuits corresponden a sentimientos negativos. Además al momento de obtener los tuits se etiqueto como 'negativo' y se aplicaron los criterios de exclusión y limpieza de tuits antes mencionados.
- **Tuits Irónicos:** Se realizó la extracción de 1125 tuits irónicos, ejecutando el script en 3 ocasiones. El criterio de búsqueda se realizó con el #sarcasmo, #ironía asumiendo que los tuits son irónicos. Además estos tuits se etiquetaron como 'ironia' y se aplicaron los criterios de exclusión y limpieza de tuits antes mencionados.

Debido a que el número de tuits irónicos era reducido (1125 tuits), se tomó la misma cantidad para el grupo de positivos y de negativos, los tres grupos de tuits fueron unidos en un solo archivo tuits.csv formando el corpus para el trabajo experimental con 3375 tuits.

3.2. Preprocesamiento

El preprocesamiento o también llamado normalización del texto es el segundo paso del modelo propuesto. El primer enfoque realiza el preprocesamiento

⁶ <https://pypi.org/project/tweet-preprocessor/>, elimina menciones, emoticones.

con bibliotecas de NLTK, y el segundo enfoque utiliza *TreeTagger*. Se explica a continuación el tratamiento realizado para cada enfoque:

Enfoque 1 - Preprocesamiento con NLTK: Las tareas realizadas en la etapa del preprocesamiento con NLTK son resumidas en figura 2.



Figura 2: Preprocesamiento con NLTK. Fuente:Elaboración propia

Se trabajaron los datos en estructura en memoria con *Pandas*⁷. Primeramente se añadió una columna llamada 'Etiqueta-Num' en la cual se introdujo una etiqueta numérica equivalente a la etiqueta obtenida en la recolección de tuits para los tres grupos, donde positivo =0, negativo=1 e ironico=2. Luego se procesaron los tuits, eliminando las palabras vacías para el español con un método de NLTK *stopwords.words('spanish')* y aplicando *SnowballStemmer('spanish')* para obtener la raíz de las palabras de los tuits.

El resultado del preprocesamiento con NLTK fue almacenado, creando una columna en el *DataFrame* llamado 'Preprocesado' como se tiene en tabla 1.

Etiqueta	Tuit	Etiqueta-Num	Preprocesado
positivo	'Suena bien! La vida es muy corta para no celebrar el amor'	0	'suen bien ! la vid cort celebr amor'
negativo	'Estoy cansadiisima de todo ya x favor'	1	'estoy cansadiisim x favor'
ironico	'Gracias a todos por respetar a la gente con pareja'	2	'graci respet gent parej'

Tabla 1: Preprocesado con NLTK

Enfoque 2 - Preprocesamiento con TreeTagger: Las tareas realizadas en la etapa del preprocesamiento con TreeTagger son resumidas en figura 3.

⁷ Biblioteca de Python que permite el fácil uso de estructuras de datos a través de su *DataFrame* el cual permite almacenar y manipular filas y columnas.

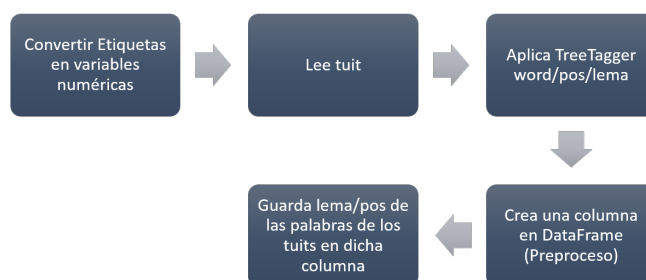


Figura 3: Preprocesamiento con TreeTagger. Fuente: Elaboración propia

Del mismo modo, se agregó una columna al *DataFrame*, en la cual se introdujo una etiqueta numérica equivalente a la etiqueta textual obtenida en la recolección de tuits para los tres grupos. Donde positivo =0, negativo=1 e irónico=2. El enfoque se realizó leyendo uno a uno los tuits, estos fueron procesados con *TreeTagger* a través de Python, el cual *tokeniza* el tuit y devuelve la palabra, el lema y el *POS Tagging* de cada palabra correspondiente al tuit. Para poder utilizar como características los lemas relacionados al pos de cada *token* del tuit, se creó una columna en la estructura de *DataFrame* llamada 'Preproceso' en la cual, se guardó únicamente el lema y *POS Tagging* (lema/pos) de cada palabra (ver tabla 2 que representa al *DataFrame*).

Etiqueta	Tuit	Etiqueta-Num	Preprocesado
positivo	'Suena bien! La vida es muy corta para no celebrar el amor'	0	'Sonar\VLfin bien\ADV !\FS el\ART vida\NC ser\VSF muy\ADV corto\ADJ para\CSUBI no\NEG celebrar\VLinf el\ART amor\NC'
negativo	'Estoy cansadiisima de todo ya x favor'	1	'Estar\VEfin cansadiisima\ADJ de\PREP todo\QU ya\ADV x\CARD favor\NC'
ironico	'Gracias a todos por respetar a la gente con pareja'	2	'Gracia\NC a\PREP todo\QU por\PREP respetar\VLinf a\PREP el\ART gente\NC con\PREP pareja\NC'

Tabla 2: Preprocesado con TreeTagger

3.3. División de conjunto de datos

Se realizó la división de los datos del mismo modo para ambos enfoques. El corpus preprocesado contiene un total de 3375 tuits de los cuales se dividió el 75 % para el entrenamiento (2531 tuits) y el 25 % para pruebas (844 tuits). La división del conjunto de datos se realizó con el método *train_test_split* que pertenece a la biblioteca *scikit-learn* de Python.

3.4. Extracción de Características

La extracción de características se realizó a partir de los datos de la columna 'Preprocesado' del *DataFrame* de cada uno de los enfoques. Para la extracción de características se utilizó *CountVectorizer*, en el cual se definió que se creara un vocabulario a partir de unigramas y en otro caso con bigramas para el primer enfoque. Para el segundo enfoque se definió el lema/pos obtenido del preprocesamiento. En el primer enfoque, se ignoran las palabras más frecuentes del 50% del corpus y las menos frecuentes debajo del 2%, para el segundo enfoque se tiene la variante en un caso se realiza el mismo tratamiento de ignorar palabras mas y menos frecuentes del corpus y en el otro caso no se aplica este tratamiento.

3.5. Vectorización o Bolsa de Palabras

Una vez extraídas las características, se realizó el proceso de vectorización el cual se aplica en ambos enfoques. La vectorización o también llamada bolsa de palabras, realiza la transformación a una matriz, en la cual se obtiene la frecuencia por característica presente en cada tuit. Para realizar esta tarea se utilizó *vect.transform(train)*, este método usa el vocabulario ajustado para crear la matriz. Si lo visualizáramos como un *DataFrame* con *Pandas* tendría un aspecto como en la tabla 3, en la cual la primera fila corresponde a las características obtenidas en la extracción y luego por cada tuit se tiene la frecuencia en que aparecen, cabe señalar que las características no contienen un orden específico.

Tuit	Etiqueta-Num	Preproceso	con\PREP	el\ART	todo\QU	...
'Suenan bien! La vida es muy corta para no celebrar el amor'	0	'Sonar\VLfin bien\ADV !\FS el\ART vida\NC ser\VSF muy\ADV corto\ADJ para\CSUBI no\NEG celebrar\VLinf el\ART amor\NC'	0	1	0	
'Estoy cansadísima de todo ya x favor'	1	'Estar\VEfm cansadísima\ADJ de\PREP todo\QU ya\ADV x\CARD favor\NC'	0	0	1	
'Gracias a todos por respetar a la gente con pareja'	2	'Gracia\NC a\PREP todo\QU por\PREP respetar\VLinf a\PREP el\ART gente\NC con\PREP pareja\NC'	1	1	1	
...						

Tabla 3: Bolsa de Palabras. Fuente: Elaboración Propia

3.6. Entrenamiento Algoritmo de Clasificación

El algoritmo utilizado para el aprendizaje supervisado probabilístico fue Multinomial Naive Bayes, esto debido a que en el trabajo experimental se utiliza la técnica de bolsa de palabras, donde se manejan datos enteros al obtener la frecuencia de aparición de cada característica encontrada en el tuit, es sabido que Naive Bayes clasifica mejor este tipo de datos además que soporta la múltiple clasificación. Por otra parte, Naive Bayes obtuvo mejores resultados en la literatura revisada y es un algoritmo fácil de entender y de rápida ejecución comparado con otros algoritmos supervisados.

El algoritmo Multinomial Naive Bayes, fue aplicado para el entrenamiento de ambos enfoques. El entrenamiento se realizó al 75 % de los datos es decir 2531 tuits, el objetivo fue predecir la clasificación múltiple: positivo, negativo o irónico.

4. Evaluación de los Resultados y Discusión

4.1. Evaluación de los Resultados

Una vez entrenado el modelo se realizó la predicción del conjunto de datos de prueba a través de *nb.predict* de *Sklearn*. El conjunto de pruebas corresponde al 25 % del *corpus*, es decir 844 tuits.

Para la evaluación de los resultados, se obtuvo la matriz de confusión y se calcularon las métricas *accuracy*, *precisión*, *recall* y *f1-score*.

Enfoque 1 - Evaluación de los Resultados

- **Matriz de confusión unigramas:**

En tabla 4 podemos observar la matriz de confusión en la cual tenemos que de 293 muestras que son positivas se predijeron 137 de forma correcta y 156 de forma incorrecta. De 289 muestras negativas se predijeron 173 de forma correcta y 116 de forma incorrecta. De 262 muestras irónicas se predijeron de forma correcta 258 y 4 de forma incorrecta.

Clase	Positivo	Negativo	Irónico
Positivo	137	111	45
Negativo	83	173	33
Irónico	4	0	258

Tabla 4: Matriz-Unigramas

- **Matriz de confusión bigramas:**

En tabla 5 podemos observar la matriz de confusión en la cual tenemos que de 293 muestras que son positivas se predijeron únicamente 27 de forma correcta y 266 de forma incorrecta. De 289 muestras negativas se predijeron 37 de forma correcta y 252 de forma incorrecta. De 262 muestras irónicas se predijeron de forma correcta 240 y 22 de forma incorrecta.

Enfoque 2 - Evaluación de los Resultados

- **Matriz de confusión lema/pos:**

En tabla 6 podemos observar la matriz de confusión en la cual tenemos que

Clase	Positivo	Negativo	Irónico
Positivo	27	16	250
Negativo	17	37	235
Irónico	9	13	240

Tabla 5: Matriz-Bigramas

de 293 muestras que son positivas se predijeron 139 de forma correcta y 154 de forma incorrecta. De 289 muestras negativas se predijeron 168 de forma correcta y 121 de forma incorrecta. De 262 muestras irónicas se predijeron de forma correcta 260 y 2 de forma incorrecta.

Clase	Positivo	Negativo	Irónico
Positivo	139	96	58
Negativo	73	168	48
Irónico	2	0	260

Tabla 6: Matriz-lema/pos

■ **Matriz de confusión lema/pos + reducción de palabras frecuentes y menos frecuentes:**

En tabla 7 podemos observar la matriz de confusión en la cual tenemos que de 293 muestras que son positivas se predijeron 144 de forma correcta y 149 de forma incorrecta. De 289 muestras negativas se predijeron 176 de forma correcta y 113 de forma incorrecta. De 262 muestras irónicas se predijeron de forma correcta 256 y 6 de forma incorrecta.

Clase	Positivo	Negativo	Irónico
Positivo	144	110	39
Negativo	80	176	33
Irónico	3	3	256

Tabla 7: Matriz-lema/pos y red. palabras

■ **Comparativa de los dos enfoques en cuanto a las métricas obtenidas:**

En tabla 8 se puede observar que el mejor resultado fue obtenido del enfoque 2 resultado de aplicación de *Tree Tagger* y reducción de palabras mas y menos frecuentes. Por otra parte el que peores resultados obtuvo es el enfoque 1 con su variante bigramas. Curiosamente el enfoque 1 variante unigramas y el enfoque 2 variante lema/pos tuvieron los mismos resultados.

En cuanto a las matrices de confusión se obtuvieron mejores resultados para la clase irónico, es posible que esto se deba a que los datos se encuentran con la palabra sarcasmo e ironía.

Variante del Enfoque	Accuracy	Precision	Recall	F1-score
Enfoque 1: Unigramas	0.67	0.66	0.67	0.66
Enfoque 1: Bigramas	0.36	0.47	0.36	0.28
Enfoque 2: lema/POS	0.67	0.66	0.67	0.66
Enfoque 2: lema/POS + reduccion de características mas y menos frecuentes	0.68	0.67	0.68	0.67

Tabla 8: Métricas Comparativa

4.2. Discusión sobre los Resultados

Los resultados obtenidos en los experimentos realizados, demuestran en general valores no muy altos en la mayoría de las métricas analizadas en ambos enfoques y en relación a resultados obtenidos en otras investigaciones sobre la minería de opiniones para el idioma inglés. Considerando que el presente trabajo tenía como objetivo realizar una clasificación multietiqueta de textos en español, determinando la polaridad del sentimiento (positivo y negativo) y a su vez detectando la ironía como problema de ambigüedad, se analizan los factores que de alguna manera han podido influir en los resultados.

Una de las primeras dificultades se tiene en la extracción de datos para formar el *corpus*, uno por la cantidad de datos obtenida y otro por su calidad con relación al etiquetado automático. Si bien muchos de los trabajos consultados en la literatura para el idioma inglés, utilizan datos extraídos de Twitter, la extracción la realizan durante mucho tiempo, al menos cuando se trata de datos como los relacionados con la ironía. Para el presente trabajo solo se pudieron hacer tres extracciones, ya que el método utilizado *search* para la extracción de tuits a través del *API Rest* de Twitter, solo permite obtener tuits de una semana de antigüedad, aun así los datos fueron escasos para el conjunto de tuits irónicos ya que además de etiquetar como *#ironia* y *#sarcasmo*, se tuvo que especificar que los datos a extraer eran en idioma español 'es', lo cual delimitó mucho más la cantidad de tuits obtenidos considerando que muchos usuarios no tienen habilitada la opción que especifica su idioma.

Por otra parte, si bien muchos autores utilizan el *hashtag #* como método confiable de recolección de tuits, como es el caso de tuits irónicos para el idioma inglés, el experimento realizado indica lo contrario. Realizando una simple revisión de datos nos damos cuenta que existen frases que realmente no corresponden a la ironía por más que estén etiquetadas en Twitter como *#ironía* o *#sarcasmo*, un ejemplo de esto son los tuits extraídos:

- 'Por que hay gente que no entiende el sarcasmo? !Que aburrida debe ser su vida sin comedia y humor!'
- 'Hay personas que no entienden el sarcasmo ni diciendoles: Lo que voy a decir es un sarcasmo o en Twitter colocando un HT · Sarcasmo'
- 'Este millennial si entiende bien lo que es SARCASMO'

Esto demuestra que el uso del *hashtag* como clasificador de tuits para el español no es del todo confiable. Consideramos que los resultados en la precisión mejorarán considerablemente si se sigue una de las propuestas de la literatura, que consiste en determinar primeramente si el tuit es objetivo o subjetivo antes de proceder a la clasificación. Otra opción sería realizar un etiquetado manual, aunque se requeriría de más tiempo y de revisores humanos, sin embargo la clasificación de la ironía no es un trabajo sencillo ni siquiera para el ojo humano.

Otro de los aspectos a considerar es el soporte de herramientas para el idioma Español. Las herramientas que se utilizan en la literatura para el preprocesamiento y la extracción de características, no son muy eficientes al ser aplicadas al idioma español, debido a la falta de desarrollo para este idioma. Un ejemplo de esto es la lematización, en NLTK el lematizador para el inglés es muy bueno y trae buenos resultados, sin embargo para el caso del español es prácticamente inexistente, por tal razón para este trabajo se tuvo que optar por el método *stemming* de NLTK que aunque existe para el español el mismo es deficiente. Para obtener el etiquetado gramatical sucede lo mismo, si bien existe la opción para el español, falta el etiquetado de muchas palabras, lo pudimos corroborar revisando la bolsa de palabras obtenidas de la aplicación con *Tree Tagger*.

5. Conclusiones y líneas de trabajo futuras

Tras haber realizado el experimento de múltiple clasificación con datos en español siguiendo las técnicas típicas de la minería de opiniones, no se pudo comprobar la posibilidad de solucionar al 100% el problema de ambigüedad de detección de ironía. Ya que si bien el segundo enfoque propuesto que utiliza *Tree Tagger* para obtener el lema/pos con exclusión de los términos más y menos frecuentes en la bolsa de palabras obtuvieron mejores resultados que el primer enfoque con unigramas y bigramas con NLTK, estos resultados no son suficientes ya que las métricas se encuentran por debajo de los 0.70. Sin embargo, considerando los problemas presentados como la cantidad de datos, la calidad del etiquetado y el soporte de herramientas para el idioma español, no se debe descartar la posibilidad de que aplicando otros enfoques y mejorando las técnicas, sea posible predecir con un mejor porcentaje de confiabilidad los textos irónicos en idioma español.

Como líneas de trabajos futuros, en primera instancia se debe obtener un *corpus* más confiable. Esto se puede realizar a través del etiquetado de clasificación manual o de forma semisupervisada. Sería necesario realizar una comparativa de

varias herramientas PLN para el idioma español. Primeramente se puede diseñar un modelo de clasificación binario que detecte si la opinión es irónica o no antes de predecir la polaridad. Para este propósito se pueden emplear otro tipo de enfoques como una combinación del aprendizaje supervisado con léxico, o en base a reglas.

Referencias

1. APPEL, O., CHICLANA, F., CARTER, J., AND FUJITA, H. A hybrid approach to the sentiment analysis problem at the sentence level. *Knowledge-Based Systems 108* (2016), 110 – 124. New Avenues in Knowledge Bases for Natural Language Processing.
2. BOUAZIZI, M., AND OTSUKI, T. A pattern-based approach for sarcasm detection on twitter. *IEEE Access 4* (2016), 5477–5488. cited By 7.
3. CHARALAMPAKIS, B., SPATHIS, D., KOUSLIS, E., AND KERMANIDIS, K. A comparison between semi-supervised and supervised text mining techniques on detecting irony in greek political tweets. *Engineering Applications of Artificial Intelligence 51* (2016), 50 – 57. Mining the Humanities: Technologies and Applications.
4. HEMMATIAN, F., AND SOHRABI, M. K. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review* (Dec 2017).
5. JOSHI, A., BHATTACHARYYA, P., AND CARMAN, M. J. Automatic sarcasm detection: A survey. *CoRR abs/1602.03426* (2016).
6. MANTYLA, M. V., GRAZIOTIN, D., AND KUUTILA, M. The evolution of sentiment analysis a review of research topics, venues, and top cited papers. *Computer Science Review 27* (2018), 16 – 32.
7. MUKHERJEE, S., AND BALA, P. Detecting sarcasm in customer tweets: An nlp based approach. *Industrial Management and Data Systems 117*, 6 (2017), 1109–1126. cited By 0.
8. R. PIRYANI, D. M., AND SINGH, V. Analytical mapping of opinion mining and sentiment analysis research during 2000 to 2015. *Information Processing and Management 53*, 1 (2017), 122 – 150.
9. SUN, S., LUO, C., AND CHEN, J. A review of natural language processing techniques for opinion mining systems. *Information Fusion 36* (2017), 10 – 25.
10. WICANA, S. G., IBISOGLU, T. Y., AND YAVANOGLU, U. A Review on Sarcasm Detection from Machine-Learning Perspective. In *2017 11TH IEEE INTERNATIONAL CONFERENCE ON SEMANTIC COMPUTING (ICSC)* (2017), IEEE International Conference on Semantic Computing, IEEE; IEEE Comp Soc, pp. 469–476. 11th IEEE International Conference on Semantic Computing (ICSC), San Diego, CA, JAN 30-FEB 01, 2017.
11. WIKIPEDIA. Análisis de sentimiento — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 12-julio-2018].
12. WIKIPEDIA. Pandas — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 12-julio-2018].
13. WIKIPEDIA CONTRIBUTORS. Natural language processing — Wikipedia, the free encyclopedia, 2018. [Online; accessed 12-July-2018].

Sistemas Conexionistas y de Visión Artificial para Detección de Iconografía Radical

Henri Estuardo Castro León, Angélica González Arrieta

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{henricastro, angelica}@usal.es
<http://diaweb.usal.es/>

Resumen Las tendencias modernas en detección de objetos sitúan este estudio en una división entre redes neuronales y demás técnicas de visión artificial. La aplicación de estas técnicas está determinada por la información disponible para el entrenamiento de los algoritmos que permitirán la predicción del objetivo. El experimento con arquitecturas basadas en redes neuronales deriva las conclusiones hacia otras heurísticas basadas en las propiedades de las imágenes. Se propone un ajuste fino para conseguir la mayor precisión en la detección de iconografía de grupos radicales, enfrentando problemas como ruido, deformaciones en la imagen y demás oclusión. Se valida la propuesta mediante experimentos que demuestran su efectiva aplicación en ambientes de producción.

Keywords: Computer vision, Object detection, Neural networks, Darkflow, Sift, Opencv

1. Introduction

En la nueva era tecnológica donde nos encontramos, el campo de la Visión Artificial *Computer Vision* ha tenido pronunciados avances hasta formar parte activa en: la industria, medicina, educación y la sociedad en general. Esto conlleva a una particular atención en su estudio y posterior reproducción, para enfocarla en tareas humanas que, por su naturaleza, son tediosas, demandan mayores recursos e ineficientes en términos de volumen de análisis.

Los sistemas de reconocimiento de patrones (campo de la visión artificial) mediante imágenes realizan la identificación o clasificación, tomando en cuenta un aprendizaje (supervisado o no supervisado) previo basado en gráficas con el objetivo target que se quiere detectar dentro de una imagen.

Muchos de los mecanismos de reconocimiento de patrones en imágenes están basados en Sistemas Conexionistas que utilizan algoritmos de Redes Neuronales Artificiales para computar un entrenamiento *training* que luego es usado para predecir (detectar) gráficos o íconos contenidos en imágenes que no fueron parte del entrenamiento previo. Este principio de funcionamiento es empleado en todas las arquitecturas de los Sistemas Conexionistas; por lo que, los datos de entrenamiento *dataset* son imprescindibles, haciendo esta parte de la tarea muy

complicada cuando la detección busca gráficos difíciles de conseguir, como por ejemplo iconografía radical.

Otro gran problema que enfrentan los sistemas conexionistas es que; las iconografías están presentes en superficies que no son sólidas, por ejemplo: ropa, banderas, etc. Por lo que la imagen muestra deformaciones producto de la posición, doblamiento y oclusión en general, afectando así la efectividad de la predicción.

En la actualidad existen otras heurísticas que buscan las relaciones *matching* entre imágenes como se menciona en el estudio [3]. Estos algoritmos se basan en la búsqueda de puntos claves *keypoints* que existen en la imagen y con los cuales se determina una similitud muy precisa y con un coste computacional mucho menor que el usado por las técnicas conexionistas.

2. Método propuesto

En esta sección se describe las técnicas y métodos utilizados para conseguir los objetivos propuestos. La extracción de características de las imágenes y su uso para realizar un emparejamiento con íconos de los grupos radicales.

2.1. Etiquetado de los datos de entrenamiento para un sistema conexionista

La mayoría de los detectores de objetos realizan el entrenamiento de los datos tomando como referencia los objetos a detectar mediante su ubicación en el espacio de la imagen; es decir, donde empieza y termina el objeto en una imagen. Una imagen puede contener varias anotaciones multidetección que hacen referencia a distintas clases.

Esta etapa del aprendizaje de las imágenes a través de redes neuronales es indispensable por su obvio propósito en materia de entrenamiento de la red. Para esta tarea existen herramientas que facilitan el etiquetado, lo cual implica que tiene que ser una tarea humana y con coste de tiempo muy representativa (argumento concluyente en posteriores secciones del documento).



Figura 1: Etiquetado de imágenes para entrenamiento de red neuronal con la herramienta "labeling"

La salida de este pre-proceso de entrenamiento es tradicionalmente un archivo de extensión xml con información posicional del objeto (iconografía en este caso), el nombre del fichero de la imagen y nombre de la clase, entre otros datos.

2.2. Implementación de Darkflow

Darkflow es la implementación oficial propuesta por sus autores en el paper [6] y [5]. Esta herramienta realiza la configuración y arquitectura de entrada para crear una red neuronal convolucional (según la metodología de YOLO). En una capa de más bajo nivel, esta implementación utiliza como biblioteca de Deep Learning a Tensorflow [1] la cual es donde se realizan los cálculos numéricos necesarios para las operaciones.

Usando las rutinas de Darkflow se realiza un entrenamiento con las imágenes que contiene iconografía de grupos radicales. Se cuenta con archivos que contienen pesos de un pre-entrenamiento con imágenes que es posible utilizar para propósitos locales como el de este estudio.

2.3. Técnicas de visión artificial

Son técnicas basadas en la identificación de la características de la imagen, aportan conocimiento sobre los objetivos que se van a indentificar como parte del estudio, basándose generalmente en las particularidades presentes en la distribución de los píxeles.

SIFT (Scale-Invariant Feature Transform) La función de transformación de escala invariable (SIFT) es una de las técnicas más populares para detección y coincidencia de características. Fue ideado por David Lowe y se considera ampliamente como el punto de referencia de facto para evaluar otros métodos.

El esquema de coincidencia de SIFT comienza por la difuminación de las versiones de muestra descendente y ascendente de la imagen con núcleos gaussianos, para simular el efecto de los cambios de escala. Los puntos clave se identifican seleccionando máximos y mínimos de diferencias en tales características distintivas de la imagen de los puntos clave invariantes de escala. La idea es seleccionar puntos que permanezcan estables frente a este escalado (simulado), de modo que sean detectables y se puedan aparear en imágenes de escala variable.

Luego se asignan orientaciones dominantes, mirando al gradiente de la imagen de un *kernel* que rodea un punto clave particular, y se alineará en la etapa de emparejamiento matching para ignorar posibles diferencias de rotación en el plano. La robustez a la distorsión local (causada, por ejemplo, al cambiar la perspectiva) se logra borrando y volviendo a muestrear los planos de orientación de la imagen local.

Una región local alrededor de un punto clave particular se normaliza con respecto a la escala, la orientación, las pequeñas distorsiones afines mencionadas anteriormente, y la iluminación. Finalmente se construye un vector de 128 elementos que servirá como el descriptor de la característica.

Las características de una imagen hacen referencia a un patrón de píxeles que forman una propiedad visual la cual se puede utilizar para identificarla, clasificarla o encontrar similitudes con otras imágenes, de forma parcial o completa.

Muchos de los algoritmos clásicos para esta tarea utilizan al gradiente de la imagen.

El gradiente de imagen Es un concepto que refiere al cambio (suficientemente pronunciado) de la dirección en la intensidad o el color de una imagen; tiene mucho uso también en la detección de bordes, contornos y más formas de procesamiento.

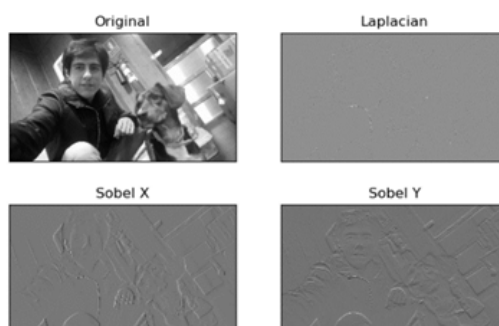


Figura2: Ejemplo de gradiente obtenida mediante un algoritmo de cálculo de gradiente (Laplacian, Sobel en x,y)

Con estos cálculos sobre los vectores numéricos de la imagen y su orientación (obtenida mediante la gradiente) esta técnica de visión artificial busca obtener los puntos clave, concepto sine qua non la propuesta de este documento tendría finalidad.

Es importante que las características de la imagen estén presentes en la mayor parte de las escalas posibles de la imagen para la ubicación de estas escalas se usa una función continua conocida como "el espacio de escala" descrito en: [7]

A mayor número de escalas donde esté presente una característica, mejor será su fiabilidad y eficiencia del algoritmo, esto es la frecuencia de muestreo en la que, de ser sobreutilizada haría solamente más pesado al algoritmo sin mejores resultados.

El descriptor de características La obtención de las regiones y escalados válidos tiene su propósito en esta parte del estudio ya que con estos parámetros se plantea un sistema de coordenadas de dos dimensiones ahora con la certeza que la varianza del escalado no inferirá en extracción de características.

El descriptor de característica es una región de la imagen altamente distintiva del resto de regiones tomando en cuenta un cierto nivel de invariabilidad, por

ejemplo: la exposición a la luz. Un descriptor se crea calculando primero la magnitud y la orientación del gradiente en cada punto de muestra de imagen en una región alrededor de la ubicación del punto clave.

El keypoint se determina por la longitud de cada flecha (dirección) correspondiente a la suma de las magnitudes de gradiente cerca de esa dirección dentro de una región de la imagen y escala, es decir: el círculo azul en esta figura.

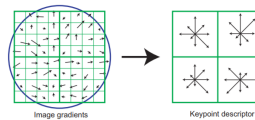


Figura 3: Direccionalidad de keypoints

El objetivo principal de esta técnica es la obtención de los *keypoints* de la imagen. Tanto como sea posible y de igual dimensionalidad, esta propiedad viene dada por el vector de características, representada por un vector numérico. En OpenCV los vectores de características se representan por arreglos numpy.



Figura 4: Obtención y marcado de keypoints en el ícono del grupo radical colombiano "Ejército de liberación nacional"



Figura 5: Obtención y marcado de keypoints en el ícono del grupo radical Isis

2.4. Emparejamiento de características

La parte más costosa desde el punto de vista computacional de muchos algoritmos de visión por computadora consiste en buscar las coincidencias más similares a los vectores de alta dimensión, también referido como el emparejamiento del vecino más cercano. Tener un algoritmo eficiente para realizar la coincidencia de vecinos más rápidos en grandes conjuntos de datos puede traer mejoras de velocidad de varios órdenes de magnitud para muchas aplicaciones.

Ejemplos de tales problemas incluyen encontrar las mejores coincidencias para las características de imágenes locales en conjuntos de datos grandes agrupando las características locales en palabras visuales usando kmeans o algoritmos similares. Igualación de características de imagen global para reconocimiento de escena, estimación de pose humana, emparejar formas deformables para reconocimiento de objetos o realizar correlación cruzada normalizada (NCC) para comparar características de íconos en grandes conjuntos de datos de imágenes. El problema de búsqueda del vecino más cercano también es de gran importancia en muchas otras aplicaciones, incluido el aprendizaje automático, la recuperación de documentos, la compresión de datos, la bioinformática y el análisis de datos.

Fast Approximate NN MatchingFL - ANN La búsqueda exacta es demasiado costosa para muchas aplicaciones, por lo que ha generado interés en los algoritmos de búsqueda de vecinos más cercanos que devuelven vecinos no óptimos en algunos casos o falsos positivos.

Dentro de esta familia de algoritmos de emparejamiento, el que mejor resultado experimental presenta es: "búsqueda de prioridad árbol k-means o los múltiples árboles aleatorizados k-d". La propuesta de este estudio desarrolla la parte de resultados usando este algoritmo.

The Randomized k-d Tree Algorithm El algoritmo de árbol aleatorio k-d es un algoritmo de búsqueda del vecino más cercano que construye múltiples árboles aleatorios que se procesan en paralelo. Los árboles se construyen de manera similar al árbol kd clásico [4], con la diferencia que, donde el algoritmo de árbol clásico divide los datos con la varianza más alta, los árboles kd aleatorizados divide las dimensiones al azar desde las dimensiones superiores con la mayor varianza.

La búsqueda en este bosque aleatorio k-d mantiene una cola de prioridad única en todos los árboles aleatorizados. La cola de prioridad se ordena aumentando la distancia al límite de decisión de cada rama en la cola, por lo que la búsqueda explorará primero las hojas más cercanas de todos los árboles. Una vez que se ha examinado un punto de datos (en comparación con el punto de consulta) dentro de un árbol, se marcará para no volver a examinarse en otro árbol. El grado de aproximación está determinado por el número máximo de hojas que se visitarán en todos los árboles, devolviendo los mejores candidatos vecinales más cercanos encontrados hasta ese punto.

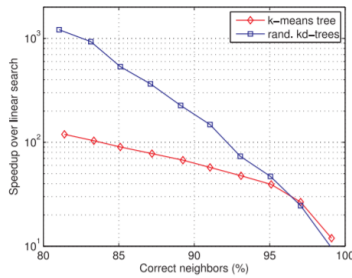


Figura 6: Se muestra el valor de búsqueda en muchos árboles kd aleatorizados al mismo tiempo, se puede observar que el rendimiento mejora con la cantidad de árboles aleatorizados. (kd-trees clásico frente a aleatorizado)

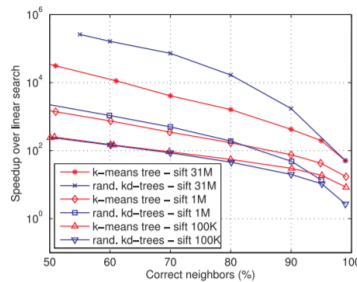


Figura 7: Aceleración de búsqueda para diferentes tamaños de conjuntos de datos

3. Resultados experimentales

En esta sección se presentan los resultados de las técnicas aplicadas utilizando la metodología propuesta. Se realiza la experimentación con imágenes de grupos radicales en distintos escenarios y con distinta posición iconográfica.

3.1. Sistemas conexionistas

Esta sección describe y visualiza los resultados obtenidos de la aplicación e implementación de la arquitectura YOLO propuesta método de detección. Además indica de manera global la distribución de los datos de entrenamiento y cómo fueron usados para el propósito.

Dakflow y Tensorflow Para este propósito se utilizó la implementación oficial propuesta por los autores del método You Only Look Once, esta es DarkFlow, la cual está escrito en Python y utiliza el famoso api de Google llamado Tensorflow como motor de deep learning. La arquitectura final de la red convolucional que arma tensorflow permanece "invisible" para el estudio porque este framework realiza los cálculos a bajo nivel para obtener velocidad en la etapa de training.

A pesar de que actualmente la tendencia es usar la unidad de procesamiento gráfico de los ordenadores GPU [8], el presente estudio se realizó usando únicamente la cpu tradicional. Es muy recomendable realizar estas pruebas sobre la gráfica para reducir considerablemente el tiempo de entrenamiento de los datos . Esta recomendación también podría ser usada en muchas otras tareas de aprendizaje profundo.

Los datos de entrenamiento Durante la investigación previa para el presente estudio, se seleccionaron grupos radicales con suficiente apareamiento en medios de publicidad con el fin de subir el volumen de datos de entrenamiento y su eficiencia para las técnicas conexionistas.

Grupo radical	Número de imágenes
Hogar Social Madrid	57
Golden Dawn	54
Isis	25

Tabla 1: Dataset de entrenamiento



Figura 8: Golden Dawn



Figura 9: Hogar Social Madrid

Íconos



Figura 10: Isis

Preparación de los datos para el entrenamiento Usando la herramienta LabelImg se procedió a el etiquetado de todas las imágenes que se tienen para que el algoritmo conozca exactamente donde se encuentra el objetivo y realice el cálculo correcto de los pesos weights.



Figura 11: Ejemplo del etiquetado de los datos, cuadros verdes. Grupo HSM

La representación final de esta tarea es un archivo formato xml que indica las coordenadas 'x' e 'y' cartesianas de la posición donde se encuentra el target, en nuestro caso el ícono. Se pueden etiquetar varios íconos en una misma imagen, lo que es muy favorable ya que esto amplía el volumen del dataset de entrenamiento.

El verdadero gran problema de esto es que debe ser de forma irrestricta una tarea humana, lo cual cambia la manera global el empleo de esta técnica para un propósito tan fino como el de este estudio.

Ejecución del entrenamiento. Darkflow provee una interfaz a través de línea de comando en la cual se envían los argumentos con los datos del modelo y configuración de la red neuronal. En este caso son los valores por defectos para un nuevo modelo. El entrenamiento de los datos toma entre diez y quince horas en un ordenador de características normales únicamente usando la CPU, aunque es muy recomendable usar la GPU para este tipo de cómputos [2]

Resultados (predicción) Darkflow también presta una interfaz para que otras aplicaciones puedan hacer uso de las predicciones, este código podría ser utilizado para conectar con otros servicios para ambientes de producción. El resultado es devuelto en formato json el cual contiene las coordenadas de la detección y su confianza.


```

from darkflow.net.build import TFNet
import cv2

options = {'model': 'cfg/yolo.cfg', 'load': 'bin/yolo_weights', 'threshold': 0.1}
tfnet = TFNet(options)
imgcv = cv2.imread("/home/shared/yolo/darkflow-master/hd.jpg")
result = tfnet.return_predict(imgcv)
print(result)

```

Figura 12: Consulta externa a una predicción a partir del entrenamiento realizado.



Figura 13: Imagen de "Golden Dawn" a predecir

Tiempo de consulta: 3.2881 segundos. La predicción a encontrado un ícono dentro de la imagen a predecir, pero la confianza con la que se cuenta no es un *confidence score* aceptable según [9].

3.2. Detección basada en detección de características

Para la implementación de los algoritmos se utilizó la librería de código abierto OpenCV y su submódulo OpenCV-Contrib-Python, ambas compiladas para ser usadas con lenguaje Python. OpenCV es una librería lo bastante madura para dar soporte al afinamiento de algoritmos (en su mayoría) desde su parte matemática core. La flexibilidad de Python nos da la libertad de enviar nuevos parámetros al algoritmo o -de no hacerlo- usa los valores por defecto.

Todos los tratamientos que se realizan sobre las imágenes deben estar primeramente convertidas/representadas a formato de arreglo array con base de entero de ocho bits, o sea: uint8. Esta tarea básica la realiza opencv de manera explícita.

La documentación oficial de opencv describe el uso de los algoritmos en su nivel paramétrico, es decir: las entradas y las salidas devueltas por sus funciones, lo que aporta una sólida base e inicio de la experimentación. Las pruebas se realizan aplicando los algoritmos con sus valores por defecto y luego afinándolos para conseguir los mejores resultados.

Los objetivos de esta parte de los experimentos están enfocados en incrementar el número de keypoints que se extraen de la imagen.

3.3. SIFT (Scale-Invariant Feature Transform)

Los valores que el algoritmo carga al momento de la instanciación de su clase están definidos de la siguiente manera:

- **Nfeatures**: La cantidad de mejores características para retener. Las características se clasifican por sus puntajes (medidos en algoritmo SIFT como el contraste local). Valor por defecto es cero, lo que indica que nos dará el máximo valor de keypoints que encuentra.
- **NOctaveLayers**: El número de capas en cada octava El número de octavas se calcula automáticamente a partir de la resolución de la imagen. Valor por defecto es 3.
- **ContrastThreshold**: El umbral de contraste utilizado para filtrar las características débiles en regiones semi uniformes, o sea el bajo contraste. Cuanto mayor es el umbral, menos características produce el detector. Valor por defecto 0.04.
- **EdgeThreshold**: El umbral utilizado para filtrar las características de borde. Valor por defecto 10.
- **Sigma**: La sigma del Gaussiano aplicado a la imagen de entrada en la octava cero. Valor por defecto 1.6.

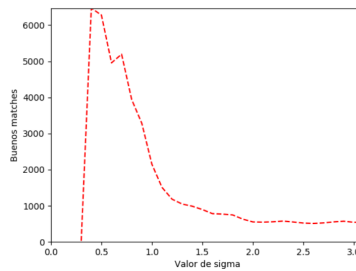


Figura 14: Experimentación del valor sigma. El ajuste de este valor en el algoritmo incrementa de manera muy pronunciada el encuentro de más keypoints en la imagen. Para este caso, nos quedamos con el nuevo valor de $\sigma = 0.4$

Experimentación del valor sigma

3.4. Resultados obtenidos

La experimentación con los parámetros de entrada del algoritmo nos devuelve un mayor número de coincidencias de características representativas en la imagen. Pero este número mayor de keypoints encontrados en la imagen no es una garantía de que el algoritmo esté trabajando bien, ya que cuando se realiza el plot de la imagen se pueden observar el gran número de falsos positivos. Esto es una gran desventaja para la aplicación de algoritmo.

Distribución de imágenes para pruebas

Parámetros usados para la experimentación con SIFT

Es decir qué: un match positivo sería sobre los 80 keypoints encontrados. Tomar en cuenta que este no es el total de keypoints, sino el total de keypoints

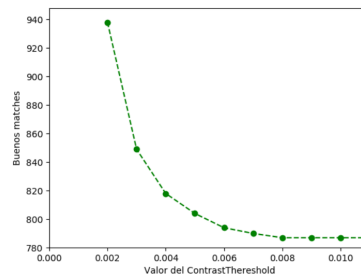


Figura 15: Experimentación del valor contrast threshold. El mejor valor para este parámetro es 0.02. Se puede notar que: a menor valor del contrast threshold mayor es el número de buenos matches obtenidos.

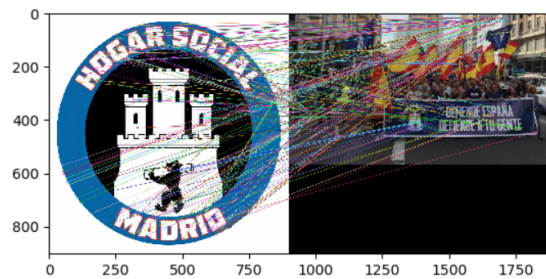


Figura 16: Ejemplo típico de algoritmo sobre ajustado, falsos positivos.

Folder	Número
HSM	50
Sin íconos	50

Tabla 2: Distribucion carpetas

Parámetro	Valor
Nfeatures	0
NOctaveLayers	3
ContrastThreshold	0.4
EdgeThreshold	10
Sigma	1.6

Tabla 3: Parámetros del algoritmo usado con SIFT

que se encuentran agrupados, contenidos o al rededor del ícono. Este sería el criterio más importante de la decisión de si una imagen contiene o no un ícono radical.

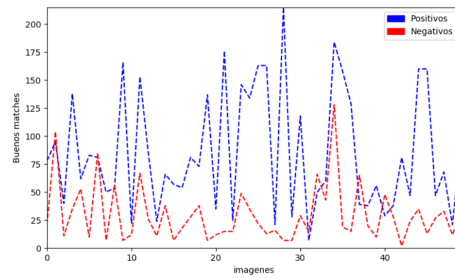


Figura 17: Gráfica de las mejores keypoints encontrados y que están agrupados cerca en imágenes positivas y negativas. Para este caso se experimentó corriendo en algoritmo sobre las 50 imágenes que contenían iconografía de Hogar Social Madrid y luego otras 50 con imágenes de ambientes normales sin ningún tipo de iconografía

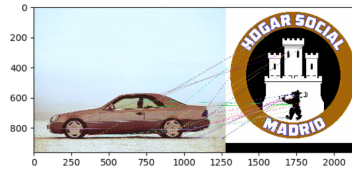


Figura 18: Algoritmo SIFT aplicado sobre una imagen que no contiene un ícono, este es un caso Negativo.

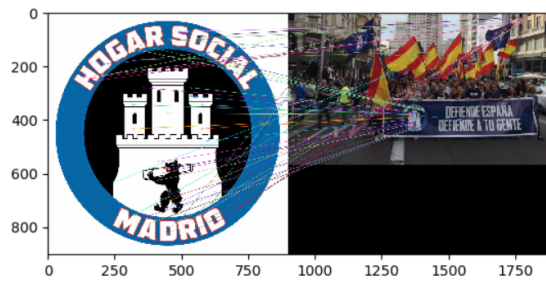


Figura 19: Algoritmo SIFT funcionando eficientemente.

Tipo de imagen	Número de imágenes	Media de keypoints emparejados
Positivas	50	85
Negativas	50	29

Tabla 4: Resultados obtenidos de hacer match del ícono de HSM contra 50 imágenes con iconografía y 50 sin ella.

3.5. Resumen de aplicación de detectores de características de OpenCV

Nombre del algoritmo	K encontrados en la img	Decs de los kps
SIFT	1583	128
SURF	1614	64
KAZE	1572	64
AKAZE	868	61
BRISK	2467	64

Tabla 5: Total de descriptores de kps de algoritmos de detección

4. Conclusiones y futuras líneas de trabajo

Las conclusiones que se derivan de los experimentos e investigación realizados están enfocadas explícitamente a la aplicación de estas técnicas sobre el caso particular de “detección de iconografía radical”. Los enfoques podrían variar en cualquier sentido cuando sean realizados sobre otro tipo de información en cuanto a: volumen de datos, tamaños de imagen, objetivo a detectar, etc.

- Se ha demostrado que el uso de las técnicas conexionistas para detección de objetos en imágenes conserva una estricta relación proporcional entre su eficiencia y los datos de entrenamiento.
- Las técnicas conexionistas necesitan un apropiado tratamiento en el etiquetado de sus imágenes objetivo para evitar ingresar ruido y restar eficiencia en el cálculo de sus pesos y posterior predicción. Por lo tanto, la tarea de etiquetado de las imágenes es una tarea imprescindible para cualquier aplicación de esta técnica.
- Se ha estimado la poca eficiencia de las técnicas conexionistas en la detección de iconografía radical por el motivo de que estas imágenes no son abundantes en ningún escenario.
- Se ha puesto en evidencia que los detectores de características de las imágenes son técnicas altamente fiables en especial cuando se necesita detectar íconos, esto debido a que las gráficas de diseño icónico presentan trazos particulares lo cual incrementa la eficiencia de los algoritmos actuales para este propósito.
- Se consiguió verificar que un número igual o mayor que 80 keypoints cercanos, es un valor muy aceptable para dar positivo en una detección de íconos en imágenes.

Para esta área de investigación y basado en el presente estudio se proponen las siguientes líneas de trabajo:

- Estudio de los detectores de características en imágenes en contexto de manifestación, es decir: donde la oclusión pueda ser anticipada y tratada de manera apropiada para imágenes al aire libre o en las calles.

- Realizar un estudio matemático para calcular las ondulaciones o deformaciones que un ícono sufrirá al estar impreso en materiales textiles como ropa o papel. Una ecuación que describa estas traslaciones y pueda servir para entrenar mejor los algoritmos.

Referencias

1. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from [tensorflow.org](https://www.tensorflow.org/).
2. Ammar Ahmad Awan, Hari Subramoni, and D.K. Panda. An in-depth performance characterization of cpu- and gpu-based dnn training on modern architectures, 11 2017.
3. J. Liu, Q. Wu, and X. Li. Research on image matching algorithm based on local invariant features. In *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 113–116, Oct 2013.
4. M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36 (11):2227–2240, Nov 2014. ISSN 0162-8828.
5. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.
6. Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.
7. Andrew P. Witkin. Scale-space filtering. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'83*, pages 1019–1022, San Francisco, CA, USA, 1983. Morgan Kaufmann Publishers Inc. URL <http://dl.acm.org/citation.cfm?id=1623516.1623607>.
8. G. Xianming, W. Baosheng, Z. Xiaozhe, and W. Xu'an. Software data plane and flow switching plane separation in next-generation router architecture. In *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 194–199, Nov 2015.
9. W. Zhiqiang and L. Jun. A review of object detection based on convolutional neural network. In *2017 36th Chinese Control Conference (CCC)*, pages 11104–11109, July 2017.

Prediction of Future Immediate Actions

Mehmet Ozturk

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{mehmet}@usal.es
<http://diaweb.usal.es/>

Resumen Human action recognition and prediction are two of the important areas in computer vision. The researches and outputs are being used in various contexts like security surveillance systems, robotics and human-computer interaction. While human action recognition is a common research area, the prediction of human actions is still an area that needs more attention and research. The work in this paper focuses to design an easy-to-implement intelligent system to recognize humans/objects by using various basic object detection algorithms and predict their immediate future actions by training a model in neural networks as one of the main machine learning approaches. The system will predict the immediate future location of a moving object/human. The experiments that have been done are also provided step by step in this paper. The pieces of codes that have been used and the results of the experiments are provided as well.

1. Introduction

In recent years, our machines have learned to see and understand their environment. Now our cameras detect our faces in images we take, and social network applications can even recognize people in the photos we upload from these cameras. In the next few years, we will experience an even more radical transformation. In a short time, the cars will be driven by themselves which is already happening, our cell phones can read and translate a letter in any language, and our x-rays and other medical images will be read and analyzed by powerful algorithms that can accurately suggest a medical diagnosis, and even recommend effective treatments.

Machine learning is becoming an important field with a combination of computing power, many image data and a set of ideas taken from mathematics, statistics and computer science. This fast-growing intersection of machine learning has taken off, affecting many of our day-to-day interactions with the world and each other. One of the most important characteristics of the current machine learning changes in computer vision is that it depends to a large extent on software tools that are freely available and developed by large groups of volunteers, amateurs, scientists and engineers in open source communities.

Machine learning is no longer just a moda term, it is all around us: from the preventing spams of our emails to the automatic labelling of friends in pictures,

and the prediction of the movies we like. As a subfield of data science, machine learning allows computers to learn through experience: making predictions about the future using data collected from the past.

One of the notable usages of machine learning is predicting future. Predicting immediate human actions has a variety of applications ranging from the interaction between humans and robots to the detecting unusual situations in surveillance videos and service-based algorithms for personal or health care purposes. As an example, in the autonomous health services, consider an agent: monitor the activities of a patient, trying to predict if the patient is losing their balance. If the agent is able to predict the next action, it could determine if the patient could fall and take an action to try to prevent it. Some people say that tennis players predict where the ball will go by looking at the actions of opposite players since the ball moves very fast so that a human being cannot easily predict its future location. Very similar to this example, now the machines try to predict the immediate future by recognizing and analyzing the action of humans.

In this work, I have focused on the detection of humans in images, possibly taken from videos and predict the following actions based on these images. I have tried to predict the next location of a human / or object in motion by using the location points obtained from the previous movements of a human being. An open source computer vision library called OpenCV was used to detect objects and create a model with its neural network class. [2] In the following section, basic concepts and various methodologies have been explained to detect objects/humans and to predict future actions. Section 3 describes the purpose of the document. In section 4, the experiments that have been carried out, from the generation images to the detection of humans and the prediction of future actions are explained and in section 5, the results of the work and future work ideas were described.

2. Proposed Method

This section describes both the proposed algorithms to detect objects, implementation of artificial neural network and the system.

2.1. Tools used

In this work, .NET Framework with C# has been used to generate random data and CSV files. To read CSV files, generate images, recognize objects and train a neural network, we used an open source computer vision library called OpenCV [2]. As an integrated development environment, Visual Studio 2017 has been used.

2.2. Generating data and images

In order to predict a future action, we need to train a neural network by using an existing dataset. The more data we have, the better predictions the neural

network can make. After lots of researches on the internet, it is understood that it is hard to find this data set, that's why I have decided to generate the data myself. Since the approach in this work is detecting people (circles) in the sequence of images and predict the future actions, we can create random data sets which simulate a human moving to different directions.

To generate random coordinates that simulate a human being that is moving to the right, we have created ten values of x, each one is greater than the previous one. A console application has been created to generate random data and write that data into a CSV file.

For each direction, 20000 coordinates of the human moves have been created. This data can be assumed that a coordinate points of a moving object such as a car or human and 16000 to train and 4000 images to test. As there are 4 directions that a human can reach, and as each image has 10 following actions and each step has its own image, after creating everything, we have $10 * 80000 = 800,000$ images. It is almost impossible find that amount of images or frames on the Internet. The best way would be to extract them from the video recordings, but it would take a lot of time. That is why we are creating our own data and images to work with.

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
0	0.03	0.13	0.3	0.4	0.43	0.59	0.68	0.77	0.86
0	0.05	0.15	0.29	0.35	0.37	0.43	0.61	0.67	0.69
0	0.02	0.18	0.25	0.27	0.28	0.44	0.56	0.59	0.73
0	0.02	0.14	0.18	0.28	0.39	0.4	0.6	0.8	0.84
0	0.07	0.15	0.2	0.25	0.32	0.47	0.49	0.65	0.73
0	0.04	0.19	0.29	0.35	0.46	0.48	0.66	0.75	0.86
0	0.03	0.11	0.13	0.27	0.4	0.5	0.54	0.6	0.83
0	0.04	0.11	0.23	0.39	0.46	0.5	0.58	0.59	0.77
0	0.03	0.14	0.23	0.32	0.37	0.58	0.64	0.7	0.79
0	0.09	0.18	0.19	0.33	0.4	0.41	0.5	0.66	0.77

Tabla 1: Initial records of data showing that a human goes to the right.

Direction	Training Images	Test Images
Right	16000	4000
Left	16000	4000
Up	16000	4000
Down	16000	4000
Sum	64000	16000

Tabla 2: Number of images (sequence with 10 coordinate points) produced

Each data in a single cell is a point in the coordinate system. We have used the frame of the images as corodinate system.



Figura 1: Coordinate system used.

Once we have the position data in CSV file, we have generated the images using OpenCV. First we read the CSV files with OpenCV and create the image files with the following code.

```
Mat imageTfm(600, 300, CV_8UC3, Scalar(255, 255, 255));
Point centerOfCircle = Point(400, 300);
circle(imageTfm, centerOfCircle, 100, Scalar(0, 0, 0),5,8);
imwrite("coordinate1", imageTfm);
```

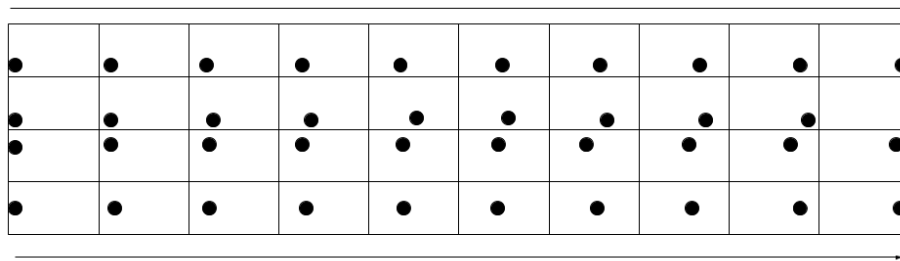


Figura 2: Four image sequences simulates an object moving to the right.

2.3. Detecting circles

In order to obtain position data from images, the system needs to detect circles, hence position data for each image. Hough Circle Transform method has been used to detect circles.

The detection of lines and circles in an image is a fundamental problem in image processing applications. The extraction of circles of digital images has received more attention for several decades because an extracted circle can be used to generate the location of the circular object in many industrial applications. So far, many circular extraction methods have been developed. The Hough Circle Transformation (HCT) is one of the best-known algorithms and aims to find circular shapes with a given radius r inside an image.[1]

The Hough transform is an effective method of detecting binary images in a straight line, circle, ellipse and other graphics. Later, it was proposed that the generalized Hough transform can detect graphics arbitrarily. The Hough transform has been applied not only to the graphic identification of recognition of borders, but also to biomedical, image processing of office documents, SAR / ISAR image processing and automatic interpretation of aerial images. [4]

OpenCV has a ready-to-use class for HCT. In a few simple steps, the circles in the object can be detected.

```
Mat src = imread(filename, IMREAD_COLOR); // read the imagefile
Mat gray;
cvtColor(src, gray, COLOR_BGR2GRAY); // convert to gray
medianBlur(gray, gray, 5); / add median blur
vector<Vec3f> circles;
HoughCircles(gray, circles, HOUGH_GRADIENT, 1, gray.rows / 16,100, 30, 1,30);
```

The HCT function of OpenCV detects the circles, draws a frame around it and creates a vector from those circles Figure:3. From these vectors, we can obtain their position in the image. We will train the neural network model using that position of data acquired from the detected images. Depending on the number of circles in the images, we can optimize the parameters of the functions so that the method recognizes better. In our example, we have only one circle per moment, but in real life there may be more circles in a frame.

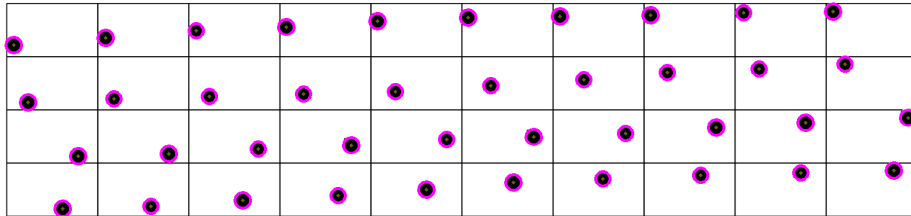


Figura3: Detected circles by HCT algorithm. Circles simulates a human moving to up

After detecting circles, the CSV files which includes the position points of each circle has been created.

2.4. Creating the artificial neural network and training

Computer Vision uses several machine learning algorithms to achieve different things. OpenCV provides a module called “ml” that has many integrated machine learning algorithms. Some of the algorithms include Bayes Classifier, K-Nearest Neighbors, Support Vector Machines, Decision Trees, Neural Networks, and more. It also has a module called flanne that contains algorithms for quick searches of nearby neighbors in large data sets. Machine learning algorithms are widely used to build object recognition, image classification, face detection, visual search, etc. [3]

Researches on artificial neural networks, commonly known as "neural networks", has been motivated from the beginning by the recognition that the human brain competes in a completely different way than the conventional digital computer. The brain is highly complex, non-linear and parallel computer (information processing system). It has the ability to organize its structural components, known as neurons, to perform certain calculations (for example, pattern recognition, perception and motor control) many times faster than the fastest digital computer that exists today.

For example, human vision, which is a task of information processing. The function of the visual system is to provide a representation of the environment that surrounds us and, more importantly, to provide the information we need to interact with the environment. To be specific, the brain routinely performs perceptual recognition tasks (for example, recognizing a familiar face embedded in an unfamiliar scene) in approximately 100-200ms, while much less complex tasks take much longer in a powerful computer.

How does a human brain do it? From birth, a brain already has considerable structure and the ability to build its own rules of behavior through what we generally call “experience.” In fact, the experience is built over time, with much of the development (that is, wiring) of the human brain that takes place during the first two years from birth, but development continues well beyond that stage.

In its most general form, a neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest: the network is usually implemented through the use of electronic components or simulated in a software package. a digital computer.

When we speak of a neural network, we should say more appropriately “Artificial Neural Network, Artifical Neural Network (ANN)”. Biological neural networks are much more complicated than the mathematical models we use for “artificial neural networks”.

There is no universally accepted definition of a neural network. But perhaps most people in the field would agree that a neural network is a network of many simple processors (“ units ”), each of which possibly has a small amount of local memory. The units are connected by communication channels (“ connections ”) that generally carry numerical data (instead of symbolic), encoded by any of several means. The units operate only on their local data and on the inputs they receive through the connections.

Some artificial neural networks are models of biological neural networks and some are not, but historically, much of the inspiration for the field of neural networks came from the desire to produce artificial systems capable of sophisticated calculations, perhaps “ smart ” , similar to those that the human brain performs routinely, and therefore possibly to improve our understanding of the human brain.

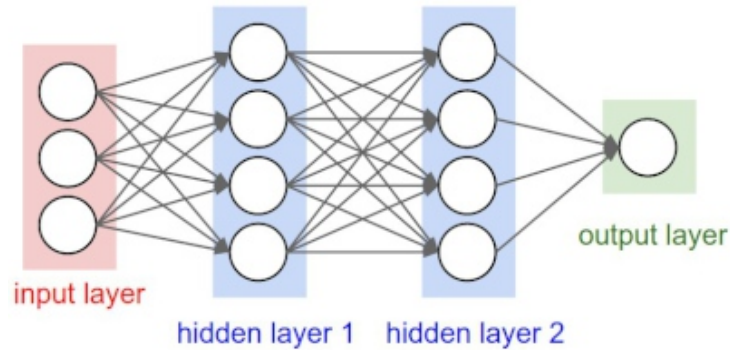


Figura 4: A Multilayer neural network

In this work, a type of artificial neural networks called Multilayer perceptron has been used, also called backpropagation. The multilayer perceptron (MLP, also known as back-propagation) is a neural network that is still among the highest performance classifications, especially for text recognition. It can be quite slow in training because it uses the gradient descent to minimize the error by adjusting the weighted connections between the numerical classification nodes within the layers. In testing, however, it is quite fast: just a series of point products followed by a crush function. In OpenCV it is implemented in the class `CvANN_MLP`.

As we have the data obtained from the images in CSV files, the first step is to read the data from the CSV files.

```
CvMLData testData;
CvMLData testResponse;
CvMLData trainData;
CvMLData trainResponse;

testData.read_csv(TEST_FILE);
testResponse.read_csv(TEST_RESPONSE_FILE);
trainData.read_csv(TRAIN_FILE) ;
trainResponse.read_csv(TRAIN_RESPONSE_FILE)

Mat testMatrix = testData.get_values();
Mat testResponseMatrix = testResponse.get_values();
```

```
Mat trainMatrix = trainData.get_values();
Mat trainResponseMatrix = trainResponse.get_values();
```

After reading the data, we create a matrix using that data. Note that we have training and test data and their response values in separate data sets.

We have 16000 records in training data and 4000 records in test data. Each record has 9 attributes. Through the use of training data, the neural network has been created and trained. It has 9 input values since the data has that number of attributes. We have 5 neurons in the hidden layer and 1 output value in the last layer. The output will be the predicted last attribute.

The following code creates a multilayer perceptron (MLP), sets the parameters such as the number of layers, the training method (BACKPROP) and starts the training.

```
CvANN_MLP mlp;
CvANN_MLP_TrainParams params;
CvTermCriteria criteria;

Mat layers = Mat(3, 1, CV_32SC1);
layers.row(0) = Scalar(9);
layers.row(1) = Scalar(5);
layers.row(2) = Scalar(1);

criteria.max_iter = 300;
criteria.epsilon = 0.01;
criteria.type = CV_TERMCRIT_ITER | CV_TERMCRIT_EPS;
params.train_method = CvANN_MLP_TrainParams::BACKPROP;
params.bp_dw_scale = 0.05;
params.bp_moment_scale = 0.05;
params.term_crit = criteria;
mlp.create(layers);

mlp.train(trainMatrix, trainResponseMatrix, Mat(), Mat(), params);
```

For better results, we train 4 different neural networks for each direction, to the right, to the left, up and down. After training the neural network with the data, it is tested with the test data. For prediction, the OpenCV MLP class has its own function:

```
Mat predictions;
mlp.predict(testMatrix, predictions);
```

The “mlp.predict” function takes test matrix as input and creates another matrix with predicted values. Next, we can generate the values to see how successful the predictions were.

Once we have the trained model, we used OpenCV's functions to save it as a YML or XML file so that we can use it to do predictions.

```
cv::FileStorage fs("model_left.yml", cv::FileStorage::WRITE); // or xml
mlp.write(*fs, "mlp");
```

In order to do new prediction, we can provide 9 data inputs which simulates 9 recent data obtained from moving human and the model can predict the last position.

3. Results

In this work, we have generated 800000 records of data which simulates a human moving to different directions. After creating these CSV files with the data, using that we have created images by using OpenCV draw circle methi, next we have detected the circles on these images and obtained their position data to predict future position values. We have used HCT Hough Circle Transformation to detect circles on the images and used MLP class of OpenCV to train a neural network and create a model.

Briefly, these are the steps of the our system.

1. Create random data which has 10 position points(attributes) which simulates a human moving 4 different directions.
2. Create the images with circles by using the data generated in previous step
3. Detect the circles on the images and obtain their position value
4. Create a neural network and train it for each direction data
5. Make predictions and compare it with the actual value

In order to see the results of the predicted values of trained neural network, we print the real value of the predicted attribute, the predicted value and the last position just before the predicted position.

Response	Predicted	Last Coordinate
0.610	0.719	0.530
0.830	0.817	0.740
0.650	0.768	0.630
0.780	0.835	0.760
0.830	0.843	0.780
0.790	0.835	0.770
0.760	0.739	0.580
0.870	0.817	0.730
0.700	0.767	0.640
0.750	0.795	0.680
0.860	0.837	0.770
0.860	0.826	0.740
0.850	0.847	0.790

Response	Predicted	Last Coordinate
0.700	0.744	0.590
0.890	0.812	0.710
0.870	0.841	0.780
0.840	0.842	0.780
0.780	0.806	0.710
0.890	0.847	0.790
0.830	0.801	0.680
0.850	0.816	0.720
0.770	0.830	0.760
0.870	0.867	0.740
0.770	0.790	0.750
0.800	0.817	0.730

Tabla 3: Some of the results of the predictions made on the data that simulated a human being goes from left to right.

Since we have decided to use the coordinate system from 0 to 1, all values are within that range. Although there are differences between the predicted values and the real value, we can see that the trained neural network can predict the next position very close. By modifying the values of the parameters of the neural network, such as the maximum iteration, epsilon and the number of neurons in the hidden layer, the predicted values can be optimized. Various ideas that might be implemented in the future is explained in the section “ Conclusion and future work ”

4. Conclusion and Future work

In this work, it has been proposed to use neural networks to predict the immediate future actions of humans. In the case of this work, we have tried to predict the future position of a human / object in motion.

The proposal is to create a easy-to-implement system that predicts the immediate future position. To do so, we have proposed to generate random location data of a moving human, generate images with that data, detect the circles on the images to obtain their position and create a neural network to training and make predictions.

OpenCV’s easy to use functions such as HCT and MLP are very helpful to create this system. In fact, OpenCV is the only library needed to create and implement this system.

Based on this work, we can think of some real life implementations as following ideas:

In a real-life environment, such as the intersection of traffic, we can specify location values in advance and collect data from those points where pedestrians are moving. Then, we can train our neural network model so that it can predict

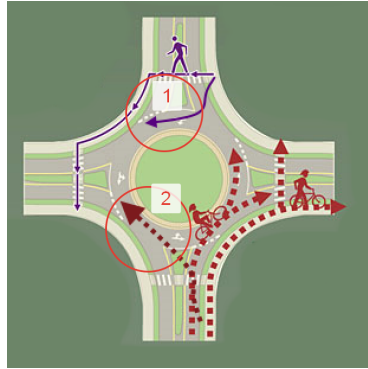


Figura5: Simple visualization of a pedestrian or cyclist going in the wrong directions.

the future location of a moving pedestrian or a car more efficiently. In a real-life application, after training the model, the system can be used in real-time monitoring. This means that, if a pedestrian (can be a child) loses control and enters traffic, which can result in an accident, the system can change the color of the red light. This would prevent an accident. The same can be applied to a car in motion.

The other feature that would be so useful in real life is to generate images (highlight) of possible locations an object / human in motion in a video. This would be useful in the surveillance system because it could warn of a possible human action to prevent, such as attacking a place that is protected by a surveillance system.

The other application would be simple to predict the location of a basketball player and predict the location in which the player can pass the ball in immediate future action.

This would be very useful for a basketball coach to predict and analyze the strategy of the opposing team and analyze each player individually from the opposing team. This would allow the coach to predict and change his team's strategy in real time. Since the system created in this work simply uses circle detection (head) with HCT, if applied in real time to various fields, such as basketball, it could change the way basketball coaches run their team.

In the following visualization, we can see that the red player starts the game, just after crossing the middle line, passes the ball with the blue player and after a while, the blue player shoots (the ball is green).

In a real life application, if the system is well trained, it could predict that the red player would usually pass the ball to his players on the left, and the blue player usually shoots right after receiving the ball, if he is on the radius . But even the implementation of this system in basketball would require a large amount of data for the players and a good recognition of the faces (to distinguish one player from the others).

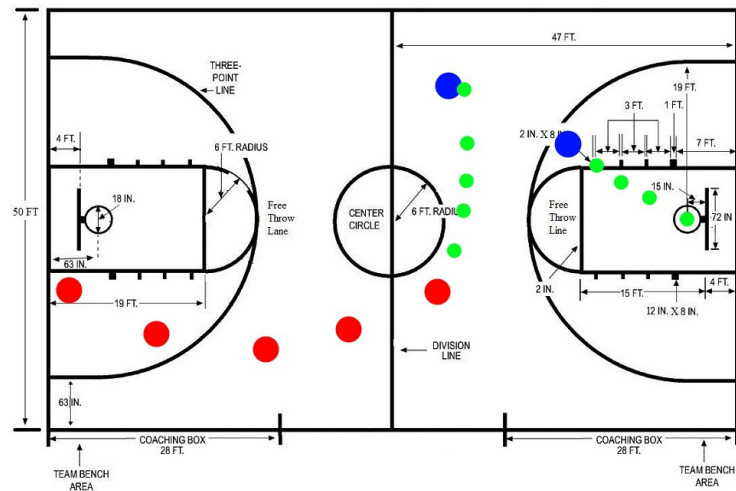


Figura6: Visualización simple de cómo el sistema vería a los jugadores y la cancha de baloncesto.

There are countless implementations of this system as long as the model is well trained and detection of objects are done well. It is important to note that this paper represents an initial step in the direction of prediction; we only predict positions assuming the rest of the scene is static. Possible future work includes modeling the simultaneous behavior of multiple elements, predicting their possible coordinated position.

4.1. Future work areas

Based on the results of this work, we can conclude some future works.

- Detecting the circles and human heads in a more efficient way using Hough Circle Transformations
- Generating data which is very close to real life data. This can be done by using videos taken for a long time.
- Improving the prediction of artificial neural network by modifying number of input data, number of total data
- Using other machine learning techniques such as Support Vector Machines to compare it with neural networks.
- Using other feature extraction techniques to compare them with Hough Circle Transformations

Referencias

1. Ali Ajdari Rad, Karim Faez, and Navid Qaragozlou. Fast circle detection using gradient pair vectors. pages 879–888, 01 2003.

2. G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
3. P. Joshi, D.M. Escriva, and V. Godoy. *OpenCV By Example*. Packt Publishing, 2016. ISBN 9781785287077. URL <https://books.google.es/books?id=WwYcDAAAQBAJ>.
4. Suping Wu and Xiangjiao Liu. Parallelization research of circle detection based on hough transform. 2012.

Análisis de Datos para el Mantenimiento Predictivo

Ricardo Martínez, Belén Pérez, Sara Rodríguez

Departamento de Informática y Automática, Facultad de Ciencias
Plaza de los Caídos s/n, 37008, Salamanca, España
{rimardo,lancho,srg}@usal.es
<http://diaweb.usal.es/>

Resumen *La competitividad en el entorno industrial y el surgimiento de nuevas tecnologías ha propiciado la creación de la Industria 4.0. Este concepto gira en torno a campos como el Internet of Things, Sistemas Ciberfísicos, Big Data y Machine Learning. Debido a que en la actualidad, la mayoría de las factorías basan sus sistemas predictivos de errores en datos a tiempo real, en este estudio se propone aplicar esta nueva tecnología para poder anticiparse a los fallos y actuar previo a que se cometa el problema. Para ello, los diferentes algoritmos más utilizados en industrias serán analizados y comparados para obtener la solución óptima. Finalmente, se presentará un caso de estudio con una base de datos industrial real donde se implementarán los diferentes algoritmos.*

Keywords: Industria 4.0, Big Data, Machine Learning, Conjuntos de datos industriales, Sistema predictivo, comparativa de algoritmos.

1. Introducción

En las industrias se han producido una gran cantidad de cambios para adaptarse al mundo moderno, para conseguir ser más competitivas ofreciendo productos más optimizados, baratos y eficientes a los clientes. Para ello, han debido modernizarse y adquirir tecnología más avanzada con el objetivo de mejorar sus procesos, tanto con una maquinaria más precisa como con una recogida de grandes cantidades de datos para realizar predictivos de fallos, reducción de tiempos de ciclo y mantenimientos más sencillos. Comenzaron a surgir, pues, las llamadas Industrias 4.0. Esta novedosa forma de producir tiene entre sus características la incorporación de IoT (Internet de las cosas) y el uso de sistemas ciber físicos [1], que trata de la simulación virtual de maquinaria para poder analizar su comportamiento y, además, se pueden realizar traspasos de información de esta unidad virtual al *hardware* real.

Actualmente, en la mayoría de las factorías se utilizan los sensores únicamente para conocer el correcto funcionamiento de la producción, midiendo diversas variables (temperatura, humedad, luz, pH, etc.) o para evitar colisiones de brazos robot con objetos del entorno, incluyendo personas, es decir, para detectar

obstáculos. Aunque en los últimos años se han comenzado a utilizar los sensores con otro fin: el de mejorar las industrias.

Un novedoso método para cumplir estas metas consiste en la masiva recolección de datos diversos, en lo que se conoce como *Big Data*, para crear grandes bancos y conseguir mucha información variada que pueda ayudar a los ingenieros. Así, el primer problema surge en cómo tratar estos bancos de datos masivos, procesarlos y obtener los resultados que se están buscando y actuar frente a ello. Es muy importante la inclusión de sistemas ciber físicos en este aspecto los cuales integran capacidades de almacenamiento, comunicación y computación, con el fin de interactuar y controlar la producción [2].

Una de las principales ventajas que aportan los grandes conjuntos de datos es la posibilidad de utilizarlos para conocer mejor cómo funciona un proceso, mejorar su funcionamiento y predecir cómo se va a comportar. Una de las técnicas utilizadas para ello es conocida como *Machine Learning* o aprendizaje automático. Se recurre, por medio de algoritmos, a la identificación de patrones complejos en el comportamiento de una situación determinada por medio de la observación de datos y con ello se tiene la capacidad de predecir comportamientos futuros [3]. Así, por ejemplo: (i) un banco podrá discernir, con esta tecnología, las probabilidades de que un cliente aleatorio se dé de baja y conocer qué factores influyen más en esta decisión, (ii) una compañía aérea podrá conocer la manera de optimizar los gastos en un vuelo prediciendo las condiciones óptimas, (iii) cualquier factoría de producción podrá establecer paros de emergencia durante el proceso cuando prevea que un error importante pueda ocurrir.

La aplicación de estos algoritmos se puede realizar con varias herramientas, según se adapten mejor o peor a cada situación determinada. Uno de los lenguajes de programación más utilizados para los procesos de *Machine Learning* es R, debido a su potencial con grandes librerías estadísticas y gráficas, además de paquetes que permiten un trabajo muy sencillo y a la inmensa cantidad de información, tanto académica como en foros que existe [4].

A lo largo de este estudio se analizarán distintos algoritmos de aprendizaje automático, y se planteará la elección de aquellos que sean más eficaces para determinadas situaciones en entornos industriales, enfocados para la realización de sistemas predictivos mediante el análisis de datos obtenidos por sensores, realizando posteriormente un caso de estudio práctico con un banco de datos concreto para evaluar la eficacia de estos algoritmos en la Industria 4.0.

La estructura del artículo se presentará de la siguiente forma:

- Para comenzar, se presentará **un estudio sobre el estado del arte** con respecto a la Industria 4.0, los grandes conjuntos de datos y el aprendizaje automático, así como los algoritmos de este campo aplicados en las industrias.

- Posteriormente, se **realizará un caso de estudio** con un conjunto de datos recogido por sensores reales en una factoría. Se escogerá el conjunto de datos más grande posible para optimizar los resultados.

- Tras ello, se mostrarán todos los **resultados que hayan sido obtenidos** para comprender de qué manera se tiene un impacto sobre el entorno industrial y se presentarán las conclusiones que se obtengan de estos.

2. Marco teórico

2.1. Industria 4.0

La Industria 4.0 se refiere a la cuarta revolución industrial, tras la mecanización, la manufactura en masa y la automatización. Los cambios radicales en la producción han generado impactos en la sociedad desde la aparición de la máquina de vapor en el siglo XVIII, aplicación que permitía, por primera vez, producir más cantidad en menor tiempo por individuo trabajando, pero la revolución que inició la industria tal y como se conoce ahora fue la segunda, alrededor de la década de los 70 del siglo XIX, sobre todo de la mano del empresario Henry Ford, que ideó la cadena de producción, en la cual los trabajadores se especializaban en una tarea para realizar productos en masa además de resultados más perfectos y menos costos. Esto también fue posible gracias a la llegada de la electricidad.

Posteriormente, la que se conocía hasta ahora como última revolución industrial, aunque más silenciosa que las anteriores, fue la aparición de la automatización monitorizada por sistemas de control y el seguimiento de la producción mediante históricos de datos. Desde hace unos años, comenzó a surgir una nueva denominación, la Industria 4.0, como una nueva forma de producir revolucionaria, con la aparición de sistemas ciber físicos y el Internet de las cosas con roles trascendentales en esta [5]. Este desarrollo industrial se observa claramente esquematizado en la Figura 1 [6].

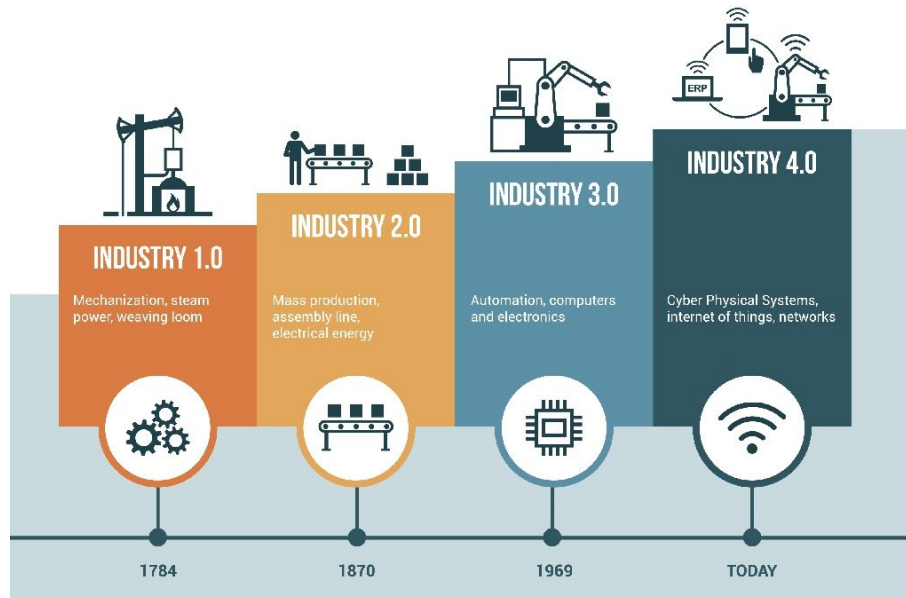


Figura 1: Revoluciones industriales [6]

El IoT o Internet de las cosas ha llegado a ser parte inamovible de estas nuevas factorías. Las Industrias 4.0 tienen el potencial de crear extraordinarios crecimientos productivos y ventajas competitivas [7]. En este tipo de fábricas, los sistemas de control comparten información entre ellos, y estos a su vez aplican los cambios necesarios, en función de estos datos, envían ordenes a máquinas esclavas correspondientes para modificar el proceso de producción, por ejemplo, para evitar paros, disminuir tiempos de ciclo, detectar imperfecciones en una pieza, etc. La idea es que estos sistemas también sean capaces de trabajar individualmente, como se ha hecho hasta ahora, ante posibles fallos de conectividad.

Otras características que marcan la diferencia de esta nueva revolución con respecto las anteriores:

- **Robots autónomos:** entre los cuales se pueden encontrar los robots colaborativos, de trascendental importancia que muchas empresas desean implantar, gracias a su adaptabilidad en cualquier entorno industrial y su papel de ayudantes de operarios.

- **Fabricación aditiva:** que consiste en la adición capa por capa de un material para generar una estructura 3D desde un archivo. Es más conocida como impresión 3D, y algunas empresas comienzan a utilizar esta técnica para producir piezas específicas necesarias para su producto final y así evitar el costo mayor que suponen los proveedores actuales [8].

- **Realidad aumentada:** aunque menos utilizada y más centrada en compañías tecnológicas, es una tecnología que pueden subcontratar las empresas productoras para, por ejemplo, ver el acabado final de un producto u observar el diseño completo de una posible línea de producción y poder tener así una idea más clara.

- **Seguridad informática:** centrada en la protección de infraestructura computacional, especialmente la información contenida en servidores. Es un avance que ha sido necesario debido al uso de tecnologías cada vez más avanzadas que utilizan servidores para almacenar todo tipo de información confidencial sobre, por ejemplo, un producto [9].

2.2. Big Data y Machine Learning

Big Data es un término que se refiere a conjuntos de datos de grandes dimensiones. Sus características principales son [9]:

- **Volumen:** Hay más datos que se transmiten por Internet cada segundo de los que existen en esta red almacenados hace 20 años. No solo se trabaja con Internet, ya que desde Intranets también surgen infinidad de datos que recopilar y analizar, del orden de petabytes cada hora en grandes empresas.

- **Velocidad:** bien es cierto que, para una gran cantidad de compañías, es más importante la velocidad a la que se crean, reciben y tratan estos datos que, a su cantidad en sí, es decir, su capacidad para trabajar en tiempo real. Hay empresas que utilizan la ubicación de personas para conocer, por ejemplo, como en días de rebajas la gente consume en multinacionales y cómo afecta esto a minoristas. Por supuesto esta característica influye mucho en profesiones determinadas como, por ejemplo, corredores de bolsa.

• **Variedad:** como se comentaba antes, una analítica tradicional de pequeños negocios, ofrece información sobre qué productos se venden mejor o qué proveedores son más eficientes. Pero *Big Data* va más allá. Ofrece la posibilidad de detalles minuciosos, por ejemplo, ¿qué elementos hacen que esos libros sean más vendidos? Tal vez la cantidad de imágenes, su tema, edición, autor, etc.

Con respecto a *Machine Learning*, se considera un campo de la inteligencia artificial cuya meta es permitir que las computadoras "aprendan". Los distintos algoritmos de aprendizaje automático se clasifican en función de la salida de estos, y se pueden agrupar en algoritmos de aprendizaje supervisado y no supervisado.

En el aprendizaje supervisado se cuenta con un conjunto de ejemplos de los cuales se conoce su respuesta o etiqueta. Lo que se busca es formular algún tipo de correspondencia que permita aproximar la respuesta para todos los objetos que se presenten. Así, se podría decir que este algoritmo trabaja como un histórico de información y aprende a poner las etiquetas de salida adecuadas a un nuevo valor. Un ejemplo sencillo de este estilo son los correos basura o *spam*. En función de unos parámetros de entrada, como por ejemplo si hay una lista de remitentes o el mensaje tiene unos términos determinados de marketing, y conociendo si se trata de correo basura o no, el algoritmo va aprendiendo. Una vez ha aprendido, al trabajar con otros correos no etiquetados predecirá si es *spam* o no. Este tipo de algoritmos se suele utilizar para problemas de clasificación (variable tipo categórica) y para problemas de regresión (variable tipo numérica). Entre los más utilizados se encuentran:

- Clasificador bayesiano
- Regresión logística
- Regresión por mínimos cuadrados
- Árboles de decisiones
- SVM

Por el contrario, el aprendizaje no supervisado se utiliza cuando no se dispone de datos etiquetados. Por eso se dicen que tiene un carácter exploratorio, porque solo se puede describir la estructura de los datos para tratar de encontrar algún tipo de organización. Haciendo esto, es posible agrupar los datos que tienen correlaciones entre ellos, si bien es cierto que puede que al final no tenga ninguna utilidad o resultado lógico. Los algoritmos más habituales son:

- *KMeans*
- Análisis de componentes principales (PCA)
- ICA

2.3. Algoritmos aplicados en industrias

En este apartado se procederá a estudiar la posible aplicación de varios algoritmos de aprendizaje automático que se pueden implementar para la creación de un predictivo en conjuntos de datos industriales. Como es de prever, no parece fácil que exista un algoritmo determinado que sea perfecto y extrapolable a

cualquier ámbito. Lo que sí que se puede pensar es que, teniendo en cuenta unas características comunes de los conjuntos de datos industriales (grandes dimensiones, valores constantes, falta de datos, estructura, etc.) e investigando a partir de diferentes estudios, se puede encaminar a elegir procedimientos determinados y a descartar algoritmos que no satisfagan el procesamiento de las características nombradas. Se procederá a trabajar teniendo en cuenta lo que suele caracterizar a estos bancos de datos [2]. Por ello, se descartarán los algoritmos que funcionen bien para poca información pero bajen rendimiento y tengan demasiado coste computacional con grandes conjuntos. Se cubrirán algoritmos para imputación de datos ausentes, para equilibrios de conjuntos de datos, selección de características más importantes y para entrenamiento de modelos predictivos.

Algoritmos de imputación de datos

KNN *K Nearest Neighbours* (KNN) es un algoritmo muy conocido y usado tanto para la clasificación como para la regresión en un espacio multidimensional. Puede ser utilizado para datos numéricos, discretos, y categóricos. Pero es un algoritmo que también puede cumplir otra función, como es el caso de tratar con los datos incompletos. La idea de utilizar KNN para este propósito es que un valor puntual puede ser calculado mediante valores en puntos que son próximos a él, basados en otras variables. Aunque ahora se esté nombrando para describir la imputación de valores, también es necesario explicar su funcionamiento como clasificador y algoritmo de regresión. Este algoritmo además es usado en industrias para la predicción de detección de fallos aunque los datos no estén etiquetados [10].

Con respecto al algoritmo de clasificación, si se desea clasificar x_x , si se tiene x_1, \dots, x_k los k vecinos más próximos a x_x se han de regresar (1):

$$\hat{f}(x) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^k \delta(v, f(x_i)) \quad (1)$$

donde $\delta(a,b) = 1$ si $a=b$ y 0 en otro caso.

El valor que se obtiene es, pues, un estimador del valor de la función del punto que se desea clasificar, que adquiere el valor de sus vecinos más cercanos. Por ello, si $k=1$, como se ha comentado, se adquirirá el valor del vecino más próximo.

MICE La metodología de este algoritmo es que la imputación múltiple es más eficiente realizada con pequeños pasos que requieren una comprobación de diagnóstico cada vez que se ejecuten. En notación matemática la ideal del algoritmo se expresa de la siguiente manera:

Se tiene el vector de variables Y_j y ($j = 1, \dots, p$) siendo p una de las variables faltantes, donde $Y = (Y_1, \dots, Y_p)$. Las partes observadas y faltantes de Y se denotan de la manera de Y_j^{obs} y Y_j^{miss} respectivamente, así que $Y^{obs} = Y_1^{obs}, \dots, Y_p^{obs}$ y

$Y^{miss} = Y_1^{miss}, \dots, Y_p^{miss}$ se refieren a los datos presentes y faltantes en Y . El número de imputaciones es igual a $m \geq 1$. El n -ésimo dato imputado es denotado como Y^n donde $n = 1, \dots, m$. Se denota como $Y_{-j} = Y_1, \dots, Y_{j-1}, \dots, Y_{j+1}, \dots, Y_p$ la colección de $p-1$ variables en Y excepto Y_j . Se considera Q como la cantidad de interés científico, como, por ejemplo, un coeficiente de regresión. En la práctica, Q es normalmente un vector multivariable.

Miss Forest Es un algoritmo de imputación de datos que utiliza el método de aprendizaje por conjuntos *Random Forest* para sustituir los valores faltantes. *Random Forest* sirve como algoritmo de clasificación y regresión, entre otros, que opera mediante la construcción de árboles de decisión durante el entrenamiento para obtener la clase o la media de la predicción como resultado.

El algoritmo de entrenamiento de *Random Forest* aplica la técnica de agregación de *bootstrap* donde se utiliza un conjunto de entrenamiento dado $X = x_1, \dots, x_n$ cuyas salidas son $Y = y_1, \dots, y_n$. Se 'enmaquetan' repetidas veces (B veces) muestras aleatorias para ajustar los árboles de decisiones. Para $b = 1, \dots, B$:

- Muestrear, con reemplazamiento, B muestras de entrenamiento de X, Y llamadas X_b, Y_b .
- Entrena un árbol de regresión o clasificación f_b en X_b, Y_b .

Después del entrenamiento, la predicción para muestras faltantes x' se puede realizar mediante una media de las predicciones de todos los árboles de regresión en x' según la fórmula 2:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (2)$$

En la figura 2 se observa una gráfica con dos *datasets* grandes y de nuevo el error que existe comparando los valores sustituidos por los reales. Se observa como MICE pierde precisión enormemente. El algoritmo KNN suele ofrecer un resultado medio bueno mientras que utilizar la media de los datos de las columnas para la imputación de valores ofrece los peores resultados [11].

También se puede observar en la figura 3 [11] el tiempo de computación de estos algoritmos según lo grande que sea el conjunto de datos y, como se observa, los algoritmos FKN y MICE tienen un alto coste computacional cuando el conjunto de datos aumenta en tamaño, mientras que el resto se mantiene con un coste computacional bajo.

2.4. Algoritmos de equilibrio de conjuntos de datos

Cuando en un conjunto de datos una clase supera en número enormemente a otra existe un desequilibrio. Este término de desequilibrio se refiere a la variable de salida resultante, que es la que se busca predecir en este caso [11]. Esto es algo

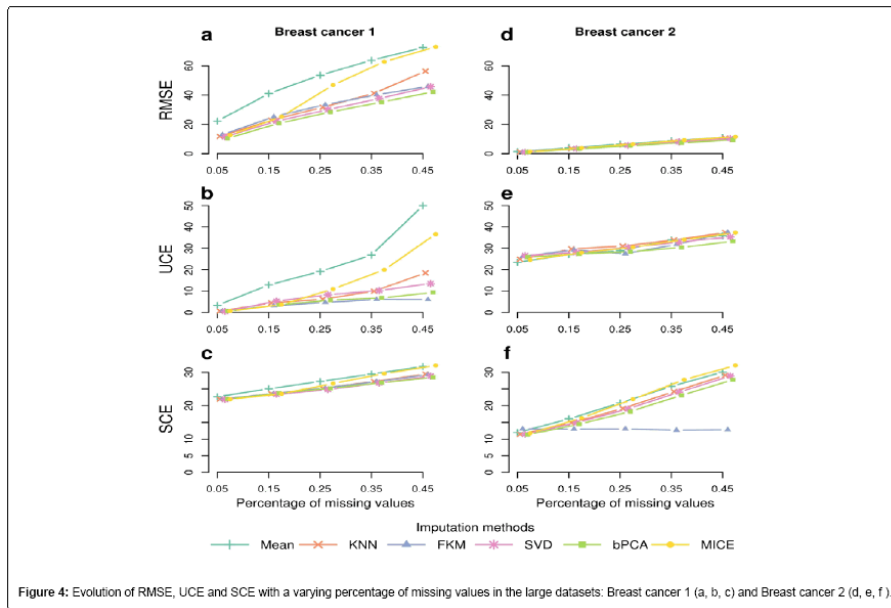


Figura 2: Rendimiento de diferentes algoritmos para conjuntos de datos grande [11]

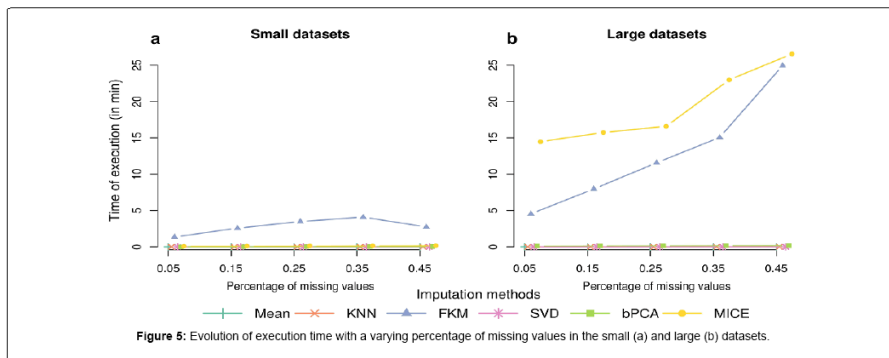


Figura 3: Tiempo de computación de los algoritmos [11]

que se produce mucho en industrias, ya que cuando se inspecciona un producto acabado, se suele encontrar muchos productos sin defectos que con defectos. Existen diferentes maneras de tratar con *datasets* desequilibrados: sobremuestreo, inframuestreo, generación de valores sintéticos o aprendizaje "sensato". El primero de los métodos supone una pérdida de información, el segundo puede

implicar overfitting, ya que se generan réplicas de datos, el tercero genera datos artificiales y el cuarto evalúa el coste asociado a las observaciones incorrectamente clasificadas.

Existe un algoritmo denominado Smote que genera datos artificiales de la siguiente manera:

- Se recoge la diferencia entre un punto determinado y su k vecino más cercano.
- Se multiplica esa diferencia por un número aleatorio entre 0 y 1.
- Lo añade al vector de características bajo consideración.
- Se selecciona así un punto en vectores de características.

Estos procedimientos suelen abarcar gran parte del preprocesamiento en los conjuntos de datos industriales [12], aunque es evidente que pueden existir también conjuntos de datos categóricos o que presenten alguna diferencia. Existe otro algoritmo, como es Rose, que utiliza el mismo principio que Smote con la diferencia de que en vez de crear muestras en vectores, las crea en el espacio de características de la clase. Se compararán ambos algoritmos posteriormente. Aunque se han comentado los algoritmos que se creen más relevantes y que han sido más citados en las fuentes de referencia que se han indicado, existen otros algoritmos como los de discretización, generación de instancias, transformación del espacio, etc. que permiten realizar un correcto preprocesamiento para elaborar un buen modelo posterior.

2.5. Selección de KPIVs

A continuación, es necesario realizar una selección de las características (KPIVs) más relevantes del conjunto de datos. Por relevantes se entiende aquellos atributos que afectan en mayor medida a la clase, es decir, al *output*. Para entenderlo perfectamente, tal y como se ha descrito antes, los atributos con valores constantes se han de eliminar, ya que son redundantes, porque al no cambiar, no van a afectar a la salida del proceso. Mientras que pueden existir características que sí afecten y para elaborar un posterior modelo adecuado, se han de elegir aquellas que tengan más peso en el resultado final.

Existen diferentes maneras de seleccionar los KPIVs, como la correlación entre atributos y la clase, es decir, la relación que existe entre las características y el output en un intervalo de -1 (inverso) a 1 (proporcional), donde 0 es nula correlación. La regresión también permite establecer mediante una fórmula cómo están relacionadas dos variables y cuantificarlo. Existen también algoritmos diseñados que utilizan estas técnicas para determinar con mayor precisión cuáles son los KPIVs.

Lasso (por sus siglas en inglés *Least Absolute Shrinkage and Selection Operator*) es un método de regresión que permite la selección de características con más peso en la salida. Este procedimiento ofrece buenos resultados de precisión y es fácilmente interpretable. Además, es adecuado para evitar el sobre aprendizaje. Este método se basa en la idea de los mínimos cuadrados.

La idea es penalizar el coeficiente beta de la fórmula 8 para conseguir los atributos más influyentes. Para ello, en esta misma fórmula el coeficiente λ es

un penalizador que reduce el coeficiente de correlación de las características, mientras que α se selecciona según el modelo que se deseé construir.

$$\min(\beta_0, \beta) = \frac{1}{N} \sum_{i=1}^N \omega_i l(y_i, \beta_0 + \beta^T x_i) + \lambda[(1 - \alpha) \|\beta\|_2^2 / 2 + \alpha \|\beta\|_1] \quad (3)$$

En esta fórmula, y_i es la salida, N es el número de casos que consisten en covariar cada una de las salidas y ω es el vector de covarianzas. Escogiendo un valor correcto para λ , se podrán seleccionar las características mínimas que más correlación tienen con la clase sin perder información.

2.6. Entrenamiento del Modelo

Una vez seleccionadas las características que tienen más peso en la salida es hora de elegir y entrenar el modelo que permita predecir cómo los valores obtenidos por sensores industriales afectan a la clase del producto. Entre diversos métodos, se pueden aplicar algoritmos de regresión logística para predecir el resultado de una variable categórica en función de las variables predictoras. Así se puede conseguir la probabilidad de que un suceso se produzca en función de cada atributo en mayor o menor medida. A modo de comparación, también se utilizará el algoritmo KNN para crear otro modelo de predicción y conocer así cual funciona mejor.

2.7. Métodos de evaluación

Para verificar que un modelo predictor creado es correcto, este se ha de someter a varios procesos de evaluación. Para ello, es necesario disponer de un conjunto de datos diferente al utilizado para el entrenamiento del algoritmo. Como resultado se pueden aplicar varias fórmulas estadísticas o herramientas como la curva ROC, que indica el ratio de variables correcta o erróneamente predichas. Las fórmulas de evaluación que se aplicarán en este trabajo recogerán datos de una matriz de confusión creada después de realizar pruebas con el conjunto de test. Se recogerán los datos de valores predichos correcta e incorrectamente. Para ello, se atenderá a la tabla 1.

Matriz de confusión		Clase Predicha	
		Correcto	Fallo
Clase Real	Correcto	TP	FN
Real	Fallo	FP	TN

Tabla 1: Matriz de confusión

Donde TP son correctos verdaderos, FP correctos falsos, TN fallos verdaderos y FN fallos falsos. Las formulas de evaluación a aplicar serán las siguientes:

$$RatioTP = \frac{TP}{TP + FN} \quad (4)$$

$$RatioFP = \frac{FP}{FP + TN} \quad (5)$$

Los ratios de las fórmulas 4 y 5 indican la cantidad de las clases correctas predichas correctamente con respecto al total de clases verdaderas totales en el caso de la primera fórmula, mientras que en el segundo caso se calcula el ratio de falsas clases correctas predichas con respecto al total de fallos reales existentes. La diferencia entre estos dos ratios permite obtener información sobre los modelos que predicen más clases verdaderas correctas en comparación con los positivos incorrectamente clasificados. Cuanto mayor sea esta diferencia, mejor será el sistema predictivo.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

La precisión o sensibilidad del sistema se podría calcular mediante la fórmula 6, donde se utiliza el valor de clases verdaderas correctamente clasificadas entre el total de atributos verdaderos. Otro parámetro denominado Valor F, que consiste en medir la precisión de un test mediante la determinación de un valor único ponderado de la precisión y la exhaustividad, se puede obtener mediante la ecuación 7.

$$ValorF = \frac{2TP}{2TP + FP + FN} \quad (7)$$

Finalmente, también se podría utilizar como evaluador el coeficiente de correlación de Matthews, que consiste en una medida de asociación entre dos variables binarias. Su interpretación es parecida al coeficiente de Pearson. Cuanto mayor sea el valor obtenido de la fórmula 8, mayor correlación existirá entre los atributos seleccionados y la salida, con lo que se comprobará un indicio de buen resultado.

$$CCM = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (8)$$

3. Caso de estudio

Este proyecto busca aplicar todos los algoritmos explicados previamente en conjuntos de datos industriales para conocer cuáles ofrecen mejores resultados con el objetivo de crear un sistema predictivo de errores. Estos conjuntos pueden tener diferentes características, desde datos numéricos hasta categóricos, tamaño, cantidad de ruido, etc. Se pretende escoger un conjunto de datos que sea el más abundante dentro de las industrias, esto es, de gran tamaño (una gran cantidad de atributos con múltiples observaciones), numérica y que no esté preprocesado. De esta manera, se consigue así realizar un desarrollo completo que más tarde pueda aplicarse a casos reales.

Se ha realizado una búsqueda en profundidad para determinar qué conjunto de datos puede resultar óptimo para la creación de un predictivo en una industria que sea lo más fiel posible a la realidad. Por ello, se ha descartado escoger conjuntos de datos que hagan referencia a predecir la vida de un producto ya acabado en base a cierta información. Lo que se busca es que mientras una factoría esté activa, se puedan predecir fallos mediante algoritmos de aprendizaje adecuados con la posibilidad de enviar mensajes de alerta a los ingenieros u operarios.

A partir de esto, se ha de diferenciar entre dos tipos de conjuntos de datos industriales. El primer tipo sería el que gira en torno al producto, es decir, tener la capacidad de conocer si el producto se está fabricando correctamente o no, qué factores son los que más relevancia tienen a la hora de que se produzcan estos fallos, capacidad de corregir el funcionamiento de las máquinas enviando una orden a esta cuando se cumplan ciertas condiciones, etc.

Por contra, también se podría abordar la situación de otra manera, y es trabajando con la maquinaria en sí, es decir, los datos que recoge mientras está en funcionamiento y si está cambiando en algo su comportamiento a lo largo del tiempo. De esta manera, se podrían predecir fallos, por ejemplo, si se detectase que un robot se encuentra fuera de sus condiciones normales, de si tarda más en completar su tiempo de ciclo, etc. De esta manera, se abordaría el problema con respecto a las herramientas de producción.

En general, la información de las industrias es confidencial o demasiado específica para que se encuentre en acceso libre. Se han encontrado conjuntos de datos de ambos tipos, aunque del segundo tipo son muy escasos, con tamaños difíciles de manejar o con falta de información sobre el resto del entorno. Por ello, se ha decidido trabajar con un banco de datos de producción de semiconductores y sobre cómo reconocer qué es lo que falla y cómo hacerlo saber a empleados.

3.1. Conjunto de datos SECOM

Para el propósito explicado en la sección anterior, se ha decidido escoger el conjunto de datos SECOM, extraído de UCI *Machine Learning* Repository que trata sobre la fabricación de semiconductores, en especial del campo de la electrónica, muy ligado al de la informática, y que es trascendental actualmente. Todos los dispositivos que se utilizan diariamente y que la tecnología facilita al

ser humano contienen este tipo de componentes. Como cualquier otro proceso industrial, y tal y como se ha comentado en apartados previos, la producción de semiconductores está monitorizada por numerosos sensores, que recogen gran cantidad de datos, tanto para enviar a controladores y actuadores como para recopilar información para los históricos.

Las diversas señales que se miden durante la manufactura, pueden incluir ruido o información irrelevante. El primero se ha de filtrar en la etapa de pre-procesamiento y la segunda se ha de reconocer a la hora de saber qué factores tienen mayor peso durante la producción. Si bien es cierto que, en ocasiones, lo que se cree como información irrelevante puede que tenga un significado importante en algún aspecto, por eso ha de valorarse también. La selección de características pues, puede utilizarse para conseguir la información necesaria.

El conjunto de datos seleccionado para el presente estudio contiene información sobre la fabricación de semiconductores, incluyendo características varias sobre su elaboración y condiciones de fabricación. Consiste en un conjunto multivariable y real. También se posee otro conjunto de datos etiquetados, con la información sobre si la unidad producida es correcta (-1) o si tiene algún fallo (1) y su fecha correspondiente. Se pretende conocer qué características tienen un impacto mayor en la eficacia de producción. El número de instancias es de 1567 y los atributos son 591 (no son especificados). Contiene valores inexistentes. Los datos ya han sido estandarizados y las características constantes eliminadas [12].

3.2. Creación del modelo

La creación del modelo será llevada a cabo siguiendo los esquemas de las figuras 4 y 5.

En primer lugar se recogerán los datos desde dos ficheros diferentes, se reorganizarán en una matriz y se les atribuirá nombres a las columnas. Habiendo creado ya la matriz, se eliminarán los valores constantes, los atributos que tengan más de cierto porcentaje de valores ausentes y características irrelevantes como la fecha y hora de producción de los semiconductores. Posteriormente se normalizarán entre 0 y 1 las instancias que queden.

En segundo lugar, se realizará una imputación de datos con técnicas como K Nearest Neighbours, MICE y MissForests que asignan valores plausibles en estos casos. Tras esto se dividirá el conjunto en dos *sets* de entrenamiento y test. Se creará ya un primer modelo predictivo de prueba usando KNN y regresión logística.

Tras esto, se realizará un equilibrio de datos, probando dos algoritmos (Smote y Rose) para compensar la falta de instancias de producción de semiconductores fallidos. Se volverá a aplicar KNN y regresión para elaborar el modelo. Finalmente, también se hará una selección de características mediante la regresión de Lasso para trabajar únicamente con los algoritmos que más influencia tienen en la salida. Se esta manera se consigue centrarse mucho más en la solución del problema y descartar información irrelevante. Se vuelve a construir un último modelo con KNN y regresión.

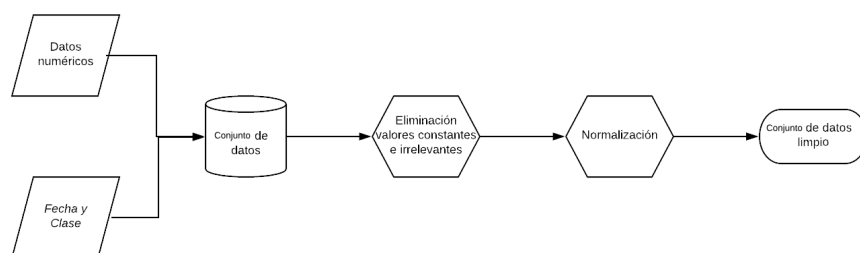


Figura 4: Proceso de preparación de datos

Se obtendrán un total de 36 resultados diferentes los cuales se han de analizar y evaluar con los métodos indicados en el estado del arte, obteniendo un total de 36 resultados por método evaluador, lo que hacen un total de 180 datos para dilucidar qué modelos son los más eficientes para este caso y consultar si se pueden extrapolar a otras líneas de producción.

Por último, se contemplará la posibilidad de que, dadas ciertas condiciones, se envíen señales a la maquinaria para que corrija su comportamiento o bien que se envíen alertas o mensajes a la línea de producción.

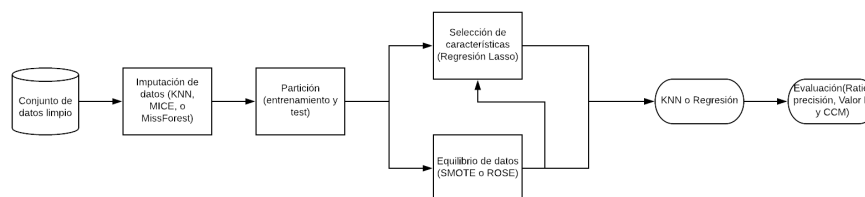


Figura 5: Obtención y evaluación del modelo

4. Resultados

4.1. Ratios, precisión, Valor F y CCM

Evaluación mediante ratio de verdaderos positivos El primer ratio a analizar es el ratio de verdaderos positivos, cuya fórmula se ha explicado en el apartado 3. Este ratio analiza toda la clase positiva real que el predictor ha clasificado correctamente, verdadero positivo (TP por sus siglas en inglés) e incorrectamente, que serían los falsos negativos (FN).

Aplicando esta fórmula para todas las combinaciones de algoritmos explicados, se puede concluir que el mejor resultado se obtiene imputando datos mediante MissForest, equilibrando los datos con el algoritmo Rose y realizando una selección de características. Realizando un modelo predictor con KNN se obtiene un ratio de verdaderos positivos del 100 %, con lo que la clase de productos correctos ha sido bien clasificada en su totalidad como puede verse en la figura 6. Los siguientes en arrojar mejores resultados son la imputación de valores por medio de KNN y el equilibrio mediante Rose: la selección o no de KPIVs es indiferente ya que el ratio es idéntico. El modelo predictor también es KNN.

La regresión logística ofrece en todos los casos peores resultados. El procedimiento que mejor ratio ofrece si se contempla la suma de ratios de KNN y regresión es aplicar únicamente MissForest como algoritmo de imputación y realizar el modelo de predicción con esos datos. Como se puede observar, la gráfica 6 ha sido ordenada de mejores a peores ratios en conjunto para tener una visión más clara de qué algoritmos funcionan mejor.

Evaluación mediante ratio de falsos positivos Se analizará la cantidad de clases positivas predichas incorrectamente con respecto a la totalidad de clases negativas reales, FP y TN respectivamente por sus siglas en inglés.

El resultado se puede apreciar en la gráfica 7. Esta vez se ha decidido ordenar a la inversa, ya que cuanto menor sea el ratio de positivos falsos, quiere decir que habrá clasificado mejor la clase negativa. Así pues, se observa que existen hasta 5 procedimientos que adquieren un valor de 0, siempre realizando el modelo predictor con KNN. Si se comparan en conjunto, aplicar KNN para imputación de datos y equilibrar el conjunto con Smote la combinación que mejor solución ofrece, seguido de aplicar KNN con selección de características y Miss Forest con Smote, cuyos resultados son idénticos.

Se podría decir que, si un procedimiento tiene un ratio de TP muy elevado, pero aún así su ratio de FP también es muy elevado, se está ante un resultado inadecuado ya que no ha clasificado casi valores negativos correctamente, con lo que siempre fallará a la hora de predecir esta clase y no se podrá conocer si existe realmente un fallo en el proceso, que es lo más trascendental en un sistema predictivo.

Evaluación mediante precisión/sensibilidad La precisión o sensibilidad se realiza teniendo en cuenta la cantidad de positivos verdaderos con respecto al total de positivos predichos, es decir, la suma de positivos falsos y positivos verdaderos. Es una manera de conocer el total de predicciones positivas que el modelo ha clasificado correctamente con respecto a las que no. También sería interesante realizarlo con las clases negativas, ya que así se tendría un indicio de si el modelo detecta correctamente cuándo va a suceder un error.

Analizando la figura 8, se puede afirmar que la sensibilidad con respecto a la clase positiva tiene resultados satisfactorios en todos y cada uno de los casos, siendo el que mejor resultado arroja la imputación de datos con KNN y el

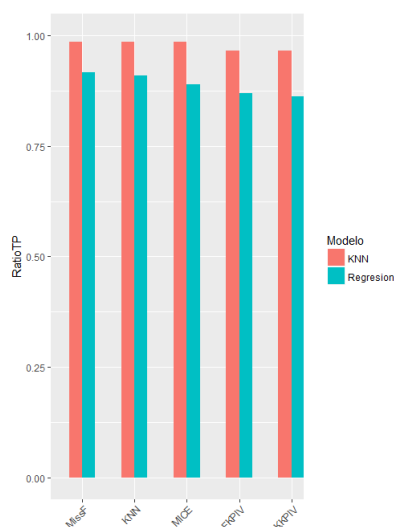


Figura 6: Ratio TP

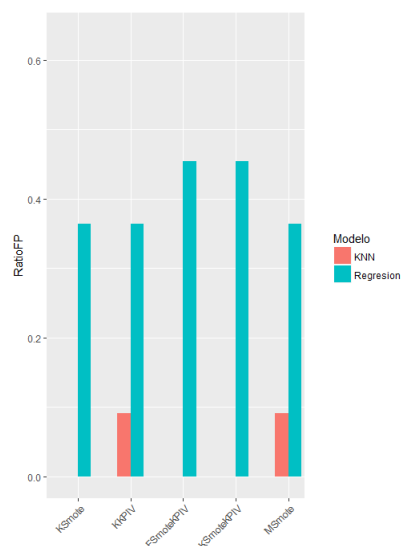


Figura 7: Ratio FP

equilibrio de datos con el algoritmo Smote, seguido de imputar datos mediante KNN con selección de características y, de nuevo, imputar con KNN, equilibrar mediante Smote y seleccionar KPIVs.

Evaluación mediante valor F El siguiente método de evaluación es el Valor F, que tal y como se ha comentado, es una herramienta que también mide la precisión de un modelo de predicción. Para ello, de la matriz de confusión se obtiene los positivos verdaderos, los positivos falsos y los negativos falsos y se trabaja con estos valores. En realidad lo que se analiza es la cantidad de positivos verdaderos con respecto a la suma de la clase positiva real y la clase positiva predicha en su conjunto.

En esta ocasión, el Valor F más elevado se consigue utilizando MissForest para imputar datos, se equilibran los datos con Rose y se aplica la regresión de Lasso para seleccionar las características más importantes, como se aprecia en la gráfica 9. Los siguientes mejores resultados se consiguen de la misma manera que en el método anterior, pero utilizando MICE o MissForest para sustituir valores faltantes. El modelo de regresión que mejor resultado consigue se obtiene únicamente aplicando KNN para imputación o bien utilizando MissForest.

Evaluación mediante el Coeficiente de Correlación de Matthews Finalmente, se aplicará el coeficiente de correlación de Matthews. Este coeficiente evalúa la fuerza de la relación de los atributos que se han contemplado con respecto a los valores del atributo clase, teniendo en cuenta los procedimientos que se han seguido para crear el modelo predictivo.

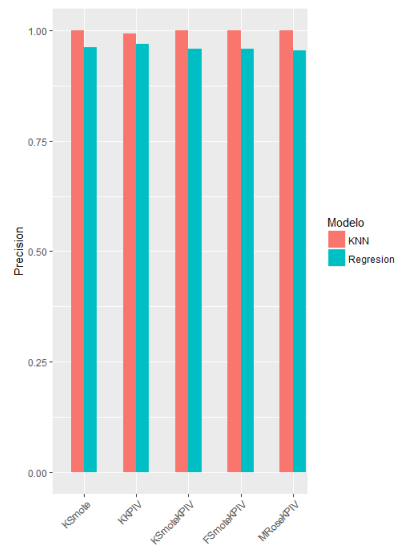


Figura 8: Precisión

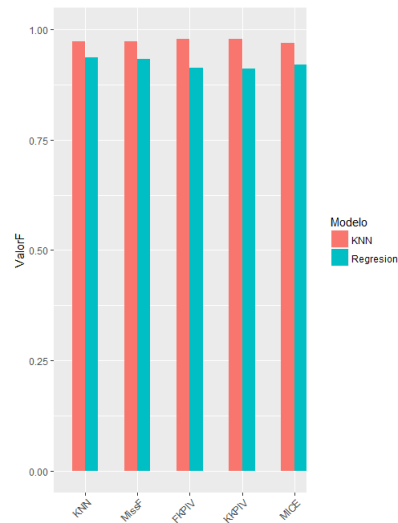


Figura 9: Valor F

En esta ocasión, estudiando la gráfica 10, se ve que el modelo creado a partir de regresión logística suele ofrecer resultados de poca correlación. También se aprecia, verificando así que se han obtenido resultados lógicos, que los procedimientos que mayor coeficiente de correlación de Matthews tienen son aquellos en los que se ha realizado una selección de atributos. Esto quiere decir que la selección es correcta, aunque en el resto de resultados no ha sido la mejor opción escoger KPIVs, porque puede que se haya perdido información al descartar ciertos atributos. Se puede concluir que los mejores resultados que se han obtenido son:

- Aplicar MissForest equilibrando datos con Rose y selección de características.
- KNN para la imputación de datos equilibrando con Rose y seleccionando características.
- MissForest con Smote y selección de KPIVs.

Visualización Ratio TP vs Ratio FP Con el objetivo de tener una idea más clara que añade más consistencia a los resultados obtenidos anteriormente, se han creado gráficas de líneas donde se contemplan el ratio de positivos verdaderos versus positivos falsos para cada procedimiento. La gráfica 11 hace referencia al modelo predictor que usa KNN mientras que la gráfica 12 analiza el comportamiento de regresión logística.

Se ha de buscar y comparar la diferencia entre los puntos de cada procedimiento entre los dos ratios. Cuanto mayor sea esta diferencia mejor resultado ofrece, ya que se han clasificado mejor los valores positivos y negativos. En el

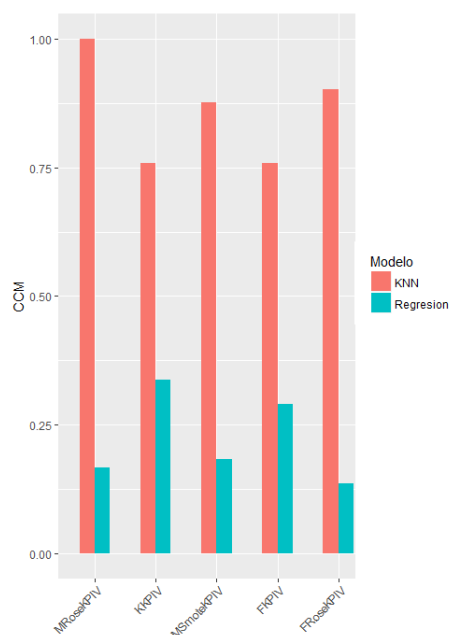


Figura 10: CCM

caso del modelo predictivo de KNN, se puede ver cómo el uso de MissForest para imputación de valores y de Rose para equilibrar datos tiene el resultado más alto de ratio TP y a su vez el más bajo de ratio FP. Se puede observar cómo equilibrar datos con Smote y seleccionar KPIVS, indierentemente del método utilizado para sustituir valores faltantes, también ofrece una buena solución. Si se observa un modelo que tiene un alto ratio de TP, puede que también tenga un alto ratio de FP, lo que hace que se detecten pocos fallos reales y lo haga un modelo incorrecto, como es el caso de KNN con Rose o todos los procedimientos que únicamente construyen el modelo con imputación de datos.

En el caso de utilizar regresión logística para el sistema predictivo, los resultados se mantienen en proporción para cada uno de los procedimientos nombrados. Así, KNN para sustitución de datos y Rose para equilibrio, vuelve a repetir como resultado menos óptimo, ya que no solo tiene un ratio FP elevado, sino que tiene el ratio TP más bajo de todos, aunque también se podría recurrir a MissForest con selección de KPIVs.

Si se pretende obtener las mejores soluciones, habría que recurrir a KNN con selección de las características más importantes, ya que tiene el menor ratio de FP con lo que detecta muy bien las unidades erróneas y también acierta a la hora de clasificar las clases positivas. Los modelos que solo contemplan imputación de datos, aunque tienen ratios TP elevados, también repiten teniendo ratios FP elevados, con lo que no clasificarán correctamente la clase negativa.

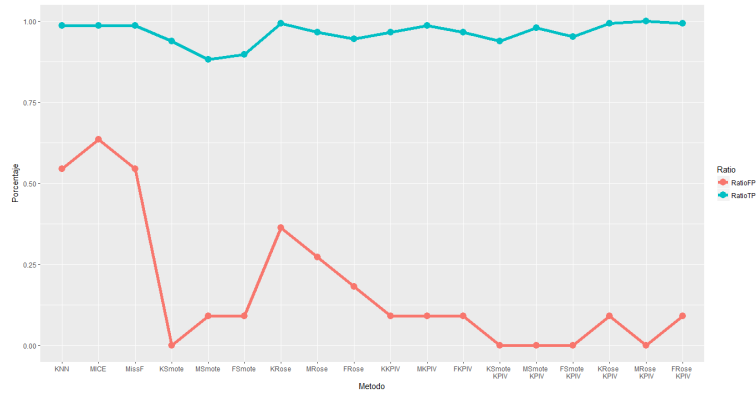


Figura 11: TP vs FP en el modelo de KNN

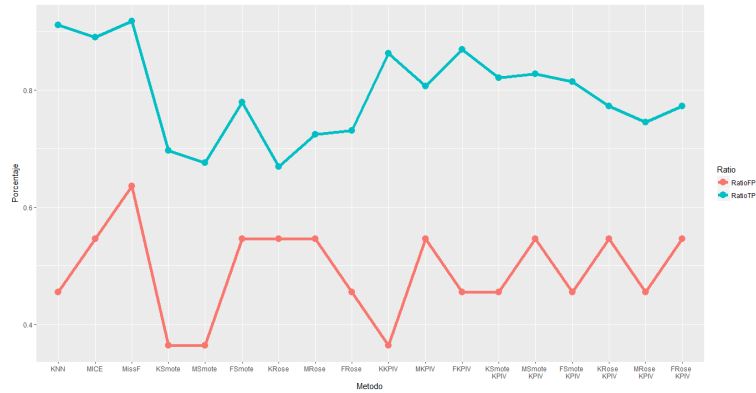


Figura 12: TP vs FP en el modelo de regresión logística

5. Conclusiones y líneas de trabajo futuras

Durante la elaboración de este trabajo de fin de máster se ha realizado una investigación en el contexto de la Industria 4.0, una revisión del estado del arte y su funcionamiento, con el fin de crear un sistema predictivo de errores a partir de un conjunto de datos industrial real. Dentro de este campo se ha expuesto cómo el *Big Data* y el *Machine Learning* desempeñan un papel fundamental en este sector.

Para la creación del sistema predictivo, se han analizado los algoritmos más utilizados en industrias en conjunto a diferentes investigaciones. El procedimiento a seguir para crear el modelo conlleva la utilización de algoritmos de preprocesamiento (imputación de valores faltantes, equilibrio de datos y selección de

características, entre otros) así como de algoritmos de predicción como pueda ser KNN o regresión logística.

En referencia al caso de estudio, se ha elegido un banco de datos que trataba sobre la producción de semiconductores, donde los datos provienen de sensores que proporcionan información sobre las piezas producidas. A este *dataset* se han aplicado todos los algoritmos investigados con varias combinaciones posibles.

Con el objetivo de conocer qué modelo aporta el mejor resultado, se dividió el banco de datos en dos: un conjunto de entrenamiento y otro para pruebas. El sistema predictivo se aplicó sobre el conjunto de *test* y se aplicaron varios métodos de evaluación basados en ratios.

En el caso de la creación del modelo predictivo con KNN, los mejores resultados se obtuvieron aplicando una selección de características, lo que no solo corrobora que es importante realizar una cuidadosa selección de atributos, sino que además prueba que el método para recoger estos atributos fue acertado. En el caso de la regresión logística, se obtuvieron peores resultados de predicción además de ser más dispersos y no mostrar un patrón de procedimiento concreto.

Una vez finalizado el trabajo, se proponen varias líneas de trabajo futuras con el fin de optimizarlo, permitir que sea extrapolable y aplicable en una factoría real:

- **Prueba con otros algoritmos:** para comprobar si se obtienen mejores resultados.
- **Implementación en otros lenguajes:** como Java o Python, para dotarle de mayor versatilidad.
- **Extrapolación a otros conjuntos de datos:** para comprobar su eficacia real.
- **Implementación real en una factoría:** para cumplir el objetivo de este estudio.
- **Envío de señales para prevención de errores:** con el propósito de evitar defectos en el producto final y evitar paradas de producción.
- **Estandarización y documentación** del sistema predictivo una vez implantado.

Referencias

1. Vasja Roblek, Maja Mesko and Alojz Krapez. *A Complex View of Industry 4.0*. SAGE Open April-June 2016.
2. Jay Lee , Hung-An Kao and Shanhu Yang. *Service innovation and smart analytics for Industry 4.0 and big data environment*. Center for Intelligent Maintenance Systems (IMS), University of Cincinnati, USA, 2014
3. Jaime G. Carbonell, Ryszard S. Michalsky, Tom M. Mitchell. *Machine Learning: An artificial intelligence approach*. Springer, Verlag Berlin Heidelberg.
4. Brett Lantz. *Machine Learning with R*. Packt Publishing Ltd., 2013
5. Rainer SchmidtEmail, Michael Möhring, Ralf-Christian Härting, Christopher Reichstein, Pascal Neumaier and Philip Jozinovic. *Industry 4.0 - Potentials for Creating Smart Products: Empirical Research Results*. Business Information Systems, 2015.

6. Dhruv Goel: Industry 4.0 and Industrial IoT in Manufacturing: A Sneak Peek, <https://ww.aberdeen.com/opspro-essentials/industry-4-0-industrial-iot-manufacturing-sneak-peek/>
7. Malte Brettel, Niklas Friederichsen, Michael Keller and Marius Rosenberg. *How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective*. World Academy of Science, Engineering and Technology International Journal of Information and Communication Engineering, 2014.
8. Arthur Tang, Charles Owen, Frank Biocca, Weimin Mou. *Comparative effectiveness of augmented reality in object assembly*. SIGCHI Conference on Human Factors in Computing Systems, Florida, USA, 2003.
9. Andrew McAfee and Erik Brynjolfsson. *Big Data: The Management Revolution*. Harvard Business Review, October 2012.
10. Q. Peter He and Jin Wang. *Fault Detection Using the k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes*. IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING, VOL. 20.
11. Peter Schmitt: A Comparison of Six Methods for Missing Data Imputation, <https://www.omicsonline.org/open-access/a-comparison-of-six-methods-for-missing-data-imputation-2155-6180-1000224.php?aid=54590>
12. Michael McCann, Yuhua Li, Liam Maguire and Adrian Johnston. *Causality Challenge: Benchmarking relevant signal components for effective monitoring and process control*. JMLR: Workshop and Conference Proceedings 6, 2008.

Autores



Alejandro Acosta Vellón nacido en Matanzas, Cuba en 1991. Cursó sus estudios pre-universitarios en el Instituto de Ciencias Exactas V.I. Lenin en La Habana. En 2009 trabaja como analista de bases de datos en el Ministerio de las FAR durante su servicio militar. En 2010 inicia sus estudios de Cibernética en la Universidad de la Habana. En 2012 se traslada a la USAL en el Grado en Ingeniería Informática. En 2018 obtiene el título de Máster en Sistemas Inteligentes y actualmente cursa el Doctorado en Informática de la USAL.

André Sales Mendes, nacido en Portugal. Realizó los estudios de bachillerado tecnológico en España. Graduan-do en Ingeniería informática y el Máster de sistemas inteligentes por la Universidad de Salamanca. Actualmente cursa el Doctorado en Informática de la USAL. Es investigador en las ramas de robótica, visión e inteligencia artificial en el grupo de investigación ESAlab de la Universidad de Salamanca.



Gabriel Villarrubia González es Profesor Ayudante Doctor de la Universidad de Salamanca en el Departamento de Informática y Automática. Ingeniero Informático por la Universidad Pontificia de Salamanca (2011), habiendo estudiado el Máster en Sistemas Inteligentes en la Universidad de Salamanca (2012). Además ha realizado el Máster Universitario en Seguridad e Internet (2014) y el Máster en Dirección de Sistemas de Información (2015). A lo largo de su formación ha seguido una línea de investigación bien definida, centrada en la aplicación de los sistemas multi-agente a los entornos de inteligencia ambiental, con especial atención a la definición de arquitecturas inteligentes y a la fusión de información. Ha participado en más de 30 proyectos de investigación, colaborando en la obtención de 15 propiedades intelectuales, habiendo realizado estancias internacionales. En el ámbito científico, es autor de más de 40 publicaciones científicas y codirector de 50 proyectos de fin de grado. Es miembro del comité científico de diferentes congresos, siendo miembro del comité organizador de (ISAMI, PACBB, MIS4TEL, PAAMS, DCAI). A nivel profesional fue

nombrado Microsoft Student Partner en el año 2010. Ha ganado diferentes premios en el desarrollo de aplicativos móviles. Actualmente centra sus esfuerzos en diseñar una arquitectura multi-agente especialmente diseñada para la fusión dinámica de información en entornos inteligentes caracterizados por estar dotados de sensores heterogéneos que capturan información del entorno.

Juan Francisco de Paz Santana, Doctor en Informática y Automática por la Universidad de Salamanca en 2010. Ingeniero técnico en Informática y de Sistemas (2003), Ingeniero Informático (2005) y Diplomado en Estadística (2007) por la Universidad de Salamanca. Actualmente es profesor titular en Departamento de Informática y Automática de la Universidad de Salamanca. Además ha realizado el Máster Universitario en el Desarrollo de Sistemas para el Comercio Electrónico y el Máster Universitario en Animación Digital. Ha obtenido diferentes premios a nivel académico como el Premio de Grado, Premio de doctorado y en otros ámbitos como un premio de SUN Microsystems y otro de la Junta de Castilla y León. En la actualidad trabaja en el campo de los sistemas distribuidos basados en sistemas multiagentes, la Inteligencia Artificial, la bioinformática, la inteligencia ambiental etc. Ha participado en 48 proyectos de investigación a nivel regional, nacional e internacional. Es coautor de más de 180 artículos publicados en diversas revistas de ámbito nacional e internacional, workshops y simposios, 50 de estas publicaciones en revistas con índice de impacto. Ha sido organizador de numerosas conferencias y además ha sido revisor en diferentes conferencias y revistas de ámbito internacional en el área de las Ciencias de la Computación e Inteligencia Artificial. Es miembro del grupo IBASAL y socio de la AEPIA. Evaluador de la ANECA desde 2010 y FONCyT en 2011.



Juan M. Corchado Rodríguez es catedrático de Ciencia de la Computación e Inteligencia Artificial. Ha sido Vicerrector de Investigación y Transferencia de la Universidad de Salamanca y actualmente es Director del Grupo en Biotecnología, Sistemas Inteligentes y Tecnología Educativa (BISITE) de esta misma universidad (<http://bisite.usal.es>). Doctor en Informática por la Universidad de Salamanca, España (1998) y doctor en Inteligencia Artificial por la University of the West of Scotland, Reino Unido (2000). Es Coordinador Académico del Instituto de Arte y Tecnología de la Animación. Anteriormente ha sido Subdirector de la Escuela Superior de Ingeniería Informática en la Universidad de Vigo (España, 1999-00) e investigador en la Universidad de Paisley (1995-98). Colabora, como investigador, con el Laboratorio Oceanográfico de Plymouth (Reino Unido) desde 1993. Ha dirigido más de 90 proyectos

de Inteligencia Artificial con financiación tanto internacional, como nacional o autonómica. Ha dirigido más de 20 tesis doctorales y es co-autor de más de 800 libros, capítulos de libros, artículos en revistas científicas, etc. la mayoría de ellos presentan aspectos tanto prácticos como teóricos relacionados con los Sistemas Híbridos de Inteligencia Artificial, la Biomedicina, la Inteligencia Ambiental, los Sistemas Inalámbricos y la Oceanografía. Es autor de más de una veintena de propiedades intelectuales y patentes.

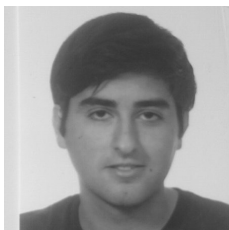
Fabio Andrés Yepes Vélez Nacido en Colombia en 1993. Graduado en Ingeniería Informática y en el Máster en Sistemas Inteligentes por la Universidad de Salamanca al cual accedió por su interés en el campo de la minería de datos y la inteligencia artificial. Actualmente se dedica al desarrollo web Full-Stack aplicando parte de lo aprendido durante el grado y el máster para crear aplicaciones que se adapten a los consumidores tanto funcional como visualmente.



Vivian Félix López Baptista es profesora titular de la Universidad de Salamanca en el área de Ciencias de la Computación e Inteligencia Artificial. Doctorada en Informática por la Universidad de Valladolid en 1996. Miembro del Grupo de Minería de Datos. Ha realizado investigación en diferentes campos como procesamiento del lenguaje natural, redes neuronales y minería de datos. Tiene 80 artículos publicados en revistas de reconocido prestigio, talleres y actas de conferencias, 20 libros y capítulos de libros y 20 informes técnicos, la mayoría de ellos en estos temas. Miembro del comité organizador y científico de varios simposios internacionales. Fue directora del Máster en Sistema Inteligente y del Programa de Doctorado en Informática y Automática de la Universidad de Salamanca desde junio de 2010 hasta octubre de 2012.

María N. Moreno García es Catedrática de Universidad del Departamento de Informática y Automática de la Universidad de Salamanca, directora del grupo de investigación en Minería de Datos y Coordinadora del Programa de Doctorado de Ingeniería Informática. Sus áreas de investigación de interés se centran en el desarrollo y aplicación de algoritmos de minería de datos en diferentes dominios como minería web y de medios sociales o apoyo a las decisiones en medicina.





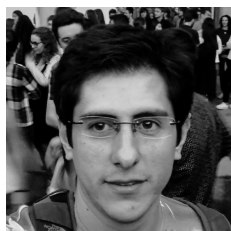
Carlos J. Gomes, graduado en Ing. Informática por la Universidad de Santiago de Compostela. Venezolano de nacimiento, con 8 años viviendo en España. Actualmente trabajo en distintos proyectos informáticos de una Consultora Gallega. He cursado el Master de Sistemas Inteligentes en Salamanca con la intención de ampliar mis conocimientos sobre los distintos enfoques en los que se puede aplicar la informática, con la intención de realizar en un futuro proyectos variados y apasionantes que se alejen de lo convencional.

Ana Belén Gil González es profesora Contratado Doctor en la Universidad de Salamanca en el área de Lenguajes y Sistemas Informáticos. Se Licenció en Ciencias Físicas por la Universidad de Salamanca. Es Doctora Europea en Informática y Automática con Premio Extraordinario de Doctorado por la Universidad de Salamanca. Ha sido investigadora en el LIP6 de la Universidad Pierre Marie Curie (París VI)-CNRS, realizando además estancias en distintos centros de Investigación (ESIEE de París, Universidad do Minho y Universidad de Aveiro). Sus principales áreas de interés en investigación están relacionadas con el desarrollo de marcos tecnológicos para la recuperación y personalización de contenido en diferentes ámbitos y la profundización en la representación y análisis de información, así como su aplicación en sistemas de recomendación. Es coautora de más de 80 publicaciones, incluidos artículos científicos, capítulos de libros y actas de congresos en diferentes áreas de investigación y editoriales. Ha participado en más de 25 proyectos competitivos de investigación de ámbito internacional, nacional y regional. Así mismo, ha participado en más de 10 contratos profesionales (Art. 83 LOU y afines), lo que ha permitido la transferencia conocimiento de la Universidad al tejido empresarial. Fruto de este trabajo se han registrado 10 propiedades intelectuales, 6 de ellas en explotación. Participa como comité científico en más de 20 congresos nacionales e internacionales así como en revistas científicas y en la organización de diferentes congresos.



Christian Bonal Martín Graduado en Ingeniería Informática por la Universidad de Salamanca. Además, ha realizado el Máster Universitario en Sistemas Inteligentes por la misma institución. Actualmente trabaja en el campo de la integración de sistemas informáticos en tecnologías como Microsoft Biztalk o TIBCO, entre otras.

Daniela Elizabeth Sanjinés Flores nació en ciudad de La Paz en Bolivia, tras finalizar el bachillerato con un alto promedio y con Diplomas de honor obtuvo una beca para estudiar el grado en México en la Universidad Autónoma de Chiapas (UNACH), realizó un intercambio por un semestre a la Universidad de Guadalajara, fue titulada con Mención Honorífica como Licenciada en Sistemas Computacionales. Retornó a Bolivia y trabajó en diferentes instituciones como Especialista en Calidad del Software a la par realizó el Diplomado en Gestión de la Seguridad y Auditoría de Sistemas en la Universidad de los Andes y tomó varios cursos relacionados a tecnología informática. Gracias a una beca otorgada por la Universidad de Salamanca (USAL) cursó el Máster en Sistemas Inteligentes en España. Actualmente trabaja como Project Manager en una innovadora empresa de desarrollo de software que aplica inteligencia artificial a sus productos.



Henri Estuardo Casrto León, es un profesional de la Ingeniería Informática que desde temprana edad se inclinó por el desarrollo de software y la tecnología en general. Ecuatoriano de nacimiento, trabajó varios años como desarrollador web y actualmente participa en un proyecto de I+D en el área de Visión Artificial en la Ciudad de Salamanca, donde también realizó la formación de Máster en Sistemas Inteligentes. Sus metas y objetivos son continuar expandiendo su horizonte en los campos de la Inteligencia Artificial para enfocarlos en la industria y los servicios sociales.

Angélica González Arrieta es doctora en Informática por la Universidad de Salamanca. Cuenta con una amplia experiencia investigadora en el campo de sistemas conexionistas. Es Profesora Titular del Departamento de Informática y Automática de dicha Universidad, del área de Ciencia de la Computación e Inteligencia Artificial. Actualmente compatibiliza su labor docente e investigadora con la dirección de diversas actividades formativas sobre recursos tecnológicos, pericia y seguridad informática, colaborando activamente con las Fuerzas y Cuerpos de Seguridad.





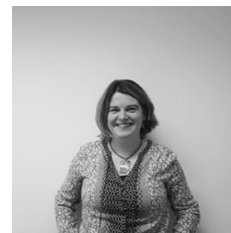
Mehmet Ozturk earned a bachelor's degree in 2014 in Mathematics and Computer Sciences. He has been working as a software developer for five years in various national and international companies. He studied a masters degree in intelligent systems in the University of Salamanca and he has started his PhD at the same university to research on the area of machine learning.

Ricardo Martínez Domínguez Nacido en Valencia en 1995. Graduado en Ingeniería Electrónica por la Universitat Politècnica de València y en el Máster en Sistemas Inteligentes por la Universidad de Salamanca con el objetivo de ampliar conocimientos en el mundo de la industria 4.0. Actualmente desarrollando trabajos referidos a este ámbito aplicando todo lo aprendido en la etapa académica.



Belén Pérez Lancho es Licenciada y Doctora en Ciencias Físicas por la Universidad de Salamanca. Realizó su formación posdoctoral en la Universidad Paris VI (curso 1996-97) y desde 1998 es Profesora Titular en el Área de Ingeniería de Sistemas y Automática de la Universidad de Salamanca. Imparte o ha impartido docencia en las titulaciones de Ingeniería Informática, Física, Estadística y en el Máster en Sistemas Inteligentes. Perteneció al grupo de investigación BISITE y ha participado en más de 20 proyectos de investigación, principalmente en aplicaciones de sistemas de control y sistemas multiagente, colaborando en varias publicaciones científicas, en la dirección de tres tesis y de numerosos proyectos de fin de carrera. Ha ocupado cargos de gestión en la Facultad de Ciencias como Secretaria Académica (años 2000 a 2004) y como Vicedecana (años 2008 a 2016).

Sara Rodríguez González Profesora Contratada Doctor en el Departamento de Informática y Automática de la Universidad de Salamanca. Obtuvo su doctorado en Informática en la misma Universidad en 2010. Recibió el título de Ingeniería Técnica en Informática de Sistemas en 2004 y de Ingeniería en Informática en 2007. Por otra parte, ha llevado a cabo otros estudios de postgrado como el Master en el desarrollo de sistemas de comercio electrónico y el Master en animación de digital. Ha participado como co-autora en artículos publicados en revistas internacionales de reconocido prestigio. También ha



participado como comité de programa en conferencias como PAAMS, IWANN, PACBB, ISAMI, HAIS o DCAI. En la actualidad, es miembro de la Asociación Española para la Inteligencia Artificial (AEPIA), del Instituto de Investigación de Arte y Tecnologías de la Animación de la Universidad de Salamanca y del grupo de investigación BISITE (Bioinformática, Sistemas Inteligentes y Tecnología) en el que sigue su labor investigadora.

