UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA

MODELO PARA EL DESCUBRIMIENTO DE PATRONES EN SERIES TEMPORALES SIMBÓLICAS

TESIS DOCTORAL

MARCO EDUARDO MOLINA BUSTAMANTE

INGENIERO ELÉCTRICO

MASTER EN CIENCIA DE LA COMPUTACIÓN

MADRID 2017

DEPARTAMENTO DE LENGUAJES, SISTEMAS INFORMÁTICOS E INGENIERÍA DE SOFTWARE

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INFORMÁTICOS, UNIVERSIDAD POLITÉCNICA DE MADRID

MODELO DE DESCUBRIMIENTO DE PATRONES EN SERIES TEMPORALES SIMBÓLICAS

Autor

Marco Eduardo Molina Bustamante

Ingeniero Eléctrico

Máster en Ciencia de la Computación

Directores

Aurora Pérez Pérez

Doctora en Informática

Juan Pedro Caraça-Valente Hernández

Doctor en Informática

Mayo de 2017

UNIVERSIDAD POLITECNICA DE MADRID

Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día.....de.....de 2017

Presidente: _____

Vocal: _____

Vocal: _____

Vocal: _____

Secretario: _____

Suplente: _____

Suplente: _____

Realizado el acto de defensa y lectura de la Tesis el día.....de.....de 2017 en la E.T.S.I. Informáticos.....

EL PRESIDENTE LOS VOCALES

EL SECRETARIO

A la razón de mi vida…

mi familia.

AGRADECIMIENTOS

Eterna gratitud a los directores del presente trabajo, Aurora y Juan Pedro, quienes han dedicado mucho de su valioso tiempo a revisar, sugerir y plantear nuevos retos durante todos estos años, constituyéndose por tanto, en actores fundamentales para el logro de los objetivos propuestos.

A Julia Beatriz Terán, le debo mi agradecimiento por su invalorable ayuda para la recogida de datos, por su tiempo dedicado a mostrarme los detalles del dominio, por el acceso a los equipos especializados para diagnóstico y por facilitar el contacto con los especialistas en el área de los Potenciales Evocados Auditivos de Tronco Cerebral.

A los empleados y funcionarios del Hospital Homero Castañer Crespo que colaboraron con el relevamiento de información y me prestaron la ayuda logística necesaria para realizar ese trabajo.

Agradezco a mis profesores, quienes me ilustraron en algunos temas del área de la informática, aún desconocidos para mí, que me fueron de mucha utilidad durante el desarrollo de la presente tesis.

Cómo no hacer llegar mi reconocimiento a mis compañeros; ellos me ayudaron a salir adelante en momentos en que la vida se volvía difícil lejos de mi familia. Me gustaría mencionar de manera especial a Fernando, María, Detzabeth, Danilo, Anaolena y Rolando.

Finalmente, no solo debo agradecer sino devolverles la el tiempo no vivido junto a ellos, a los miembros de mi familia: Anita Lucía, Anita Belén, Rafaela y Martín, quienes también hicieron grandes sacrificios para que yo pueda realizar mis estudios durante más de cuatro años.

RESUMEN

La clasificación de series temporales numéricas es una tarea de *data mining* indispensable en casi todos los dominios, incluyendo algunas ramas de la medicina. Los métodos de clasificación conocidos se ejecutan usando procedimientos que toman en cuenta los valores numéricos de las series sin prestar atención al contexto, la forma y el significado que esos valores pudieran tener dentro de la serie. Pocos estudios enfocan la abstracción del contenido de la serie para descubrir conocimiento compatible con la problemática propia del dominio y en términos inteligibles para los expertos del dominio.

El propósito de la presente tesis es obtener un método para clasificar series temporales, sobre la base del descubrimiento de patrones frecuentes encontrados en conjuntos de secuencias simbólicas. Las secuencias simbólicas, por su parte, serán generadas a partir de series temporales numéricas por medio de un proceso de abstracción temporal que tome en cuenta el conocimiento experto del dominio.

Para lograr el propósito, se ejecutan tres pasos que consisten en:

- En primer lugar, se transforman las series temporales numéricas en secuencias temporales simbólicas, en las que los símbolos tienen como objetivo representar los conceptos relevantes del dominio. Esos símbolos pueden ser definidos usando conocimiento, tanto experto como público, sobre el dominio;
- A continuación se aplica una técnica de descubrimiento de patrones simbólicos sobre las secuencias simbólicas obtenidas. Esta técnica identifica las subsecuencias encontradas frecuentemente en el grupo de población y se denominan patrones frecuentes que son representativos de los grupos de población;
- Finalmente, se emplea una técnica de clasificación basada en los patrones identificados, a fin de clasificar nuevos individuos. Gracias a la inclusión de conocimiento del dominio, los resultados de la clasificación pueden ser explicados usando la terminología del dominio. Esto hace que los resultados sean más fáciles de interpretar para los especialistas del dominio.

Este método ha sido aplicado a las series temporales generadas por las pruebas médicas de Potenciales Evocados Auditivos de Tronco Cerebral (PEATCs). Preliminarmente, se realizaron experimentos para analizar varios aspectos del método, incluyendo la mejor configuración de los parámetros de la técnica para el descubrimiento de patrones. Luego se aplicó el método a las respuestas auditivas del tronco cerebral (ABRs, siglas de la expresión en inglés *Auditory Brainstem Responses*) de 83 individuos pertenecientes a cuatro clases (sanos, con pérdida

conductiva de audición, con schwannoma vestibular – implicación del tronco cerebral y con schwannoma vestibular - implicación del 8º-nervio). De acuerdo con los resultados de la validación cruzada, la exactitud global del clasificador fue del 99.4%, la sensibilidad fue de 97.6% y la especificidad fue del 100% (sin falsos positivos).

El método propuesto reduce la dimensionalidad del problema de forma efectiva. Adicionalmente, si la transformación simbólica incluye el conocimiento correcto del dominio, podría decirse que el método produce una representación de datos que denota los conceptos relevantes del dominio con mayor claridad. Durante la experimentación aplicando el método, se encontraron patrones en series temporales de ABRs y se alcanzó un alto grado de precisión al predecir correctamente cuándo un paciente tiene un desorden auditivo o no.

ABSTRACT

Numeric time series classification is an indispensable *data mining* task for almost all domains, including many branches of medicine. The known classification methods run using procedures that take into account only the numeric values of data without paying attention to the context, the form and the meaning that these values could have within the series. Few studies focus on the abstraction of the content of the series to discover knowledge compatible with the problems of the domain and in terms intelligible to domain experts.

The purpose of the present thesis is to propose and proof a method to classify temporal series, based on the discovery of frequent patterns that will be found in sets of symbolic sequences. Those sequences will be obtained from numerical time series, through a process of temporal abstraction that takes into account the domain's expert knowledge.

To achieve the purpose, three steps are performed:

- First, numeric time series are transformed into symbolic temporal sequences where the symbols aim to represent the relevant concepts of the domain, these symbols can be defined using domain knowledge, both expert and public;
- Then a symbolic patterns discovery technique is applied to the obtained symbolic sequences. This technique identifies the subsequences frequently found in the population group and they are called frequent patterns that are representative of the population groups;
- Finally, a classification technique based on the identified patterns is used, in order to classify new individuals. Thanks to the inclusion of domain knowledge, classification results can be explained using domain terminology. This makes the results easier to interpret for domain specialists.

This method has been applied to time series generated by medical exams with brainstem auditory evoked potentials (BAEPs). Preliminary experiments were carried out to analyse several aspects of the method including the best configuration of the pattern discovery technique parameters. We then applied the method to the BAEPs of 83 individuals belonging to four classes (healthy, conductive hearing loss, vestibular schwannoma - brainstem involvement and vestibular schwannoma - 8th-nerve involvement). According to the results of the cross-validation, the classifier overall accuracy was 99.4%, sensitivity (recall) was 97.6% and specificity was 100% (no false positives).

The proposed method effectively reduces the problem's dimensionality. Additionally, if the symbolic transformation includes the right domain knowledge, the method arguably outputs a data representation that denotes the relevant domain concepts more clearly. The method is capable of finding patterns in BAEPs time series and is very accurate at correctly predicting whether or not new patients have an auditory-related disorder.

ÍNDICE GENERAL

CAPÍTULO 1. INTRODUCCIÓN

Las series temporales constituyen un tipo de datos complejo que consiste en una secuencia (potencialmente grande) de valores ordenados en el tiempo, que representan las magnitudes de determinados eventos. En los últimos años, la investigación sobre series temporales ha atraído mucho el interés de la comunidad científica de *data mining*, cuyas técnicas son usadas sobre tales estructuras para encontrar correlaciones, patrones o previsiones, que puedan constituir conocimiento relevante en el dominio de aplicación. Sobre esta base, numerosas han sido las aplicaciones que se han encontrado en los diferentes dominios. Algunos ejemplos son:

Industria

- Detección de anomalías [Shin et al. 2005]
- Análisis del habla [Timofte 2007], [Agbinya 1996]
- Clasificación de imágenes [Barat et al. 2010]
- Medidas de combustión en máquinas [Finney et al. 1998]
- Predicción de eventos [Molina et al. 2009]

Medicina

- Expresión génica [Das et al. 2007]
- Diagnóstico médico [Ordóñez, Jardins 2011]
- Vigilancia médica [Burkom et al. 2006]

Astronomía

- Análisis de datos astronómicos [Pelt 2003]
- Análisis de datos espaciales [Eckley 2001]

Finanzas

- Análisis económico [Brida 2000]
- Predicción de Bolsa [Alvo et al. 2011]
- Predicción de stock [Azevedo et al. 2012]
- Predicción de series temporales financieras [Diggs, Povinelli 2003]
- Estrategias de inversión [Canelas et al. 2012]

Educación

- Problemas de aprendizaje [David, Balakrishnan 2010]
- Análisis de asistencia a clases [Koopmans 2012]
- Previsión de inscripción en el sistema escolar [Lavilles, Arilla 2012]

Otras áreas

- Análisis y control de tráfico [Silvasan et al. 2014]
- Comparación de huellas [Parthasaradhi et al. 2005]
- Análisis de datos sísmicos [Bertens, Siebes 2014]
- Agricultura [Romani 2010]

Predicción de carga eléctrica [Alfares, Nazeeruddin 2002]

Ecología [Cazelles et al. 2008]

Estimación de tasa de desempleo [Brakel, Krieg 2008]

Patrones musicales [Lartillot, Ayari 2006]

Redes de comunicaciones [Homayounfard, Kennedy 2009], [Homayounfard 2013]

Las aplicaciones mencionadas, se basan en sofisticadas técnicas que implementan tareas de *data mining* sobre series temporales, que permiten descubrir conocimiento oculto en la enorme cantidad de datos existentes. Entre las tareas de data mining más usadas se incluyen: descubrimiento de reglas de asociación, determinación de *clusters*, clasificación, detección de atípicos, detección de anomalías y descubrimiento de patrones. Son dos los principales problemas que hay que enfrentar para realizar estas tareas, el primero es la gran dimensionalidad de las series temporales, lo cual se subsana con métodos de reducción de dimensionalidad como la segmentación o la simbolización; otro problema es el establecimiento de un criterio de similitud para lo cual se utiliza un concepto de distancia. La solución que se escoja en cada caso dependerá del dominio.

1.1. Propósito y objetivos de la tesis

El presente trabajo tiene como propósito la creación de un modelo para la clasificación de series temporales numéricas, empleando para ello conocimiento experto y/o conocimiento público. Tanto el conocimiento experto como la información existente sobre los conjuntos de series son de suma importancia en el ámbito del *data mining* en series temporales, especialmente para aquellos dominios en los cuales se parte de un punto en el cual ya existe un camino andado en el diagnóstico de situaciones anómalas.

Para lograr tal propósito, han de cumplirse los objetivos que se mencionan a continuación:

1. Convertir una serie temporal numérica en una secuencia simbólica que refleje las características específicas del dominio al mismo tiempo que conserve toda la información relevante.

2. Encontrar patrones en conjuntos de secuencias temporales simbólicas pertenecientes a una misma clase, utilizando técnicas de *data mining*.

3. Clasificar secuencias temporales simbólicas nuevas, sobre la base de los patrones encontrados.

Los objetivos enumerados se cumplirán mediante la realización de modificaciones a técnicas existentes, para el cálculo de las medidas de similitud y para la obtención de patrones en

secuencias simbólicas y, adicionalmente, se aplicará un método nuevo para la transformación de las series temporales numéricas en secuencias simbólicas. Dicho método se basa en una definición de símbolos, realizada por el experto en el dominio, lo cual es algo novedoso en el ámbito del *data mining* en series temporales.

Durante el desarrollo de esta tesis se pretende crear un modelo genérico, en cuanto al dominio de aplicación, esto es, el modelo deberá ser aplicable en cualquier campo, siempre y cuando se cumplan las restricciones a las que se hace referencia en el epígrafe 3.3 de la presente tesis.

Para probar el modelo que se propone en el capítulo 4, se decidió trabajar en colaboración con el Hospital Homero Castanier Crespo en Ecuador en un dominio médico dentro del área de la audiología, concretamente en exámenes de Potenciales Evocados Auditivos de Tronco Cerebral PEATC aplicados al diagnóstico de problemas de oído interno y de la porción del tallo cerebral asociada a la función auditiva. Las pruebas experimentales fueron aplicadas a un grupo de personas adolescentes con alto riesgo de adquirir ese tipo de patologías, como son la pérdida auditiva conductiva y los schwannomas vestibulares.

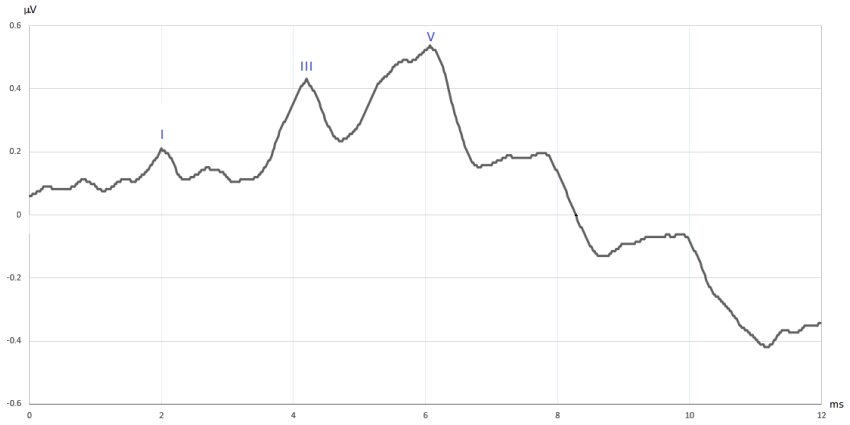

Figura 1.1. Serie temporal que representa una ABR.

Los PEATC son potenciales electrofisiológicos generados como respuesta a estímulos acústicos. Las señales medidas son el resultado de varias corrientes iónicas asociadas con procesos de transducción que tienen lugar en las vellosidades de la cóclea y con la generación de potenciales de acción en las fibras del nervio auditivo. Un uso común de los PEATC en aplicaciones clínicas es el diagnóstico de problemas auditivos y de tumoraciones a nivel vestibular. Los datos resultantes, que se denominan ABR (Auditory Brainstem Response), son series temporales con una duración de hasta 15ms, que reflejan la evolución en el tiempo de la respuesta del tronco cerebral a un estímulo acústico. La figura 1,1 muestra un ABR con las ondas más comúnmente usadas en la práctica (los picos relevantes I, III, y V). Los PEATC han sido

usados extensamente como una herramienta electrofisiológica no invasiva, para el estudio del sistema auditivo humano [Burkard el al. 2007a], [Burkard el al. 2007b].

1.2. Síntesis del contenido

La presente tesis ha sido organizada en nueve capítulos, siendo el primero la presente introducción y los ocho restantes se resumen a continuación:

- **Capítulo 2.** Conceptos y trabajos relacionados. En este capítulo se pretende dar una visión general del estado del arte, tanto en lo concerniente al *data mining* como en lo que tiene que ver con las series temporales. Como es de suponer, en este compendio se hace énfasis en la presentación de aquellos aspectos que más se ajustan a los objetivos de esta tesis, como son el cálculo de las distancias de edición y la búsqueda de patrones en secuencias de símbolos.
- **Capítulo 3.** Descripción del problema. Aquí se describen los aspectos que motivaron la realización de la presente tesis y se describe detalladamente el problema que se pretende resolver. Se mencionan los pros y los contras de los modelos existentes, específicamente en lo concerniente a la transformación de series temporales numéricas en secuencias simbólicas.

- **Capítulo 4.** Modelo propuesto. En este capítulo se presenta el modelo que se propone para el descubrimiento de patrones en series temporales simbólicas, mediante la explicación detallada de todos los pasos del proceso. Para asegurar una mejor comprensión de las descripciones que aquí se realizan, se han incluido ilustraciones consistentes en abstracciones esquemáticas de los elementos del proceso.

- **Capítulo 5.** Experimentación. El modelo propuesto en el capítulo 4 se prueba a través de una rigurosa experimentación, que se documenta en este capítulo. Se presentan los conceptos más relevantes del dominio de prueba; el proceso de adquisición de los datos usados para el experimento y la prueba del modelo propiamente dicha.

- **Capítulo 6.** Resultados. Aquí se presentan los resultados obtenidos en la fase de experimentación

- **Capítulo 7.** Conclusiones. Este capítulo presenta las conclusiones que han podido ser extraídas de la experimentación y sus resultados, como una forma de contrastar lo que fue propuesto con lo que finalmente se obtuvo.

- **Capítulo 8.**Líneas futuras de investigación. Mucho camino queda por andar a partir de la presente propuesta; en este capítulo se mencionan las posibles líneas de investigación que pudieran ser abordadas a futuro a fin de avanzar con el desarrollo de las ideas expuestas en la presente tesis.

- **Capítulo 9.** Referencias. Todas las referencias bibliográficas, que fueron consultadas para la realización de este trabajo y que de alguna manera sustentan el mismo, son enumeradas

CAPÍTULO 2. CONCEPTOS Y TRABAJOS RELACIONADOS

El volumen de datos que se recogen día a día en las áreas de la ciencia, la tecnología, la ingeniería, los negocios y la medicina, entre otras, crece a un ritmo dramático y ha convertido en una tarea casi imposible la obtención de información estratégica de calidad mediante la utilización de los métodos tradicionales del campo de las bases de datos [Goebel, Gruenwal 1999]. Y este ritmo no ha hecho más que aumentar en los últimos años. Esto ha llevado a los investigadores a proponer nuevas formas de almacenamiento de datos y nuevos procesos de recuperación de información que se engloban en el concepto de Descubrimiento de Conocimiento en Bases de Datos o KDD - por sus siglas en inglés, *Knowledge Discovery in Databases* - que ha sido definido en la literatura como "el proceso no trivial de identificación de patrones en los datos, que sean válidos, novedosos, potencialmente útiles y, por último, inteligibles" [Fayyad et al. 1996] y que será expuesto en el presente capítulo a través de los aspectos subyacentes a esta tesis y que son el *data mining* y el análisis de series temporales.

Generalmente se tiende a intercambiar los términos KDD y *data mining*, como si se tratase de dos sinónimos [Goebel, Gruenwal 1999] aunque, siendo estrictos, el *data mining* es un paso dentro del proceso de KDD como se puede apreciar en la figura 2.1. Es necesario tomar en cuenta que para que sea posible la realización del *data mining* sobre un conjunto de datos específico, se debe cumplir con las tres primeras fases del proceso de KDD, o sea las de selección, pre-procesamiento y transformación de los datos.

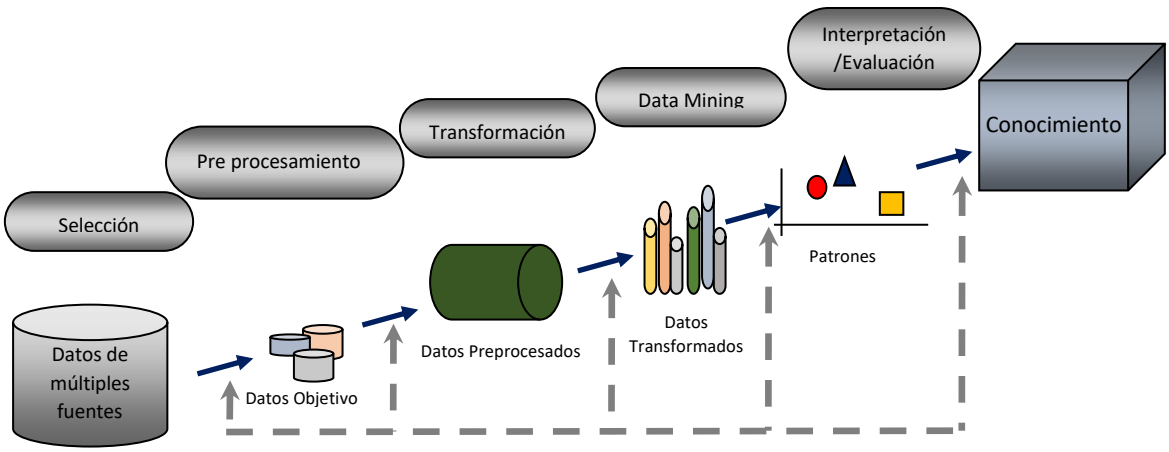

Figura 2.1. Proceso de Descubrimiento del Conocimiento

El *data mining* como núcleo del proceso de KDD es definido en la literatura como "El proceso de exploración y análisis, por medios automáticos o semiautomáticos, de grandes cantidades de datos a fin de descubrir patrones y reglas significativos" [Jain, Dubes 2013] y se relaciona estrechamente con otras áreas del quehacer científico como la estadística, *machine learning* y bases de datos. De esas áreas se han derivado muchas de las técnicas que hoy en día se conocen en el ámbito del *data mining* [Fayyad et al. 1996].

En la siguiente sección se presentan los conceptos y las técnicas más relevantes del *data mining*, haciendo énfasis en aquellos aspectos que se corresponden con el modelo que se propondrá en el capítulo 4 de la presente tesis.

2.1. *Data mining*

Las etapas previas a la realización del *data mining*, sobre un conjunto de datos son muy importantes, pues los datos tienen que llegar limpios - sin ruido - , depurados y con un nivel de abstracción tal que permita conseguir niveles razonables de eficiencia y rendimiento. Cada una de las etapas del proceso de KDD es iterativa, lo que significa que, en caso de que no se llegue a obtener un resultado satisfactorio, se puede volver a uno de los pasos anteriores, para refinarlo y realizar los cambios convenientes y así mejorar los resultados del proceso.

En el siguiente epígrafe se describen brevemente las etapas del proceso de KDD, previas al *data mining*, mientras que en el resto del capítulo se realiza un recorrido por los diferentes métodos de *data mining*.

2.1.1. Etapas previas al *data mining*

2.1.1.1. Selección de datos

Los datos pueden provenir de la combinación de varias fuentes. En esta fase se realiza una selección observando posibles problemas de compatibilidad entre las fuentes. En esta fase es necesario tener un gran entendimiento acerca del dominio y, por consiguiente, la selección de los datos se orientará a obtener la muestra adecuada de registros en los que estén involucradas la variable objetivo y las variables independientes. La variable objetivo es aquella sobre la cual se desea obtener las predicciones, cálculos o inferencias y las independientes son aquellas que servirán como soporte para realizar los cálculos o los procesos pertinentes.

Una buena selección de los datos, esto es, si los datos seleccionados son realmente los adecuados para lograr los objetivos previstos, se constituye en un gran aporte para el proceso de descubrimiento del conocimiento.

La figura 2.1 ilustra este paso del proceso de KDD. Podemos observar que la entrada del proceso es un conjunto de datos provenientes de varias fuentes y que, tras la selección, se tiene el conjunto de datos objetivo. Además, si el conjunto de datos objetivo obtenido no se corresponde con el requerido por el proceso, es posible una nueva selección, en la que se corrijan las falencias de la anterior.

2.1.1.2. Pre-procesamiento de datos

El *data mining* debe ser realizado, idealmente, sobre datos limpios y debidamente resumidos, lo cual implica la realización de una serie de actividades tendientes a depurar los datos objetivo y obtener así un conjunto de datos confiables sobre el cual se podrá continuar con el proceso. Las actividades a las que se hace referencia son:

- Eliminación de ruido,
- Eliminación de datos erróneos,
- Tratamiento de valores que faltan o valores nulos,
- Tratamiento de valores inconsistentes y/o atípicos,
- Creación de nuevos atributos y/o eliminación de otros,
- Integración de las fuentes de datos.

Dos son los tipos de contaminación que generalmente presentan los datos objetivo: en primer lugar está el ruido, que suele contaminar los datos de principio a fin; y, en segundo lugar están los datos erróneos, valores inconsistentes y valores que faltan, que suelen estar dispersos a lo largo de la muestra, pero requieren un riguroso tratamiento porque su presencia muy posiblemente introducirá distorsiones en el resultado final del proceso.

A fin de compatibilizar las múltiples fuentes de datos, una actividad adicional es necesaria; consiste en crear nuevos atributos y/o eliminar algunos ya existentes, de tal manera que las mismas estructuras de datos sean compartidas por todas las fuentes.

Finalmente, se requiere fusionar las varias fuentes de datos en una sola estructura [Zaiane 1999], lo cual es posible gracias a que las fuentes fueron debidamente compatibilizadas.

2.1.1.3. Transformación de datos

Los datos pre-procesados, resultantes de la etapa anterior, pueden requerir un nuevo tratamiento, antes de la realización del *data mining*, pues es necesario que su formato se ajuste al método de *data mining* que se pretenda aplicar. Habitualmente, se suelen cambiar atributos de un tipo de dato a otro, para ajustar de mejor manera la estructura del conjunto de datos resultante al método de *data mining* que se va a aplicar. Esta fase tiene que ver también con la reducción, tanto de la cantidad de dimensiones (variables), como de la numerosidad de los datos (número de registros), siendo necesaria la eliminación de los atributos irrelevantes y, en ocasiones, la agregación de alguno nuevo, que equivalga a la combinación de dos o más atributos. Por ejemplo, si en un conjunto de datos están los atributos longitud y ancho, mientras por otro lado está el

atributo costo por metro cuadrado; lo más conveniente podría ser combinar longitud y ancho en uno solo llamado superficie.

Como ya se mencionó anteriormente, el KDD es un proceso iterativo [Zaiane 1999], a lo cual hay que añadir que también es interactivo, razón por la cual el ingeniero de *data mining* es capaz de interactuar con el proceso en todas sus etapas.

2.1.2. Métodos de *data mining*

El fin último del *data mining* es obtener patrones de interés a partir de los datos. Estos patrones pueden ser representados de varias maneras, dependiendo de los métodos que pudieran haber sido aplicados sobre ellos. Entre las formas más comunes de representación de patrones están las reglas de asociación, reglas de clasificación, grupos o *clusters*, patrones secuenciales, resúmenes obtenidos usando alguna estructura jerárquica o taxonomía [Hilderman, Hamilton 1999], etc. La importancia que un determinado patrón pudiera tener dentro del proceso de descubrimiento del conocimiento está dada por el grado de interés que, quienes toman decisiones en el dominio, tengan en éste [Padmanabhan, Tuzhilin 1998].

Los métodos de *data mining* pueden ser clasificados, de acuerdo con la literatura, de varias maneras. En el presente repaso, los métodos se clasificarán de acuerdo con los tipos de conocimiento a ser buscados, pues de esta manera se obtiene una visión clara de los diferentes requerimientos y técnicas de *data mining* [Chen et al. 1996], según ello, las técnicas se dividen en reglas de asociación, clasificación y agrupamiento, que serán descritos con cierto nivel de detalle a continuación.

2.1.2.1. Reglas de asociación

Una aplicación muy conocida de *data mining* ha sido la determinación de asociaciones entre los productos que un cliente coloca en la cesta de la compra; el problema se puede enunciar como "Encontrar los conjuntos de productos que se compran juntos con una probabilidad mínima s y una fiabilidad mínima d". El resultado de la búsqueda de tales conjuntos servirá para la toma de importantes decisiones en el negocio y el consecuente mejoramiento en la calidad de atención a los clientes y optimización de costos y ganancias.

La definición formal para la técnica de minería de reglas de asociación según [Agrawal et al. 1993a] es la siguiente:

"Encontrar patrones frecuentes, asociaciones, correlaciones u otras estructuras entre conjuntos de ítems u objetos en bases de datos transaccionales, bases de datos relacionales u otros repositorios de información".

El formato de una regla de asociación es: Cuerpo → Cabeza[soporte, confianza]

Ejemplo: Compra(X,"cerveza") → compra(X,"jamón")[1.5%,62%].

La regla del ejemplo afirma que si el cliente X compra cerveza, entonces comprará también jamón, con un soporte del 1.5% y una confianza del 62%. El soporte, en este caso, significa que la probabilidad de que una transacción contenga tanto cerveza como jamón es del 1.5%, mientras que la confianza indica que la probabilidad de que una transacción, de la cual se conoce que contiene cerveza, contenga también jamón es del 62%; a este tipo de probabilidad se la denomina probabilidad condicional.

Formalmente:

Sea *I* un conjunto de ítems con $|I|$ = n, $I = \{i_1,i_2,...,i_n\}$
Sea *D* un conjunto de transacciones con $|D|$ = m, $D = \{d_1,d_2,...,d_m\}$
Una regla de asociación, $A => B$, donde $A \subset I, B \subset I, A \cap B = \emptyset$
Probabilidad del conjunto A, $P(A) = \frac{\sum_i C(A,d_i)}{|D|}$, donde $C(X,Y) = \begin{cases} 1 \; si \; X \subseteq Y \\ 0 \; de \; otro \; modo \end{cases}$
El *soporte* de la regla $A => B$ se define como:

$$soporte(A => B) = P(A,B)$$

Expresión 2.1. Soporte de la regla $A => B$

La *confianza* de la regla $A => B$ se define como:

$$confianza(A => B) = P(B|A) = \frac{P(A,B)}{P(A)} = \frac{soporte(A,B)}{soporte(A)}$$

Expresión 2.2. Confianza de la regla $A => B$

Varios algoritmos para búsqueda de reglas de asociación han sido desarrollados, entre los cuales se destaca el algoritmo Apriori, que ha sido difundido extensamente a través de la literatura científica y académica. A continuación se relacionan once algoritmos de reglas de asociación, de los cuales se mencionarán algunas de sus características; finalmente se detallará el funcionamiento del algoritmo Apriori.

AIS [Agrawal et al. 1993a]

- Agrawal, Imielinski y Swami son los apellidos de los autores del artículo en el que, por primera vez, se publicó un algoritmo para encontrar reglas de asociación en conjuntos de datos, las siglas de AIS se corresponden con las iniciales de los tres apellidos ya mencionados.

- Este algoritmo es considerado en la literatura como el primer algoritmo diseñado para descubrir reglas de asociación.

- Sea I un conjunto de ítems con $|I| = n$, $I = \{i_1, i_2, ..., i_n\}$

- El consecuente de las reglas de asociación solamente puede contener un ítem. La definición formal de una regla es: sea $X => I_j$ [c] donde X es un conjunto de ítems e $I_j \in I$ y c es la confianza de la regla.

- Un problema grave que tiene este algoritmo es que genera una gran cantidad de ítems candidatos irrelevantes, lo cual compromete el rendimiento y el uso de memoria.

SETM [Houtsma 1995]

- El nombre SETM del algoritmo se debe a la naturaleza del mismo, SET-oriented Mining, cuyo objetivo es obtener reglas de asociación mediante operaciones de conjuntos.

- Usa el SQL como lenguaje en un entorno de bases de datos relacionales, con lo cual, según sus creadores, aprovechan el optimizador de consultas para encontrar la mejor manera de realizar las operaciones requeridas.

- Parte del supuesto de que en la base de datos existe una tabla con el esquema *SALES(transact-id, item)*, en la que se almacenan todos los ítems conjuntamente con la clave primaria de las transacciones.

- Se realizan varios barridos a la base de datos, lo cual genera un detrimento en el desempeño del algoritmo.

- Los conjuntos de ítems candidatos se guardan en la base de datos, que demanda un excesivo coste de almacenamiento.

Apriori [Agrawal, Srikant 1994]

- Considerado por [Cheung et al. 1996] como un gran logro en la historia de la minería de reglas de asociación.

- La principal diferencia entre el algoritmo Apriori y el algoritmo AIS radica en la manera en que se eligen a los conjuntos de ítems candidatos.
- La *propiedad apriori* es la que hace de este algoritmo uno de los más importantes para la obtención de reglas de asociación: "Todos los subconjuntos de ítems no vacíos de un conjunto frecuente de ítems, deben también ser frecuentes". Esta propiedad permite que los superconjuntos de conjuntos no frecuentes de ítems sean descartados, de forma definitiva, del espacio de búsqueda.
- Mejora considerablemente el uso de la memoria.
- Mejora substancialmente el desempeño del algoritmo con respecto a sus antecesores.

Apriori TID [Agrawal, Srikant 1994]

- Este algoritmo usa la misma función que el algoritmo Apriori para determinar los conjuntos candidatos de ítems.
- La gran diferencia con el algoritmo Apriori radica en el barrido de la base de datos pues, mientras Apriori realiza un barrido completo, el algoritmo Apriori TID realiza barridos cada vez menores, conforme va aumentando el número de ítems *k* en los conjuntos candidatos.
- A partir del segundo nivel, el conteo para el cálculo del soporte lo realiza en la memoria principal, siendo ésta la característica más influyente en este algoritmo.
- El uso de memoria se ve afectado por la característica mencionada arriba.
- El desempeño de este algoritmo es superior al del algoritmo Apriori cuando los tamaños de conjuntos candidatos son pequeños y caben en la memoria principal; sin embargo, es superado cuando esos conjuntos son grandes.

Apriori Hybrid [Agrawal et al. 1996]

- Combina las ventajas de Apriori y Apriori TID, para lograr un mejor rendimiento y una optimización del uso de la memoria.
- Inicia el proceso con el algoritmo Apriori y, cuando los conjuntos de candidatos se hacen pequeños, de tal manera que quepan en la memoria principal, cambia al algoritmo Apriori TID.

OCD [Mannila et al. 1994]

- OCD son las siglas de *Offline Candidate Determination*. El principal objetivo de este algoritmo es disminuir drásticamente el número de conjuntos candidatos, al pasar de un nivel a otro.

- Para lograr lo mencionado en el párrafo anterior, se usa una técnica de análisis combinatorio de la información obtenida de barridos previos, para eliminar conjuntos candidatos innecesarios.

- Según sus autores, este algoritmo supera en desempeño al algoritmo AIS con un factor de cinco.

Partition [Savasere et al. 1995]

- Según sus autores, este algoritmo fue diseñado para búsqueda de reglas de asociación entre ítems en bases de datos de gran tamaño, y han logrado una mejora del rendimiento del CPU y de las operaciones de entrada y salida por un factor de cuatro, con respecto a uno de los mejores algoritmos previamente existentes.

- Realiza como máximo dos barridos a la base de datos para obtener todas las reglas.

- En el primer barrido, divide el conjunto de transacciones en particiones y encuentra los candidatos a conjuntos grandes de ítems y en el segundo barrido determina los conjuntos grandes de ítems con sus soportes.

- Las reglas son determinadas en la memoria principal, sin acceder a la base de datos.

Sampling [Toivonen 1996]

- Utiliza técnicas estadísticas para determinar las reglas de asociación en grandes bases de datos.

- El nombre del algoritmo se debe a que toma una muestra - *sample* en inglés - aleatoria de la base de datos sobre la cual realiza la búsqueda de reglas de asociación que, probablemente, sirven para toda la base de datos. Para lograrlo, utiliza uno de los algoritmos conocidos, por ejemplo Apriori.

- Una vez encontradas las reglas sobre la muestra, las verifica sobre el resto de la base de datos.

- Las reglas que no han sido encontradas sobre la muestra, pueden ser determinadas realizando un segundo barrido a la base de datos.

- El autor afirma que el algoritmo encuentra las reglas de asociación muy eficientemente, con un solo barrido de la base de datos.

CARMA [Hidber 1998]

- Su nombre, según sus siglas en inglés, significa **C**ontinuous **A**ssociation **R**ule **M**ining **A**lgorithm.
- Este algoritmo genera un superconjunto de todos los conjuntos grandes encontrados durante un primer barrido a la base de datos, junto con sus soportes, que van encogiendo hasta terminar este barrido.
- Como máximo, en dos barridos a la base de datos el algoritmo obtiene los conjuntos grandes de ítems y sus soportes definitivos.
- Para todos los conjuntos grandes encuentra las reglas de asociación cuya confianza es mayor que la confianza mínima.
- En cuanto al uso de memoria, este algoritmo se muestra más eficiente que Apriori.

Los algoritmos mencionados, son apenas algunos de los que existen en la literatura sobre el tema. Puede resultar interesante leer los estudios realizados por [Hipp et al. 2000], [Zhao, Bhowmick 2003], [Suriya et al. 2012], [Dixit 2014] y [Saxena, Gadhiya 2014], y analizar la propuesta realizada en [Ayad et al. 2001]. Para concluir con esta subsección, se presenta la explicación detallada del funcionamiento del algoritmo Apriori.

Un ejemplo de cómo se desarrolla el proceso de reconocimiento de reglas de asociación siguiendo los pasos del algoritmo Apriori, se observa en la figura 2.2. En primer lugar se tiene una base de datos transaccional que consiste en una sola tabla cuya estructura es el identificador de la transacción (Tid) y los ítems (Items) asociados a ellas.

El algoritmo Apriori es iterativo: en cada ciclo el objetivo es encontrar los conjuntos candidatos para formar las reglas de asociación que se buscan. En el k-ésimo ciclo (k es un número natural) se dice que se ha llegado al nivel k y se buscan los conjuntos candidatos de orden k, conocidos en la literatura como los k-itemsets. Cada k-itemset tiene k ítems y su frecuencia en la base de datos es el soporte. Volviendo a la figura 2.2, en el nivel 1 se extraen los 1-itemsets que son todos los conjuntos con un ítem, cuyo soporte sea superior al soporte mínimo, que es un parámetro del algoritmo. En el ejemplo de la figura 2.2., el conjunto que tiene soporte inferior al soporte mínimo es {D}. Por lo tanto los conjuntos superiores ya no contendrán el ítem D, de acuerdo con la propiedad apriori. En el siguiente ciclo, se combinan los 1-itemsets con ellos mismos para formar

conjuntos de dos ítems y, basado en el soporte, encontrar los 2-itemsets. En este nivel se puede observar que los conjuntos {A,B} y {A,E}, marcados con amarillo, son descartados, para a partir de los cuatro 2-itemsets restantes, formar el único 3-itemset posible. El algoritmo termina aquí su etapa de búsqueda de los k-itemsets, pues ya no es posible formar conjuntos de nivel más alto. Finalmente, se determinan las reglas de asociación a partir de los k-itemsets encontrados, se cuentan y se determinan sus factores de confianza.

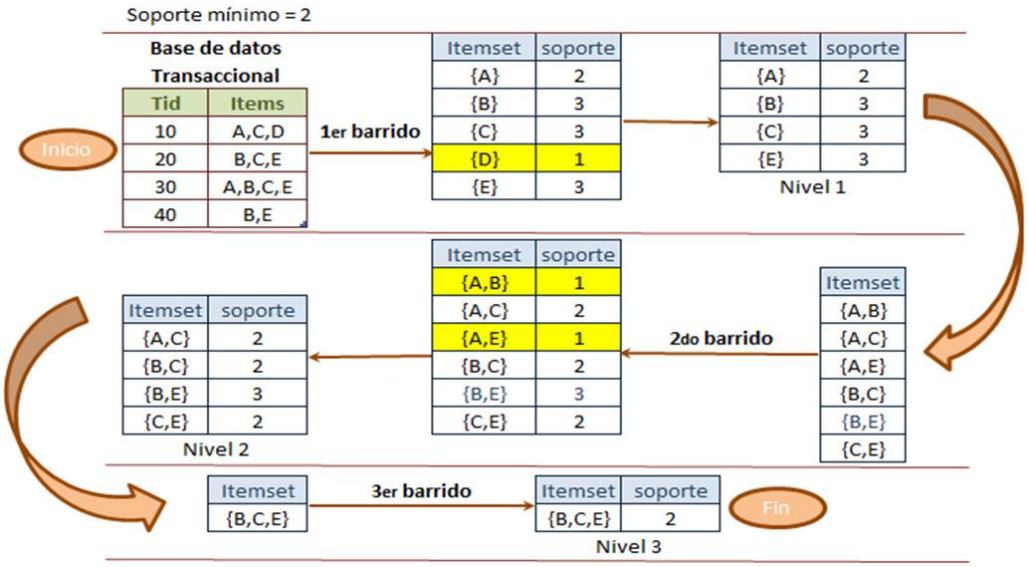

Figura 2.2. Ejemplo de funcionamiento del algoritmo Apriori.

El método que se propone en la presente tesis utiliza el algoritmo Apriori, en combinación con una variante del cubo de trabajo de Han, como parte medular del proceso de descubrimiento de patrones en secuencias simbólicas (sección 4.2).

2.1.2.2. Clasificación

En términos generales, la clasificación es la acción de asignar un objeto a una categoría de acuerdo con las características de éste. En *data mining*, el término clasificación se refiere a la tarea de analizar un conjunto de datos pre-clasificados a fin de aprender un modelo o una función, a partir de datos históricos etiquetados, que puede ser usada para ubicar un dato desconocido en una de varias clases definidas con anterioridad [An 2006].

Los métodos de clasificación que han sido propuestos, en la vasta literatura existente sobre *data mining*, son prácticamente innumerables, pues sobre la base de cada método que ha aparecido, se han propuesto modificaciones, lo que ha dado lugar a la aparición de nuevos métodos, y así sucesivamente. En la presente tesis, se describirán los métodos de los tipos más significativos con sus variantes más relevantes.

A continuación se listan los tipos de métodos de clasificación que destacan en el ámbito del *data mining*, según [Phyu 2009].

- Árboles de decisión,
- Redes Bayesianas,
- Clasificación basada en los k-vecinos más próximos,
- Razonamiento Basado en Casos,
- Algoritmos Genéticos,
- Técnicas de Lógica Difusa.

Árboles de decisión

Los árboles de decisión son clasificadores inductivos, cuya técnica consiste en subdividir recursivamente el conjunto de instancias [Maimon 2010], hasta llegar a un punto en el cual no hay forma de continuar subdividiéndolo o cuando se ha alcanzado un determinado umbral. Estos árboles se componen de ramas y nodos. Hay dos tipos de nodo que son especiales, por el nivel de protagonismo que tienen en este tipo de técnica de clasificación. Se trata del nodo raíz y de los nodos hojas, que son conocidos como raíz y hojas simplemente. La raíz posee una o más ramas de salida y carece de ramas de entrada, por su parte las hojas tienen una única rama de entrada y ninguna de salida, mientras que todos los nodos restantes - también llamados nodos de prueba o internos - poseen una rama de entrada y al menos dos de salida.

Las técnicas de clasificación basadas en árboles de decisión se fundamentan en la estrategia "divide y vencerás". Esto se logra gracias a que cada nodo interno divide el conjunto de instancias en dos o más sub-espacios, a través de una condición que involucra a un atributo en particular. Si el atributo es literal, la condición se refiere a un solo valor; si es numérico, la condición se refiere a un rango de valores.

Un árbol de decisión, posee al menos cuatro componentes:

1. Un resultado categórico también llamado variable dependiente. Esta variable es la que contiene la etiqueta de clase, cuyo valor determinará a qué clase pertenece el dato, sobre la base de los valores de las variables independientes.
2. Variables independientes que guardan alguna relación con la variable independiente.
3. Conjunto de entrenamiento, que posee valores, tanto para las variables independientes como para la variable dependiente.

4. Conjunto de test.

Al igual que otros clasificadores, un problema que se enfrenta al momento de construir un modelo de árbol de decisión, es la necesidad de contar con conjuntos de entrenamiento y de test con tamaños adecuados para garantizar su eficacia.

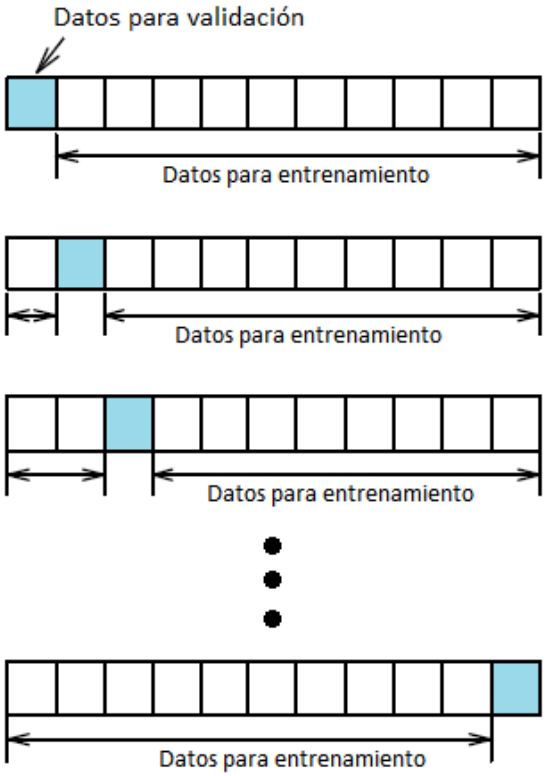

Figura 2.3**.** Partición del conjunto de datos en la validación cruzada.

Existen dos maneras de enfrentar esta necesidad:

- Tomando un porcentaje de los datos de entrada como conjunto de entrenamiento y otro porcentaje como conjunto de test. Estos porcentajes son determinados por el analista de datos.

- Usando una técnica llamada *validación cruzada*, que elimina la necesidad de contar con un conjunto de test separado. Esta técnica divide los datos en k particiones de igual tamaño y realiza k modelos, en cada uno de los cuales toma una de las particiones como conjunto de test y las k - 1 restantes como conjunto de entrenamiento.

La figura 2.3 ilustra la dinámica de la técnica de validación cruzada [Lewis 2000].

Un ejemplo de árbol de decisión es el presentado en [Breiman et al. 1984], que plantea el siguiente problema: "En la Universidad de California, San Diego Medical Center, cuando un paciente con ataque cardiaco es admitido, se miden diecinueve variables durante las primeras veinticuatro horas. Estas variables son presión arterial, edad y diecisiete covariables binarias que sintetizan los síntomas médicos que son considerados importantes para determinar la condición del paciente. Se desea crear un método para identificar los pacientes de alto riesgo".

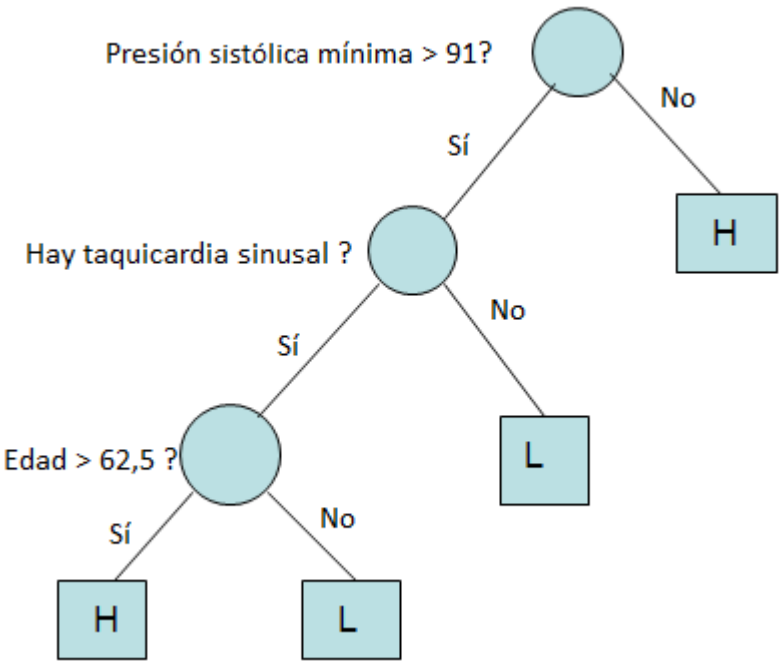

Figura 2.4. Ejemplo de árbol de decisión.

La solución al problema podría ser un árbol de decisión, como el que se presenta en la figura 2.4. En cada uno de los nodos se encuentra un test, cuyas posibles respuestas son "Sí" o "No". En las hojas del árbol están los resultados posibles correspondientes a la variable dependiente, o sea, si es o no es paciente de alto riesgo (la letra H significa alto riesgo y la letra L significa bajo riesgo). La figura 2. muestra un modelo, con tres variables independientes (presión sistólica, taquicardia sinusal y edad), mientras que la variable dependiente es el nivel de riesgo del paciente.

La técnica de los árboles de decisión para clasificar datos tiene aproximadamente cincuenta años de evolución [Lewis 2000]. A lo largo de todos esos años, innumerables algoritmos se han diseñado para la construcción de estos árboles, cada uno de los cuales tiene alguna peculiaridad. A continuación se mencionan algunos de los más importantes:

AID

Automatic **I**nteraction **D**etection: AID es el primer algoritmo para crear modelos de árboles de decisión [Morgan, Sonquist 1963], motivo por el cual tiene mucha importancia en la literatura, aunque en la práctica es muy poco utilizado, pues presenta una relativa inflexibilidad y varias deficiencias analíticas [Struhl 1992].

Este algoritmo se limita a realizar divisiones dicotómicas de la muestra, lo cual tiene consecuencias poco deseables cuando existen atributos con más de dos valores probables, esto provoca que este tipo de variables tengan mayor probabilidad de ser las mejores para realizar la división de la muestra, debido a una mayor varianza. Este algoritmo selecciona el atributo que minimiza la suma de los cuadrados de los errores.

ID3

Iterative **D**ichotomiser Tree: Al igual que AID, ID3 es capaz de generar modelos de árboles de decisión a través de divisiones dicotómicas de sus nodos, por lo cual es ideal para el uso en conjuntos de datos cuyos atributos sean del tipo binario - "sí"/ "no", verdadero/falso, etc. - cuando el conjunto de datos posee atributos con un mayor número de valores, los resultados no son los ideales. La diferencia principal, entre ID3 y AID radica en el criterio usado para seleccionar el atributo utilizado en cada paso para subdividir los datos. ID3 realiza esto usando el concepto de entropía [Shannon 1948]: selecciona el atributo que maximiza la ganancia de información, mediante el cálculo de la entropía [Quinlan 1983].

CART

Classification **A**nd **R**egression **T**rees: Cuando el atributo seleccionado para dividir la muestra en un nodo es continuo, los algoritmos que soportan este tipo de dato utilizan algún método para crear particiones en el rango de valores comprendido entre el máximo y el mínimo, de tal manera que estas sean las óptimas, mediante un proceso llamado *discretización de atributos continuos* que normalmente es univariante, o sea que la segmentación la realiza de manera paralela a los ejes cartesianos. El algoritmo CART, por su parte, utiliza una forma oblicua bivariante de segmentación del espacio, lo que aporta mayor flexibilidad a la hora de dividir rangos de valores continuos. La función usada para medir el nivel de impureza de un nodo se fundamenta en la suma de los residuos al cuadrado [Breiman et al. 1984].

C4.5

Este algoritmo debe su nombre a una versión del programa desarrollado en lenguaje C [Quinlan 1993], como una extensión de ID3. Al igual que otros algoritmos para

construcción de árboles, C4.5 realiza divisiones de la muestra en forma recursiva, admitiendo divisiones de nodos en dos o más grupos. Para la selección del atributo por el cual se divide un nodo, mientras que el algoritmo ID3 utiliza un concepto tomado de la teoría de la información llamado *ganancia de información* [Kumar, Kalia 2012], el mismo que consiste en calcular cuál es el atributo que aporta más información con respecto a la variable independiente, utilizando el concepto de entropía [Shannon 1948], C4.5 usa la ganancia de información normalizada denominada ratio de ganancia que permite contrarrestar el efecto del número de valores posibles de los atributos, lo cual es un problema que tiene el ID3.

GUIDE

Generalized **U**nbiased **I**nteraction **D**etection and **E**stimation: Es un algoritmo para generar árboles de decisión diseñado específicamente para eliminar los sesgos a la hora de seleccionar las variables. GUIDE controla estos sesgos mediante el análisis Chi-cuadrado de residuos y la calibración inicial de las probabilidades más significativas [Loh 2008]. Adicionalmente, permite escoger entre tres roles para cada una de las variables independientes: sólo partición de la selección, sólo regresión o ambos.

FACT

Fast and **A**utomatic **C**lassification **T**ree: A juzgar por los resultados que se obtienen del uno y otro método, FACT es similar a CART, pero con una ventaja en cuanto a velocidad. FACT es un algoritmo capaz de manipular variables categóricas ordenadas o no ordenadas, hacer tratamiento de valores faltantes y costos de clasificación errónea. Por otro lado, además de la estructura del árbol, produce una clasificación de las variables según su importancia y una estimación del error [Loh, Vanichsetakul 1986].

SUPPORT

Smoothed and **U**nsmoothed **P**iecewise **P**olynomial **R**egression **T**rees: SUPPORT es un algoritmo para la creación de árboles de decisión basados en estimaciones. Cada estimación se compone de pedazos, cada uno de los cuales es obtenido encajando una regresión polinómica en una subregión del espacio de datos, cuya segmentación se la realiza recursivamente. En caso de que se requiera un suavizado de la estimación, los pedazos polinomiales obtenidos pueden ser unidos mediante un promedio ponderado y entonces se obtiene la estimación suavizada mediante tres pasos. Primero, el espacio regresivo es segmentado recursivamente hasta que los datos en cada pedazo encajan con un polinomio de orden fijo. A continuación, los datos del entorno de cada segmento son

encajados en el polinomio. Por último, se realiza una estimación final de la función de regresión mediante un promedio de los pedazos del polinomio, usando funciones ponderadas suavizadas, las cuales tienden rápidamente a cero cuando están fuera de su partición asociada [Chaudhuri, Dayal 1997].

QUEST

Quick **U**nbiased **E**fficient **S**tatistical **T**ree: Este método de clasificación elimina los sesgos en la selección de variables, al igual que GUIDE, pero además supera en rendimiento al algoritmo CART para la partición de variables continuas. QUEST utiliza pruebas de hipótesis para seleccionar la variable de división, esto es, análisis de varianza F-tests para las variables continuas y pruebas Chi-cuadrado para variables categóricas; la variable con la menor probabilidad de importancia es seleccionada para dividir el nodo [Loh 2008].

CRUISE

Classification **R**ule with **U**nbiased **I**nteraction **S**election and **E**stimation: CRUISE es un algoritmo cuyo rendimiento es superior al de sus antecesores y basa su fortaleza en el uso de modelos de discriminantes lineales bivariantes lo cual, opuestamente a los métodos univariantes, permite generar árboles menos complejos y con una reducción considerable del tiempo de generación. El modelo de discriminante es aplicado a todos los nodos y a través de todo el proceso de generación del árbol, a diferencia de otros algoritmos que solo lo aplican en los nodos terminales del árbol podado. En el caso de CRUISE, la poda del árbol se efectúa usando los costes de clasificación errónea del nodo modelado. La segmentación de los datos se realiza mediante particiones ortogonales a los ejes, a diferencia de los métodos antecesores más recientes que usan orientaciones arbitrarias; la justificación para ello es que se conserva la facilidad de interpretación del árbol [Kim, Loh 2003].

CHAID

CHi-squared **A**utomatic **I**nteraction **D**etection: CHAID es una variante de AID en el sentido de que permite la división de la muestra hasta en quince grupos, dependiendo de la cantidad de valores posibles de la variable dependiente, en cualquier punto del árbol. Además, la selección de la variable de división se realiza usando un procedimiento que ajusta la importancia de la variable observada tomando en cuenta el número de valores posibles de esta [Struhl 1992]. Esto último da a todas las variables las mismas oportunidades de aparecer en el análisis. El nombre de este algoritmo proviene del criterio

usado para realizar la división de la muestra en grupos, esto es, se usa el criterio de división Chi-cuadrado, más específicamente, usa el valor p del Chi-cuadrado [Ritschard 2010].

Para concluir con la descripción de los árboles de decisión, cabe decir que la gran ventaja de éstos, como estructuras de clasificación, es su relativa facilidad de interpretación, pero al mismo tiempo su mayor debilidad es el alto grado de complejidad que pueden alcanzar cuando el número de atributos es elevado y la existencia de atributos continuos y/o categóricos sumados a la existencia de muchas clases. En los casos más complejos, los costes computacionales para generar un árbol de decisión pueden ser demasiado elevados.

Clasificación Bayesiana

El uso de teorías estadísticas bayesianas en el ámbito del *data mining* ha dado lugar a una aplicación de las mismas en lo que se denominan los Clasificadores Bayesianos, que se describen en la presente sub-sección.

Un clasificador bayesiano es un clasificador estadístico que es capaz de determinar la probabilidad de que un ejemplo pertenezca a una determinada clase.

Los métodos bayesianos de clasificación presuponen que los ejemplos de una clase no dependen de las ejemplos de otras clases, a lo que se denomina independencia de clase condicional, lo cual habitualmente está muy lejos de la realidad [Bashir et al. 2006]. Para realizar la clasificación, utilizan el teorema de Bayes que se muestra en la expresión 2.3.

$$P(Q|T) = \frac{\mathrm{P(T|Q)P(Q)}}{\mathrm{P(T)}}$$

Expresión 2.3. Teorema de Bayes

Donde:

T es un ejemplo con n atributos,

Q es una *hipótesis*, por ejemplo: El paciente T tiene un ligamento roto,

P(Q|T) es la probabilidad de que Q se cumpla, conociendo T (probabilidad a posteriori de Q),

P(T|Q) es la probabilidad de que T se dé, suponiendo Q (*probabilidad a posteriori de T*),

P(Q) es la probabilidad de que Q se cumpla a priori. (*Probabilidad a priori de Q*),

P(T) es la probabilidad de que T se dé a priori. (*Probabilidad a priori de T*).

La clasificación bayesiana se detalla a continuación:

Sea E un conjunto de m datos de entrenamiento.

Sea la el ejemplo $T = (t_1, t_2,..., t_n)$ formada por los valores de los n atributos $A_1, A_2,..., A_n$.

Sean $C_1, C_2,...,C_s$ las clases a las que pertenecen las m ejemplos de E.

T pertenece a la clase C_i, si y solo si $P(C_i|T) > P(C_j|T)$ para $1 \leq j \leq s,\ j \neq i$

Aplicando el teorema de Bayes (Expresión 2.3) y como P(T) es constante, entonces $P(T|C_i)P(C_i)$ debe ser maximizado.

Se conoce que $P(C_i) = |C_{iE}|/|E|$ $|C_{iE}|$ es el número de ejemplos de la clase C_i en E.

Para reducir el costo computacional, se asume la independencia condicional de clases:

$$P(T|C_i) = \prod_{k=1}^{n} P(t_k|C_i)$$

Expresión 2.4. Fórmula general de la probabilidad para la clase C_j

Las probabilidades $P(t_1|C_i), P(t_2|C_i), \dots, P(t_n|C_i)$ se calculan según el tipo del atributo A_k

- Si A_k es categórico:

$$P(t_k|C_i) = \frac{|P(t_k|C_i)|}{|C_{iE}|}$$

Expresión 2.5. Probabilidad si A_k es categórico

- Si A_k es continuo: se asume una distribución gaussiana para A_k con media μ y desviación estándar σ y se tiene que

$$P(t_k|C_i) = g(t_k, \mu_{Ci}, \sigma_{Ci})$$

Expresión 2.6. Probabilidad si A_k es continuo

donde $g(t_k, \mu_{Ci}, \sigma_{Ci}) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(t_k-\mu_{Ci})^2}{2\sigma_{Ci}^2}}$.

Según la literatura, los clasificadores bayesianos han sido comparados con otras técnicas de clasificación y los resultados no resultan tan desalentadores, pues, a pesar de su relativa sencillez, logran un rendimiento similar a los árboles de decisión [AL-Nabi, Ahmed 2013] y una mayor precisión que las redes neuronales [Mollazade et al. 2012] y que k-vecinos más próximos [Sreemathy, Balamuguran 2012]. En el dominio de la clasificación de textos, el clasificador bayesiano ha sido extensamente usado y con mucho éxito [De Campos, Romero 2009] [Sreemathy, Balamuguran 2012]. Por tanto, se trata de una técnica muy importante dentro del contexto del *data mining*.

Clasificación de los k-vecinos más próximos

El método de clasificación de los k-vecinos más próximos (KNN por sus siglas en inglés, k-Nearest Neighbors) fue descrito por primera vez a inicios de los años 50. Este método requiere mucho tiempo de procesamiento, cuando el conjunto de datos de entrenamiento es grande. Por ello solamente pudo ganar popularidad en los años 60, cuando ya se disponía de una mayor capacidad de cálculo y almacenamiento en los ordenadores. Esta técnica ha sido muy usada en el reconocimiento de patrones [Han et al. 2012].

Para el cálculo de la proximidad de los objetos, el método kNN utiliza el concepto de distancia, que será detallado en la sección 2.1.2.3, por lo tanto, solamente se hará referencia a la distancia euclidiana para fines de descripción de este método.

La distancia euclidiana entre dos objetos p-dimensionales tiene la siguiente forma:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{p} (x_{ir} - x_{jr})^2}$$

Expresión 2.7. Distancia Euclidiana

Se presupone que existe un conjunto de datos de entrenamiento, de los cuales se conoce a qué clase pertenecen. El clasificador kNN es capaz de clasificar un objeto nuevo, asignándolo a la clase más común entre los k objetos más próximos al objeto nuevo que se desea clasificar. En caso de que k = 1, la clase elegida será aquella a la que pertenezca el objeto más próximo a aquel que se quiere clasificar.

La característica fundamental de este clasificador es que guarda en memoria los objetos del conjunto de entrenamiento y cuando se requiere la clasificación de uno nuevo, entonces se realiza el proceso de aprendizaje, lo cual significa que el mayor coste computacional del método se encuentra en la fase de clasificación. Por este motivo se dice que esta forma de aprendizaje pertenece al tipo de aprendizaje denominado aprendizaje perezoso.

A continuación se detalla el proceso para clasificar una instancia nueva con el clasificador kNN:

Sea D el conjunto de instancias de entrenamiento, con n tuplas p-dimensionales, es decir, x_1, x_2, ..., x_n tuplas que pertenecen a D y $x_i \in R^p$ con $1 \leq i \leq n$

Sean y_1, y_2,..., y_n las clases a las que pertenecen x_1, x_2,..., x_n respectivamente

Cuando llega una nueva instancia no clasificada x_q, se constuye un vector con las distancias entre x_q y cada una de las n instancias del conjunto de entrenamiento. La clase a la que se

asigna x_q será aquella a la que pertenezcan la mayor parte de las k instancias más próximas a x_q. En caso de empate, se seleccionará la clase que tenga mayoría en la muestra.

Como se puede observar, este método es bastante sencillo de implementar pero puede resultar muy costoso en el momento de clasificar una nueva instancia. Tiene un pobre desempeño si se lo compara con los árboles de decisión y los clasificadores bayesianos [AL-Nabi, Ahmed 2013]; además, su desempeño y grado de precisión disminuye conforme aumenta el tamaño de la muestra [Raikwal, Saxena 2012].

Razonamiento Basado en Casos

El razonamiento basado en casos o CBR - por sus siglas en inglés Case Based Reasoning - predice la clase de un caso, sobre la base de la similitud entre éste y los casos clasificados con anterioridad, denominados casos resueltos [Armengol 2007]. Los casos no necesariamente se parecen a las instancias de la clasificación de los k-NN, sino que pueden ser descripciones simbólicas complejas. Algunos ejemplos de casos se pueden encontrar en los siguientes dominios [Ke. Wu 2010]:

- **Help Desk** - Solución de problemas de servicio al cliente;
- **Ingeniería** - Diseño técnico;
- **Leyes** - Marcos legales para resolver conflictos;
- **Medicina** - Las historias y tratamientos de los pacientes se usan para diagnosticar y tratar nuevos pacientes.

Algunas definiciones relativas con los casos, usadas en la técnica CBR son:

- *Caso*: generalmente, un caso denota una situación problemática.
- *Caso pasado*, *caso previo*, *caso resuelto* o *caso guardado*: una situación ya experimentada que ha sido capturada y aprendida de tal manera que puede ser reutilizada para la solución de futuros problemas.
- *Caso nuevo*: Se refiere a un caso no resuelto y describe un nuevo problema.

La figura 2.5 muestra esquemáticamente el funcionamiento de un sistema de Clasificación que utiliza la técnica de CBR. En primer plano se observa la Base de Casos, que almacena los problemas resueltos con anterioridad y que se convierte en los casos de entrenamiento en el momento de clasificar un nuevo caso. Este nuevo caso, al ingresar en el clasificador CBR, es comparado con los casos que se encuentran almacenados en la base de casos y, a través de un mecanismo de estimación de distancias, se emite un resultado que puede consistir en un conjunto

de casos resueltos más próximos (más parecidos), los cuales quedarían a consideración del usuario para su análisis y decisión final.

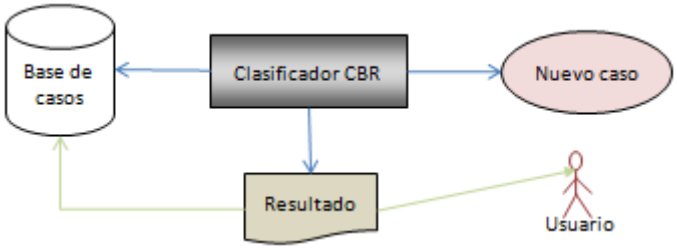

Figura 2.5. Esquema del método CBR de clasificación.

El método CBR de clasificación es cíclico, lo cual significa que un nuevo caso que pasa por el proceso de clasificación, una vez resuelto, pasa a formar parte de la base de casos y se convierte automáticamente en nuevo conocimiento [Aamodt, Plaza 1994] que será aplicado en la solución de casos futuros.

En el ámbito científico y comercial, el razonamiento basado en casos CBR posee una gran importancia, pues, al tratarse de la combinación de cuatro áreas de investigación [Richter, Aamodt 2006] (ver figura 2.6), da lugar a que exista interés en encontrar nuevos matices para las técnicas y métodos desarrollados hasta la actualidad y, además, existen muchas posibilidades que están pendientes de investigación como, por ejemplo, el uso de la similitud simbólica para la resolución de problemas usando técnicas CBR [Armengol 2007].

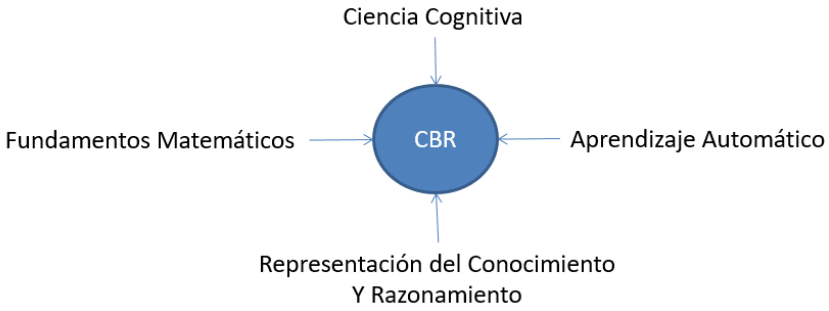

Figura 2.6. Áreas del CBR

Algoritmos Genéticos

Un algoritmo genético (GA) es un método estocástico de búsqueda inspirado en el proceso de evolución biológica para la solución de problemas. Es una técnica de búsqueda para encontrar soluciones aproximadas a problemas de optimización y búsqueda [Sivanandam, Deepa 2008]. De hecho, la técnica de algoritmos genéticos forma parte de un grupo de técnicas que parten del mismo principio, la teoría darwiniana de la evolución de las especies, pero que difieren en la manera de modelarlo. Se trata de la Computación Evolutiva, cuyas cuatro técnicas asociadas son:

- Algoritmos Genéticos;
- Estrategias Evolutivas;
- Programación Genética; y,
- Programación Evolutiva.

Cada una de estas técnicas tiene su particular importancia; sin embargo, abordaremos solamente la de los algoritmos genéticos, que ha sido aplicada con relativo éxito en la tarea de clasificación en *data mining*, pues la ventaja de los GAs sobre otras técnicas de clasificación se hace más evidente cuando el espacio de búsqueda es muy grande [Gundogan et al. 2004].

A continuación, se introducen algunos conceptos relacionados con los GAs y se presentan ejemplos de sistemas de clasificación desarrollados con esta técnica.

Un GA es básicamente un ciclo iterativo en el cual se busca maximizar o minimizar, dependiendo del caso, una función de evaluación, cuya estructura depende de la aplicación. El ciclo comienza con la generación de una población inicial. Una población es una colección de individuos codificados que son considerados posibles soluciones al problema. La generación de la población inicial incluye un proceso de codificación que, comúnmente, se reduce a la conversión de los datos reales en secuencias de bits, cuya longitud depende del tamaño de la población. Cada bit es análogo al gen de la teoría genética y toda la secuencia de bits se corresponde con un cromosoma y cada bit ocupa un lugar, llamado locus, dentro del cromosoma.

A todos los cromosomas de la población se les asigna un valor de la llamada función de aptitud, fitness function en inglés, que se asocia con la capacidad de adaptación de cada individuo y, por tanto también, con la mayor o menor probabilidad de reproducirse, pues, según la teoría de la conservación de las especies, los individuos más adaptados son los que conservan la especie mientras que los menos adaptados propician su desaparición. Estos valores de la función de aptitud permitirán realizar la selección de los individuos que mejor resuelvan el problema. De hecho, aquel que minimice la función de aptitud a un nivel inferior a un umbral, se convertirá automáticamente en la solución buscada.

Si la solución no está entre los individuos seleccionados, se procede a aplicar el operador de cruce, que combina los individuos entre sí, para generar nuevos individuos mejor adaptados y que, por tanto, se espera que converjan hacia la solución. En esta operación se utilizan ciertas heurísticas que permiten determinar la dirección de la búsqueda. Al grupo de individuos "mejorados" se les aplica, opcionalmente, un operador de mutación, lo cual puede redundar en un mejoramiento superior aún al ya logrado con el operador de cruce.

Esta nueva versión de la población es nuevamente evaluada, e inicia así un nuevo ciclo del proceso iterativo que terminará cuando la evaluación determine que se ha alcanzado el umbral previsto.

En la figura 2.7. se muestra un diagrama de flujo simplificado de un algoritmo genético, según [De Jong et al. 1994].

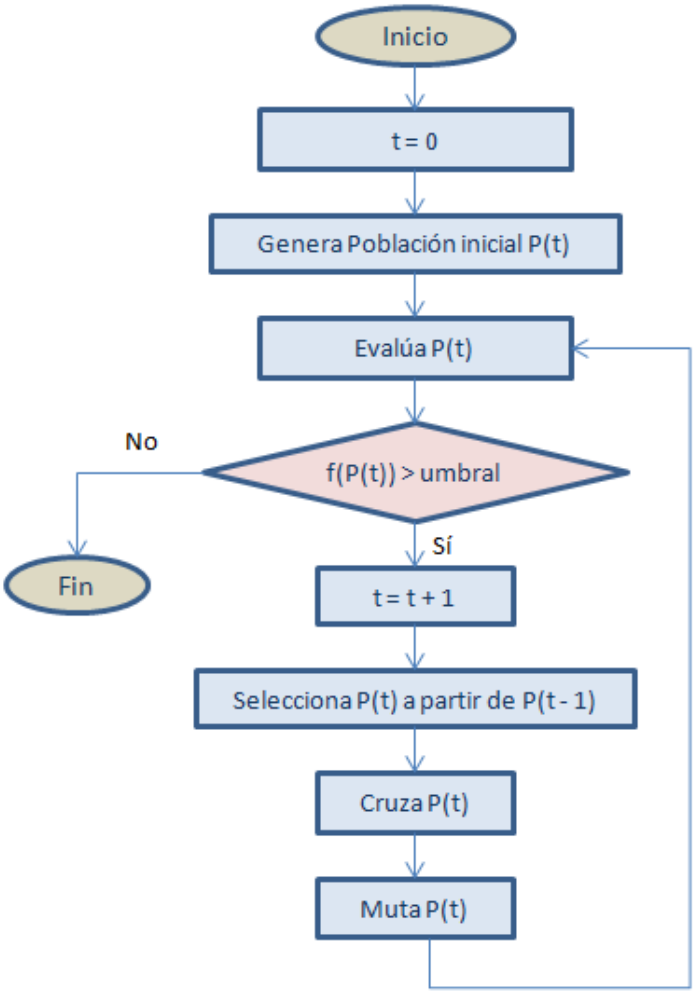

Figura 2.7. Esquema simplificado de un Algoritmo Genético

Entre las innumerables aplicaciones que se han hecho de los GAs para clasificación en *data mining*, se detallan a continuación cuatro que ilustran los conceptos mencionados y proveen una visión de las variaciones entre una implementación y otra:

GABIL [De Jong et al. 1994]

- Un individuo es una secuencia de longitud variable que representa un conjunto de reglas de longitud fija.

- Para la mutación utiliza la inversión de bits tradicional.

- Para el cruce, los puntos correspondientes de cruce de los dos padres deben coincidir semánticamente.

- La función de aptitud contiene solamente el porcentaje de individuos correctamente clasificados por un conjunto individual de reglas.

GIL [Janikow 1993]

- Un individuo es un conjunto de reglas, pero sus valores de atributo son codificados directamente.

- GIL tiene operadores genéticos especiales para manipular conjuntos de reglas y condiciones de reglas.

- Los operadores pueden realizar generalizaciones, especializaciones u otras operaciones.

- Además de la exactitud, la función de evaluación de GIL incluye también la complejidad de un conjunto de reglas.

COGIN [Greene, Smith 1993]

- En un punto cualquiera de la búsqueda, el modelo del sistema es representado por una población de reglas de longitud fija.

- El tamaño de la población varía de un ciclo a otro, en función de la restricción de cobertura que se aplique.

- La función de clasificación se calcula sobre la base de la ganancia de información de la regla R y una penalización del número de clasificaciones erróneas realizadas por R.

- Las medidas de entropía que se usan para calcular la ganancia de la regla R, se basan en el número de individuos.

- El método de codificación de COGIN no puede evaluar la entropía de una partición completa, formada por la clasificación, a pesar de que la regla R coincide o no coincide.

GA approach [Noda et al. 1999]

- La función de aptitud está formada por dos partes:

  1. La que mide el grado de interés de la regla.

2. La que mide la precisión de la predicción.

- El cálculo del grado de interés se basa en la siguiente idea:

Mientras sea mayor la frecuencia relativa del valor a ser predicho por el consecuente, menor es el grado de interés de tal valor. En otras palabras, mientras más raro es el valor de un atributo objetivo, mayor es el grado de interés de la regla que lo predice.

Estos han sido solamente unos pocos ejemplos de los sistemas de clasificación mediante algoritmos genéticos existentes en la literatura; otros pueden ser encontrados, por ejemplo, en [Minaei-Bidoli 2003] y [Gudogan 2004].

Técnicas de Lógica Difusa

Así como en el mundo real no todo es blanco o negro, sino que existen niveles de gris, colores y tonalidades, según la teoría de conjuntos difusos no todos los ejemplos pertenecen o no a un conjunto sino que tienen un grado de pertenencia, y sobre este concepto gira dicha teoría. En esta sub-sección se hará una breve introducción a la teoría de conjuntos difusos y luego se presentará un ejemplo de cómo, usando esos principios, se puede realizar clasificación. Para distinguir a los conjuntos clásicos de los conjuntos difusos; a aquellos se los denomina conjuntos nítidos [Chen, Pham et al. 2001].

Sea A = {u} un conjunto nítido donde u ϵ U:

Sea X_A la función característica de A: $X_A(u) = \begin{cases} 1\ si\ x\ \in\ A \\ 0\ si\ x\ \notin A \end{cases}$

Esto significa que u puede ser o no ser elemento de A.

Según la teoría de conjuntos difusos, un elemento puede pertenecer o no pertenecer a un conjunto, pero además existe una tercera posibilidad y es que tenga un cierto grado de pertenencia al conjunto. A continuación, se presenta formalmente este enunciado:

Sea F un conjunto difuso y sea u ϵ U.
A cada u le corresponde un valor x = [0,1] que representa el grado de pertenencia de u a F.
Sea $\mu_F(u)$ la función de pertenencia de u a F:

$$\mu_F(u) = \begin{cases} 0\ si\ u \notin F \\ 1\ si\ u \in F \\ 0 < \mu_F(u) < 1\ si\ u\ es\ miembro\ difuso\ de\ F \end{cases}$$

Expresión 2.8. Grado de pertenencia

Los operadores difusos de unión e intersección se definen de la siguiente forma:

Sean A y B dos conjuntos difusos con universo U:

$$\forall u \in U \begin{cases} \mu_{A\cup B}(u) = max(\mu_A(u), \mu_B(u)) \\ \mu_{A\cap B}(u) = min(\mu_A(u), \mu_B(u)) \end{cases}$$

Expresión 2.9. Operadores difusos de Unión e Intersección

Los Sistemas Difusos se basan en reglas difusas para la toma de decisiones, con lo cual se alcanzan mejores resultados que con los conjuntos nítidos. La figura 2.8 muestra, de forma esquemática, los componentes principales de un sistema difuso.

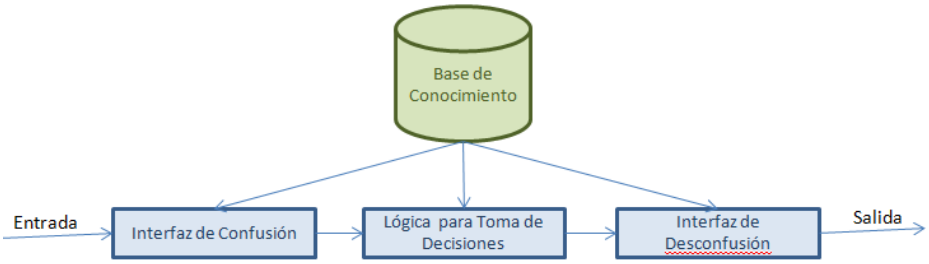

Figura 2.8. Esquema general de un Sistema Difuso

La Interfaz de fuzzificación tiene tres funciones: Leer los datos de entrada; Escalar los datos para que encajen en el universo adecuado [Lee 1990]; y Convertir los datos de entrada en variables lingüísticas apropiadas para que sean manipuladas por conjuntos difusos.

La Base de conocimiento es una base de datos que contiene conocimiento experto en forma de reglas si - entonces, que pueden ser convertidas en Sugerencias Difusas y, a partir de éstas son construidas las Reglas Difusas.

La Lógica de toma de Decisiones es el núcleo del sistema y se encarga de interpretar las sugerencias y, sobre esa base, realiza la toma de decisiones.

A manera de un breve ejemplo de cómo se construye un sistema de clasificación basado en lógica difusa, a continuación se muestra la implementación desarrollada por [Svensson 2011], incluyendo únicamente los detalles más relevantes.

Se parte de un conjunto de datos, muy conocido en la literatura de *data mining* [Fisher 1936]. Se trata de la flor llamada iris, cuyas especies son tres - setosa, versicolor y virgínica - y el problema consiste en lograr una forma de clasificarlas considerando sus cuatro características: longitud del sépalo, longitud del pétalo, altura del sépalo y altura del pétalo.

Por otro lado, se cuenta con el conocimiento experto, con el que se han logrado obtener cuatro reglas expresadas en forma de sentencias si - entonces:

Regla 1. si (x_1 = corto ∨ largo) y (x_2 = medio ∨ largo) y (x_3 = medio ∨ largo) y (x_4 = medio) entonces iris = versicolor.

Regla 2. si (x_3 = corto ∨ medio) y si(x_4 = corto) entonces iris = setosa

Regla 3. si (x1 = corto ∨ medio) and (x3 = largo) and (x4 = largo) entonces iris = virgínica

Regla 4. si (x1 = medio) and (x2 = corto ∨ medio) y (x3 = corto) and (x4 = largo) entonces iris = versicolor

donde: x_1 es longitud del sépalo, x_2 es altura del sépalo, x_3 es longitud del pétalo, x_4 es altura del pétalo, max(i) y min(i) son los valores máximo y mínimo, respectivamente, que puede alcanzar el i-ésimo atributo dentro de una categoría determinada.

Para la normalización de atributos se usa la siguiente expresión.

$$x_i' = \frac{x_i - \min(i)}{\max(i) - \min(i)}$$

Expresión 2.10. Normalización de atributos

A cada atributo del conjunto de datos se le asigna uno de los tres términos lingüísticos: corta, media o larga. La figura 2.9 muestra la función de pertenencia para el presente caso.

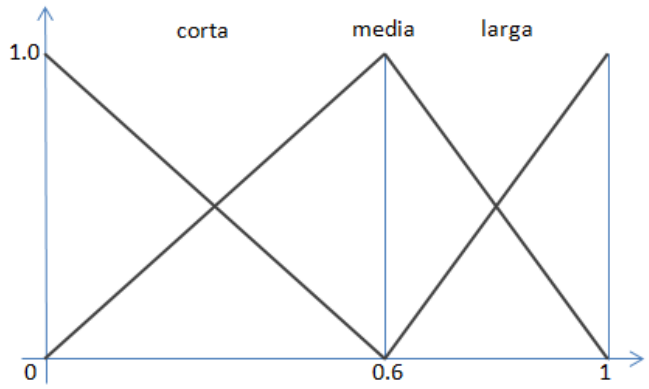

Figura 2.9. Función de pertenencia para los conjuntos difusos Corta, Media y Larga

Interfaz de Fuzzificación

Durante el proceso de fuzzificación, cada uno de los parámetros de entrada será asignado a uno de los tres conjuntos difusos, usando la función de pertenencia adecuada.

La función de pertenencia de la figura 2.9 permite extrapolar tres fórmulas que permitirán determinar a cuál de los tres conjuntos difusos pertenecen los datos de entrada.

$$\mu_{corta}(x) = \begin{cases} 0\ si\ x < 0 \\ \left(\frac{0.6 - x}{0.6}\right)\ si\ 0 \leq x \leq 0.6 \\ 0\ si\ x > 0.6 \end{cases}$$

Expresión 2.11. Función de pertenencia para el conjunto difuso *corta*

$$\mu_{media}(x) = \begin{cases} 0\ si\ x < 0 \\ \left(\frac{x}{0.6}\right)\ si\ 0 \leq x \leq 0.6 \\ \left(\frac{1 - x}{0.4}\right)\ si\ 0.6 \leq x \leq 1 \\ 0\ si\ x > 1 \end{cases}$$

Expresión 2.12. Función de pertenencia para el conjunto difuso *media*

$$\mu_{larga}(x) = \begin{cases} 0\ si\ x < 0.6 \\ \left(\frac{x - 0.6}{0.4}\right)\ si\ 0.6 \leq x \leq 1 \\ 0\ si\ x > 1 \end{cases}$$

Expresión 2.13. Función de pertenencia para el conjunto difuso *larga*

Base de Conocimiento

Las reglas lógicas de tipo si <condición> entonces <valor>, se convierten en reglas difusas, usando la teoría de conjuntos difusos, por lo tanto, el y lógico se convierte en *min*() y el o lógico se convierte en *max*().

Por ejemplo, la Regla 2 sería transformada en regla difusa de la siguiente manera:

si (x_3 = corta ∨ media) y (x_4 = corta) → *min*(*max*(x_{3corta},x_{3media}),x_{4corta})
entonces iris = setosa

donde x_{ntipo} es el valor de la función; x_i es el i-ésimo atributo de la función de pertenencia.

El resultado es un valor en el intervalo [0,1] que representa el grado con el cual una entrada pertenece a un determinado conjunto difuso.

La expresión 2.14 muestra que la salida es un conjunto donde a, b, c son los grados de pertenencia de la entrada a cada una de las especies.

$$F_R = \left\{\frac{a}{setosa}, \frac{b}{versicolor}, \frac{c}{viginica}\right\}$$

Expresión 2.14. Función de pertenencia para el conjunto difuso *larga*

Toma de Decisiones

En virtud de que la salida de la función de pertenencia para cada regla da como resultado un conjunto de valores, de forma sencilla se pueden juntar las funciones de pertenencia de las cuatro reglas - R1, R2, R3 y R4 -, combinando las salidas, con lo cual se obtiene la expresión 2.15:

La expresión 2.15 indica que la función de pertenencia con valores más altos será la que aporte a F, consecuentemente, los parámetros de entrada serán considerados como pertenecientes a la especie con el más alto grado de pertenencia.

$$F = F_{R1} \vee F_{R2} \vee F_{R3} \vee F_{R4} \left\{ \frac{A}{setosa}, \frac{B}{versicolor}, \frac{C}{viginica} \right\}$$

Expresión 2.15. Combinación de las cuatro reglas

La lógica difusa tiene ventajas y desventajas que deben ser consideradas al momento de decidir la técnica a ser usada para clasificación en *data mining*. Un estudio interesante de este importante aspecto puede ser encontrado en [De Reus 1994].

2.1.2.3. Agrupamiento o Clustering

Uno de los grandes objetivos del *data mining* es encontrar similitudes entre objetos y agruparlos de acuerdo con el grado de afinidad entre ellos. El Agrupamiento, conocido como *clustering* en la literatura, se puede llevar a cabo mediante un conjunto de técnicas que permiten cumplir con el objetivo planteado. La comunidad científica ha realizado un enorme trabajo para encontrar métodos que mejor realicen esta tarea y el *clustering* ha sido aplicado en un gran rango de aplicaciones con grandes bases de datos con muchos atributos [Berkin 2006]. En este apartado se abordará el tema a partir del concepto formal y de una taxonomía de los métodos desarrollados, para luego hacer una descripción de los más importantes, según la literatura científica.

Clustering es el proceso de agrupamiento de una colección de objetos (usualmente representados como puntos en un espacio multidimensional) en subconjuntos de objetos similares [Lingras, Huang 2005]. Los objetos de un conjunto o *cluster* generado son muy similares entre sí pero, al mismo tiempo, son distintos de los objetos de otro conjunto. Los algoritmos de *clustering* son considerados como no supervisados, lo cual significa que no requieren de un conocimiento a priori sobre los conjuntos que se van a generar, contrariamente a lo que sucede con los algoritmos de clasificación, en los cuales se parte de un conocimiento previo de las clases; por esta razón, al *clustering* también se lo conoce como "clasificación no supervisada" [Han et al. 2012].

A continuación, se presenta una taxonomía de los principales algoritmos de *clustering* encontrados en la literatura:

- Métodos Jerárquicos
  - Algoritmos Aglomerativos
  - Algoritmos Divisivos
- Métodos de Reubicación de Particiones

- *Clustering* Probabilísticos
- Métodos de las K-medoides
- Métodos de las K-medias

➢ Métodos de Particionamiento Basado en Densidad

- *Clustering* de Conectividad Basada en Densidad
- *Clustering* de Funciones de Densidad

➢ Métodos Basados en Cuadrícula

➢ Métodos Basados en Coocurrencia de Datos Categóricos

➢ Otras Técnicas de *Clustering*

- *Clustering* Basado en Restricciones
- Particionamiento de Grafos

➢ Algoritmos de *Clustering* Escalable

➢ Algoritmos Para Datos de Alta Dimensionalidad

- *Clustering* de Subespacio
- Técnicas de Co-*Clustering*

Previamente a la presentación de los métodos de *clustering* más relevantes existentes en la literatura especializada, a continuación se definen algunos conceptos fundamentales relacionados con el *clustering*, a fin de formalizar la presentación.

Sea un conjunto de datos X con m objetos multidimensionales con n atributos, donde m, n ϵ N. Para fines de implementación, X es representado por medio de la Matriz de Datos, que se define como una matriz con m filas, una por cada objeto, y n columnas, una por cada atributo. Generalmente, los valores de atributo vienen expresados en unidades que los hacen incompatibles entre sí, lo cual hace que sea improcedente la realización de operaciones entre ellos. Para solucionar este inconveniente, se procede a la normalización, que no es más que colocar todos los datos con sus valores de atributos en la misma escala. Las siguientes son dos maneras muy comunes de normalización:

Sea x_{ij} el valor del j-ésimo atributo del i-ésimo objeto de la matriz de datos original, entonces el valor normalizado x'_{ij} será:

$$x'_{ij} = \frac{x_{ij}}{max_i|x_{ij}|}$$

Expresión 2.16. Normalización de valores de atributos (1)

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

Expresión 2.17. Normalización de valores de atributos (2)

En las expresiones 2.16 y 2.17:

$max_i|x_{ij}|$ es el valor máximo del atributo j entre todos los objetos.
σ_j es la desviación estándar de los valores del atributo j en todos los objetos.
μ_j es la media de los valores del atributo j de todos los objetos.
Los valores de x' obtenidos estarán en el intervalo [-1,1].

Otro elemento importante para la búsqueda de grupos es la Matriz de Similitud S o la Matriz de Disimilitud D. Para evitar confusiones, comúnmente se hace referencia a la Matriz de Proximidad P o Matriz de Conectividad que corresponde a cualquiera de las dos anteriores. Estas son matrices mxm que contienen las distancias entre los objetos de X, tomados por pares. Por ejemplo, en la matriz de proximidad P, el elemento p_{ij} representa la distancia entre los objetos x_i y x_j. Su importancia en el proceso de *clustering* resulta crucial para evaluar la pertenencia o no de un objeto a un determinado grupo.

Existen innumerables medidas de distancia en la literatura. Se mencionan aquí algunas de las más utilizadas. La naturaleza de un dominio en particular puede hacer que se decida la utilización de una determinada medida ya existente o incluso puede que sea necesario la creación de una nueva. No es raro ver que para la solución de un problema totalmente nuevo aparezca asociada una nueva manera de calcular la distancia entre los objetos involucrados.

Distancia de Minkowsky:

$$P_{ij} = \left(\sum_{k=1}^{d} |x_{ik} - x_{jk}|^r \right)^{1/r}$$

Expresión 2.18. Distancia de Minkowsky

Donde d es la dimensionalidad de los objetos de datos, x_{ik} y x_{jk} son los valores del atributo *k* para los objetos x_i y x_j respectivamente, mientras que *r* es un parámetro. Al ser una distancia cuyo cálculo depende de un parámetro, cada valor del mismo llegará a diferentes resultados, lo cual significa que existen numerosas derivaciones de esta fórmula. Los casos más conocidos son para *r = 1* (Distancia de Manhatan), *r = 2* (Distancia Euclidiana) y $r \to \infty$ (Distancia Suprema)

Similitud del coseno:

Esta es una medida basada en vectores. Es utilizada especialmente cuando los datos son dispersos, esto es, cuando los valores nulos no tienen significación en el dominio. A continuación se presentan dos de las formas de calcular esta distancia:

Sean X e Y dos vectores que representan dos objetos: $X = \langle x_1, \dots, x_n \rangle$ y $Y = \langle y_1, \dots, y_n \rangle$

$$sim(X, Y) = X \cdot Y = \sum_i x_i \times y_i$$

Expresión 2.19. Similitud del coseno (Producto interno)

$$sim(X, Y) = \frac{X \cdot Y}{\|X\| \times \|Y\|} = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x^2} \times \sqrt{\sum_i y^2}}$$

Expresión 2.20. Similitud del coseno (Producto interno normalizado)

Donde: $\|X\|$ y $\|Y\|$ son las normas de X e Y respectivamente.

Coeficiente de Jaccard Extendido:

$$CJE(X, Y) = \frac{X \cdot Y}{\|X\|^2 + \|Y\|^2 - X \cdot Y} = \frac{\sum_i (x_i \times y_i)}{\sum_i x^2 + \sum_i y^2 - \sum_i (x_i \times y_i)}$$

Expresión 2.21. Coeficiente de Correlación de Jaccard Extendido

Este coeficiente es usado para comparar documentos, se parece mucho a la similitud del coseno. Cuando su valor es 1 significa que los datos son iguales, excepto en cuestión de magnitud. En el caso en que el valor es 0, los datos son totalmente diferentes.

Coeficiente de Correlación de Pearson:

$$P = \frac{cov(X, Y)}{\sigma_X \times \sigma_Y} = \frac{\sum_i (x_i - \mu_x) \times \sum_i (y_i - \mu_y)}{\sqrt{\sum_i (x_i - \mu_x)^2 \times \sum_i (y_i - \mu_y)^2}}$$

Expresión 2.22. Coeficiente de Correlación de Pearson

Este coeficiente es muy útil cuando los datos están relacionados entre sí de forma lineal. El valor de este coeficiente está en el rango [-1, 1] donde 1 significa que existe una fuerte correlación positiva, -1 indica una fuerte correlación negativa y 0 indica que no existe ninguna relación entre los datos.

En la sección 2.2 del presente capítulo se presentarán más medidas de similitud aplicables a las series temporales.

Métodos Jerárquicos de *clustering*:

El *clustering* jerárquico genera una jerarquía o árbol de *clusters*, cuya representación se denomina dendograma [Berkin 2006] y que puede crecer tanto mediante un método de abajo hacia arriba como de arriba hacia abajo. Los métodos de abajo hacia arriba se conocen como aglomerativos, en los cuales el algoritmo considera de inicio que los datos se encuentran divididos

en m *clusters*, donde m es el número de objetos de la muestra; dichos *clusters* se van aglomerando hasta llegar a un número esperado de *clusters* o hasta que se cumpla una condición de finalización. De forma inversa, los métodos de arriba hacia abajo se corresponden con los algoritmos divisivos, los cuales inician con un *cluster* único que se subdivide en *clusters* hijos, los cuales se vuelven a subdividir, y así recursivamente, hasta que se alcanza un número previsto de *clusters* o una condición específica.

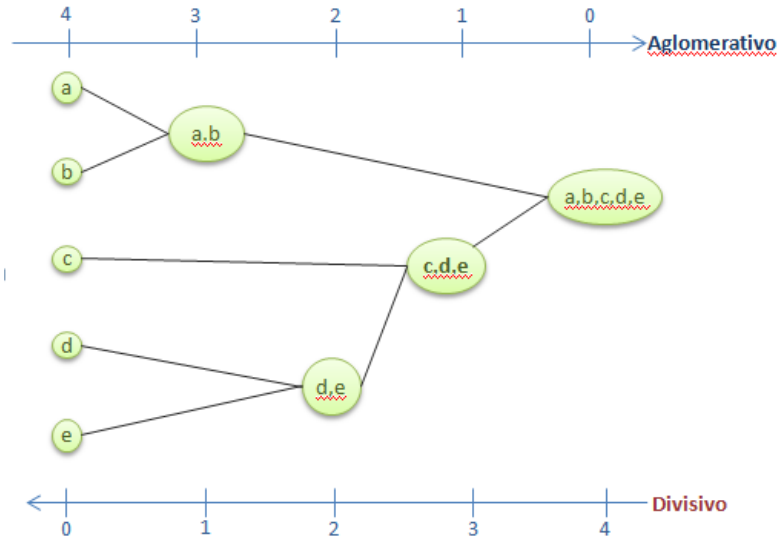

Figura 2.10. *Clustering* Jerárquico Aglomerativo y Divisivo

La figura 2.10 esquematiza el funcionamiento de este tipo de algoritmos, tanto para los métodos aglomerativos como para los divisivos. Estos algoritmos con metodologías en direcciones opuestas, podrían producir resultados muy diferentes [Kaufman 1990].

Según [Rasmussen 1992] Los métodos jerárquicos aglomerativos más comunes son los siguientes:

De enlace simple: Este método junta en cada paso los dos objetos más similares - o menos distantes - , en un sólo *cluster*. Su implementación presenta cierta eficiencia y ha sido muy usado en aplicaciones reales. Los *clusters* que se forman son generalmente dispersos o encadenados, por lo que son válidos para definir *clusters* helicoidales, pero no para el caso de *clusters* esféricos o *clusters* escasamente separados. Es muy sensible al ruido y a los ejemplos atípicos.

De enlace completo: El método de enlace completo, también conocido como de enlace máximo, que mide la proximidad entre dos *clusters* usando la distancia entre sus objetos más lejanos o la similitud entre sus objetos menos semejantes. A este método se le conoce como de enlace completo porque en un *cluster* todos los objetos están enlazados por alguna similitud

mínima. Los *clusters* generados por medio de este método se caracterizan por ser pequeños y compactos, es decir, que sus objetos se encuentran muy cercanos unos a otros. Este método no es muy bueno para encontrar *clusters* convexos o muy grandes pero, por otro lado, es menos susceptible a los objetos atípicos y al ruido.

De enlace promedio de *cluster*: En este método, el promedio de las distancias de pares de objetos tomados de un mismo *cluster* se usa para determinar la similitud entre *clusters*, con lo cual, todos los objetos contribuyen a la similitud inter *cluster*. El resultado es una estructura intermedia entre los dos métodos de enlace simple y de enlace completo de *cluster*, vistos anteriormente.

Método de Ward: El método de Ward es conocido también como el método de la varianza mínima porque en cada nivel minimiza la suma de errores al cuadrado resultante de combinar dos *clusters* en uno, basado en la distancia euclidiana entre sus centroides, lo cual produce *clusters* homogéneos, y una estructura jerárquica simétrica. En este método se define el término centro de gravedad de *cluster* (centroide), lo cual proporciona una manera bastante adecuada para representar un *cluster*. De acuerdo con pruebas realizadas por [Lorr 1983], este método es muy sensible a los objetos atípicos y no se comporta bien con los *clusters* alargados.

Métodos de Reubicación de Particiones:

Clustering Probabilístico:

Los métodos probabilísticos de *clustering* son un intento de evitar el clásico cálculo de la distancia, el cual es reemplazado por un modelo estadístico de medición de distancias entre *clusters*, lo cual es especialmente favorable cuando hay valores nulos dentro del conjunto de datos. En la práctica se parte de la suposición de que los modelos generadores de los datos adoptan una distribución de probabilidad específica, por ejemplo la distribución de Gauss, que es gobernada por parámetros. Por lo tanto, en estos métodos, la tarea de aprender un modelo de función generadora se resume a encontrar los parámetros que mejor se ajustan a los datos del conjunto en cuestión [Berkin 2006].

Método de las k-medias:

Sea k un número entero positivo y m el tamaño del conjunto de datos a ser agrupado. El método de *clustering* k-medias clasifica los m objetos en un número k de *clusters* disjuntos, sobre la base de los valores de sus atributos (k es un parámetro). Para lograrlo, k-medias define prototipo basado en un centroide que es la media de un grupo de puntos y se aplica a un espacio multidimensional continuo [Yadav 2013].

El algoritmo se divide en dos fases: la primera define los k centroides, uno para cada *cluster*, y la segunda fase toma cada punto perteneciente al conjunto de datos y lo asocia al centroide más próximo. Para el cálculo de la distancia entre los datos y los centroides, normalmente se usa la distancia euclidiana. Una vez que todos los puntos han sido asignados a sus respectivos centroides, se ha terminado con el primer agrupamiento y, a partir de ahí, de forma iterativa se recalculan nuevos centroides y se calculan los *clusters* con ellos, y así hasta que los centroides permanezcan sin cambios, lo cual significa que se ha alcanzado el criterio de convergencia del modelo [Aggarwal, Aggarwal 2012]. Este método ha sido muy importante en el ámbito del *data mining*, hasta tal punto que fue considerado como uno de los 10 algoritmos más influyentes de la comunidad investigadora [Wu et al. 2008].

Método de las k-medoides:

El método de las k-medoides es muy similar al método de las k-medias. La diferencia radica en que k-medoides define como prototipo al punto más representativo de un grupo y puede ser aplicado a una gran variedad de datos, puesto que requiere una medida de proximidad para un par de objetos. La diferencia entre un centroide y una medoide radica en que la medoide es un punto de datos existente. Es decir, el centroide se calcula como el centro de un *cluster* y el resultado obtenido puede no coincidir con ninguno de los objetos del conjunto de datos; sin embargo, la medoide es el objeto del conjunto de datos que se encuentra más cerca del centro del *cluster*.

El método de k-medias y el de k-medoides comparten algunas características generales: siempre convergen; diferentes centroides o medoides iniciales generan resultados diferentes y, además, nunca alcanzan el mínimo global [Tibshivani 2013].

Métodos de Partición Basados en Densidad:

Los algoritmos basados en densidad promueven el crecimiento de un *cluster* dado, mientras la densidad de los puntos vecinos exceda un umbral predeterminado [Han et al. 2012]. Estos métodos son apropiados para el tratamiento del ruido que pudiera existir en el conjunto de datos, son muy eficientes a la hora de encontrar *clusters* con formas arbitrarias, necesitan tan solo un escaneo del conjunto de datos pero, al mismo tiempo, requieren que se inicialicen los valores de densidad en calidad de parámetros de entrada [Elavarasi 2011]. El usuario introduce dos parámetros: el primero es ϵ, que representa el radio en el cual se buscarán los puntos cercanos a un punto dado y el segundo es minP, que corresponde a un número mínimo de puntos que deben estar alrededor del punto dado para que sea considerado del área densa.

Métodos Basados en Cuadrícula:

Los métodos basados en cuadrícula transforman el espacio de objetos en una cuadrícula con un número finito de celdas, independientemente de la distribución de los objetos en el espacio. Los *clusters* se forman a través de la unión de cuadrículas. Todas las operaciones se realizan sobre esta cuadrícula. Entre las operaciones que se realizan no está el cálculo de distancias y la complejidad del algoritmo depende del número de celdas ocupadas y no del número de objetos del conjunto de datos, todo lo cual desemboca en un tiempo menor de procesamiento.

Otros tipos de métodos que no han sido mencionados en este breve repaso de los métodos de *clustering* pueden ser encontrados en los estudios realizados por [Babbu et al. 2011], [Becker 2005], [Grira et al. 2005], [Kameshwaran, Malarvizhi 2014] y [Hruschka et al. 2009].

2.2. *Data mining* en series temporales

Este epígrafe es el más importante dentro del capítulo referente al estado del arte, puesto que trata de los conceptos inmersos en el ámbito de las series temporales y las técnicas de *data mining* aplicables a ellas y es, precisamente en ese ámbito, en el que se realiza la contribución de la presente tesis.

Una serie temporal es una secuencia de valores numéricos, o de otro tipo, ordenados cronológicamente y que ocurren, habitualmente, a intervalos uniformes de tiempo. Dar seguimiento al comportamiento de los datos de un fenómeno específico, a través del tiempo, puede producir información importante.

Un ejemplo simple de serie temporal es la que se forma al registrar el precio de un determinado producto, de forma diaria, durante un periodo de tiempo. Dicha serie temporal tiene dos componentes: por un lado el tiempo, que es la variable independiente, cuyo registro es diario, por lo que su valor pertenece al conjunto de los números naturales, mientras que por otro lado está el precio que se constituye en la variable dependiente y cuyo valor pertenece al conjunto de los números reales. Cada valor de precio se corresponde con exactamente un valor de tiempo.

La figura 2.11 ilustra gráficamente una serie temporal, como la del ejemplo mencionado, con 90 puntos registrados.

Día	Precio
1	10.34
2	12.30
3	10.73
4	14.60
5	14.31
6	12.87
⋮	⋮
86	14.36
87	10.78
88	11.80
89	13.91
90	10.92

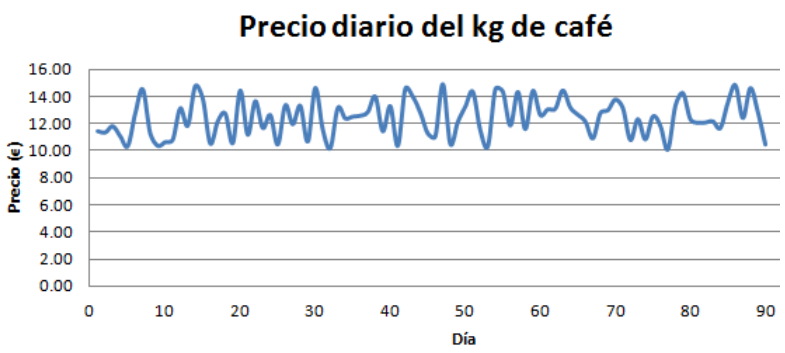

Figura 2.11. Ejemplo de Serie Temporal.

2.2.1. Series temporales numéricas

Las series temporales están presentes en casi todos las áreas del conocimiento y quehacer humano, razón por la cual han atraído la atención de muchos investigadores que han creado métodos y modelos útiles para la extracción de información relevante. Un ejemplo del análisis de series temporales numéricas se encuentra en [Brillinger 2000] y una visión desde la estacionalidad de las series se describe en [Nason 2006].

Resulta imposible enumerarlas todas las áreas en las que son aplicables las técnicas de análisis de series temporales numéricas, sin embargo se presentan aquí algunas a modo de ejemplo:

> Astronomía [Yule 1927], meteorología [Schuster 1898], sismología [Turkey 1965], oceanografía [Groves, Hannan 1968] y [Hassleman et al. 1963], ingeniería en comunicaciones y procesamiento de señales [Rice 1963], control de plantas de procesos continuos [Tee, Wu 1972], neurología y electroencefalografía [Deistler et al. 1986] y [Yuzuriha 1960], y economía [Gudmunsson 1971].

En la gran mayoría de las aplicaciones, lo que se busca es determinar la tendencia, la estacionalidad y la presencia de ciclos, para realizar predicciones a partir de los datos conocidos. Estos cuatro conceptos resultan ser fundamentales en la teoría del análisis de series temporales [Yafee, McGee 2000]:

> **Tendencia**: Se llama tendencia a un cambio sistemático en el nivel de los valores de la serie, a lo largo del tiempo. Las tendencias se clasifican de acuerdo a su tipo y su longitud. Existen los tipos de tendencia determinísticos y los estocásticos. Hay tendencias locales o de corto alcance y globales o de largo alcance.

Estacionalidad: se refiere a los cambios periódicos que definen el comportamiento de la serie temporal, de manera coincidente con las estaciones climáticas. La estacionalidad implica periodos inferiores a un año.

Presencia de ciclos: La presencia de ciclos en una serie temporal sirve para encontrar indicadores que ayuden a la mejor predicción y cálculo de tendencias. Esto ha llevado a que los investigadores busquen ciclos en las series temporales. En el caso del dominio económico, se ha tratado de encontrar ciclos superiores al ciclo usual de un año, para extraer resultados que podrían resultar ser de mucho interés.

Predicción: es el uso de un modelo para predecir valores futuros de una serie temporal, sobre la base de valores previamente existentes.

Al analizar una serie temporal numérica es su estacionalidad, según [Thomson 1994]. Una de las clasificaciones divide a estas series temporales en dos tipos, estacionarias y las series no estacionarias. A continuación se proporciona una breve descripción de estos dos tipos, según [Alonso, García 2012]:

Series temporales estacionarias son aquellas que toman valores estables en el tiempo alrededor de un nivel constante, sin mostrar una tendencia creciente o decreciente a largo plazo. Por ejemplo: precipitación anual en una región, promedios anuales de temperaturas o la proporción de nacimientos correspondientes a varones. La caracterización más evidente de las series temporales estacionarias radica en que poseen una media, una varianza y una estructura de auto-correlación invariables en el tiempo.

Series temporales no estacionarias son aquellas que muestran tendencia, estacionalidad y otros efectos evolucionarios en el tiempo. Por ejemplo: los ingresos anuales de un país, ventas de una empresa o la demanda de energía. Son series que evolucionan en el tiempo con tendencias más o menos estables, en otras palabras, sus medidas de dispersión como la media y la varianza varían en el tiempo.

Los modelos que se utilizan para extraer conocimiento a partir de las series temporales numéricas se basan en técnicas estadísticas, donde entran en escena las distribuciones de probabilidad, las medidas de dispersión, regresiones, etc. y, por otro lado, las transformadas de Fourier, Wavelet, etc. para el análisis en el dominio de la frecuencia [Moler 2004]. Mientras estas técnicas se basan únicamente en el análisis puramente numérico de las series, en la última década se ha buscado optimizar el análisis de las series temporales aplicando técnicas de *data mining* lo cual implica que éstas deben ser representadas de forma apropiada pues, en caso contrario, se vuelve sumamente difícil la extracción de conocimiento a partir de esos datos [Samia 2004] y una

de las maneras más prometedoras de realizar tales representaciones es mediante la transformación de los datos numéricos en simbólicos.

2.2.2. Series temporales agregadas y simbólicas

Las series temporales son usadas en dominios con conjuntos de datos cada vez más numerosos, lo cual dificulta, si no imposibilita, su análisis por medio de técnicas tradicionales, razón por la cual los métodos de *data mining* han sido adoptados para la extracción de información a partir de conjuntos de series temporales. Uno de los enfoques que se usan para que sea posible la aplicación exitosa de tales métodos a las series temporales, incluye un procesamiento previo de éstas, mediante una transformación de las mismas, con lo cual se busca una considerable reducción de su dimensionalidad. La idea es lograr representaciones más abstractas sin que se pierdan las características predictivas de las originales. Los puntos de la serie temporal original se agrupan en intervalos, cada uno de los cuales pasa a ser un objeto cualitativo de alto nivel [Batal et al. 2012]. La secuencia resultante de esta transformación se conoce como una serie temporal agregada y, cuando tales agregaciones se asocian a un alfabeto finito, se denomina serie temporal simbólica o secuencia simbólica. A partir de esas agregaciones es posible la aplicación de técnicas de *data mining* especializadas como indexación, clasificación, agrupamiento, extracción de reglas de asociación y detección de anomalías.

Tres son las condiciones que debe cumplir una transformación [Boucheham 2012]. Considerando que P = $\{p_1,p_2,...,p_N\}$ es una serie temporal de tamaño N y que Q = {q1,q2,...,qK} es una aproximación de tamaño K, las tres condiciones son:

1. K < N, (Regla de la reducción de datos)
2. $q_1 = p_1$ y $q_k = p_N$ (Los extremos coinciden)
3. $\|P, Q\| < \varepsilon$ (La distancia entre la serie original y la aproximada es menor que un umbral)

A continuación se detallan dos técnicas de transformación que producen importantes agregaciones y que destacan por la utilidad que han tenido en la resolución de numerosos problemas y, seguidamente, se describirán algunas de las técnicas más importantes para la transformación a series temporales simbólicas:

2.2.2.1. Técnicas de agregación

Transformada de Fourier Discreta - (Discrete Fourier Transform DFT)

La Transformada de Fourier Discreta ha sido utilizada en gran medida en el análisis de señales y fue introducida en el ámbito del *data mining* en [Agrawal et al. 1993b] para la búsqueda de coincidencias en las consultas a bases de datos de series temporales y aplicada con éxito en otros

casos ([Loh et al. 2000], [Chu, Wong 1999] y [Rafiei 1999]). El argumento que sustenta su eficacia se basa en la idea de que, de acuerdo con el teorema de Parseval (Expresión 2.23) que declara que la energía de una señal es exactamente la misma medida en el dominio de la frecuencia que si se mide en el dominio del tiempo, la distancia euclidiana de una serie temporal medida en el dominio del tiempo será exactamente igual a la distancia de la misma serie medida en el dominio de la frecuencia.

Dicha argumentación no tendría ningún sustento práctico si se considera que la dimensionalidad de una serie temporal en el dominio del tiempo es exactamente igual a su dimensionalidad en el dominio de la frecuencia; pero se echa mano de un supuesto válido que afirma que solamente las frecuencias de los primeros índices representan a la serie temporal con un alto grado de aproximación, con lo cual se logra una gran reducción de la dimensionalidad para la realización de la tarea de *data mining*.

Partiendo de una serie temporal $X = \{x_0, x_1, \dots, x_n\}$, su Transformada de Fourier Discreta viene dada por la expresión 2.24, mientras que la expresión 2.25 proporciona la vía de retorno a la serie original en el dominio del tiempo. $\vec{X} = \{\vec{x}_0, \vec{x}_1, \dots, \vec{x}_n\}$ es la representación de X en el dominio de la frecuencia y $j = \sqrt{-1}$.

$$\sum_{t=0}^{n-1} |x_t|^2 = \sum_{f=0}^{n-1} \left|\vec{x}_f\right|^2$$

Expresión 2.23. Teorema de Parseval

$$\vec{x}_f = {}^{1}\!/_{\sqrt{n}} \sum_{t=0}^{n-1} x_t e^{-j2\pi f t/n} \; f = 0,1, \dots, n$$

Expresión 2.24. DFT de la serie temporal *X*

$$x_t = {}^{1}\!/_{\sqrt{n}} \sum_{f=0}^{n-1} \vec{x}_f e^{j2\pi f t/n} \; t = 0,1, \dots, n$$

Expresión 2.25. Fórmula para la reconstrucción de la serie original

Transformada Wavelet Discreta - (Discrete Wavelet Transform DWT)

Una Transformada Wavelet Discreta (DWT) es una transformación en la cual las muestras de onda (wavelets) son tomadas en periodos discretos de tiempo, con la característica de que mantiene la resolución temporal. En otras palabras, captura tanto la información relativa al tiempo como a la frecuencia, lo cual es una ventaja en el momento del análisis de series temporales.

Varias son las "versiones" de DWTs existentes en la literatura. Para cada problema que reviste una determinada característica especial, se han adaptado nuevos matices, como por ejemplo: Problema de Indexación [Chan, Fu 1999]; Procesamiento Digital de Señales [Akansu, Haddad 2001], [Liang, Lin 2002], [Pandey, Satish 1998], [Mallat 1987] y [Mallat 1989]; Computación Gráfica [Stollnitz et al. 1995], [Gortler 1995] y [Karczmarczuk 1995]; Procesamiento Digital de Señales [Agbinya 1996] y [Singh, Valtorta 1995]. Al ser aplicadas a la transformación de series temporales, DWT tiene tres características favorables:

- Permite obtener una buena aproximación a la serie original, tomando tan sólo los primeros coeficientes;
- El tiempo computacional requerido para su cálculo es proporcional al tamaño de la serie temporal, lo cual significa que su desempeño es bueno;
- Conserva la distancia Euclidiana. Sean X e Y dos series temporales. Sean R y S sus aproximaciones mediante DWT. La distancia Euclidiana D será: $D(X,Y) = \sqrt{2}D(R,S)$.

A continuación, se presenta un caso en el cual se usa la transformación DWT para una aplicación de reconocimiento de voz [Singh et al. 2013].

A partir de una serie temporal X de longitud n, el proceso para obtener su representación, aplicando una técnica de DWT, consta de $\log_2 n$ etapas como máximo. En el primer paso, se generan dos conjuntos de coeficientes a partir de X: coeficientes de aproximación (cA_1) y coeficientes de detalle (cD_1). Estos vectores se generan con la convolución de X a través del filtro de paso bajo para cA_1, y a través del filtro de paso alto para cD_1, seguido por un diezmado diádico (figura 2.12).

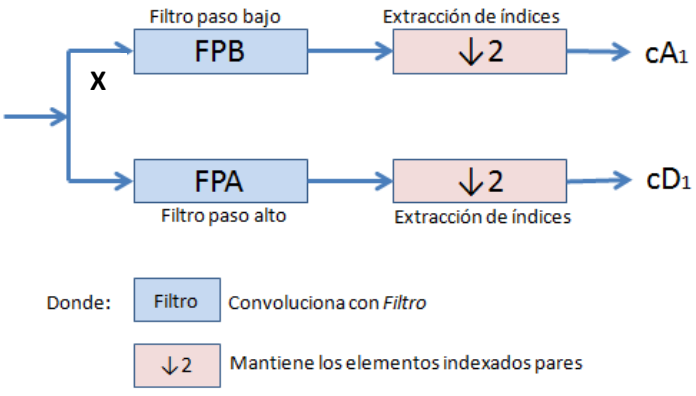

Figura 2.12. Primera fase de descomposición de la DWT

La longitud de cada uno de los filtros es 2N. Si n = longitud(X), la longitud de las series que se obtienen después de los filtros es n + 2N – 1, de donde se puede deducir que los coeficientes cA_1 y cD_1 tienen longitud $\frac{n-1}{2} + N$.

A seguir se producen pasos sucesivos en los cuales se dividen los coeficientes de aproximación y de detalle en dos partes, usando el mismo esquema, reemplazando cA_j y cD_j para generar cA_{j+1} y cD_{j+1}, y así sucesivamente hasta que el número de coeficientes sea 2 (figura 2.13).

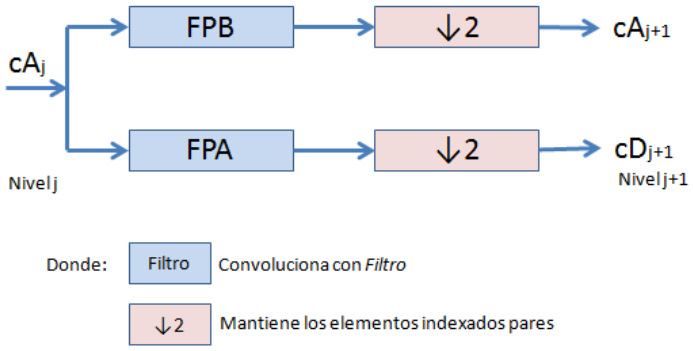

Figura 2.13. Paso genérico de descomposición de la DWT

La descomposición de X, analizada a nivel j, tendrá la siguiente estructura: [cA_j, cD_j, ... cA_1, cD_1]; por otro lado, IDWT (DWT Inversa) reconstruye cA_{j-1} y cD_{j-1} a partir de cA_j y cD_j, colocando ceros en lugar de los coeficiente eliminados y convolucionando los resultados con filtros de reconstrucción (figura 2.14).

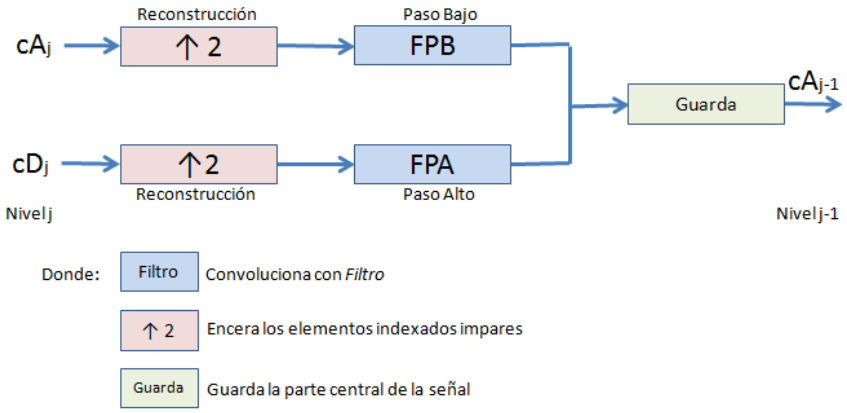

Figura 2.14. Reconstrucción de la serie original.

2.2.2.2. Técnicas de simbolización

El principal problema con el que hay que enfrentarse a la hora de analizar series temporales es el de la alta dimensionalidad de estas. Muchos son los autores que han propuesto métodos para reducir dicha dimensionalidad, a fin de lograr una disminución del tamaño de las series temporales

y por ende, la consecución de mejoras de rendimiento y ahorro de almacenamiento. Algunas de las más importantes propuestas se describen brevemente a continuación.

Descomposición de Valor Sencillo - (Singular Value Decomposition SVD).

La Descomposición en Valores Sencillos SVD, es una técnica de transformación que ha sido usada para la indexación en bases de datos de secuencias temporales [Keogh et al. 2000a], [Korn et al. 1997] y también se ha empleado en la indexación de imágenes y otros objetos [Wu et al. 1996], [Kanth et al. 1998].

A diferencia de otras técnicas, SVD es una transformación que usa el enfoque global, lo cual significa que interviene en todo el conjunto de datos y lo rota de manera tal que sus ejes, ortogonales entre sí, produzcan las varianzas máximas posibles.

Como ventaja de esta técnica, para la ejecución de consultas, se puede señalar que es muy eficiente y eficaz para recuperar los resultados; por otro lado, su debilidad radica en la cantidad de recursos que consume al construir el índice, que tiene que ser reconstruido cada vez que se modifica la base de datos.

Piecewise Aggregate Approximation PAA.

Esta técnica de transformación de series temporales fue ideada de forma casi simultánea por dos equipos de investigadores que trabajaron de forma independiente [Keogh et al. 2000a] y [Yi, Faloutsos 2000]. El primero la llamó Piecewise Aggregate Approximation mientras que el segundo la denominó Segmented-Means. En la literatura se hace mayor referencia a PAA.

Para obtener una representación aproximada de una serie, PAA divide la serie en un conjunto finito de segmentos de igual tamaño y almacena las medias de los valores correspondientes a los puntos que corresponden a cada segmento.

Esta técnica ha tenido una gran aceptación dentro de la comunidad científica puesto que, a pesar de su sencillez, logra muy buenos resultados en términos de exactitud y de eficiencia. La figura 2.15 muestra un ejemplo del uso de esta técnica para aproximar una serie temporal; en la serie original, los puntos están muestreados a intervalos de 0.05 segundos, mientras que los intervalos del PAA tienen una duración de 0.25 segundos. Se puede apreciar la importante reducción de dimensionalidad que existe para este caso específico, en el cual una serie de 160 puntos se transforma en una secuencia de 32 intervalos.

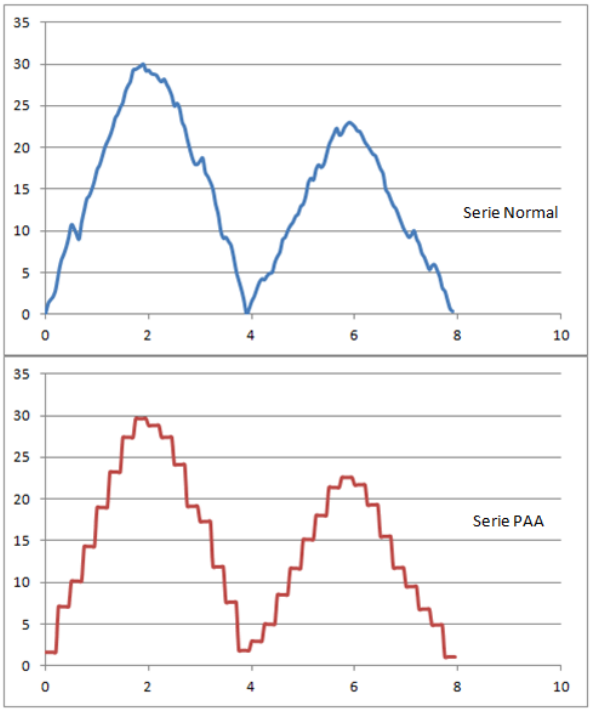

Figura 2.15. Ejemplo de transformación mediante PAA

Adaptive Piecewise Constant Approximation APCA

Esta técnica fue presentada por primera vez en [Chakrabarti et al. 2001] y su fundamento es esencialmente el mismo que el de PAA, con la única diferencia de que los intervalos de agregación pueden tener longitudes variables, al contrario de lo que sucede con PAA.

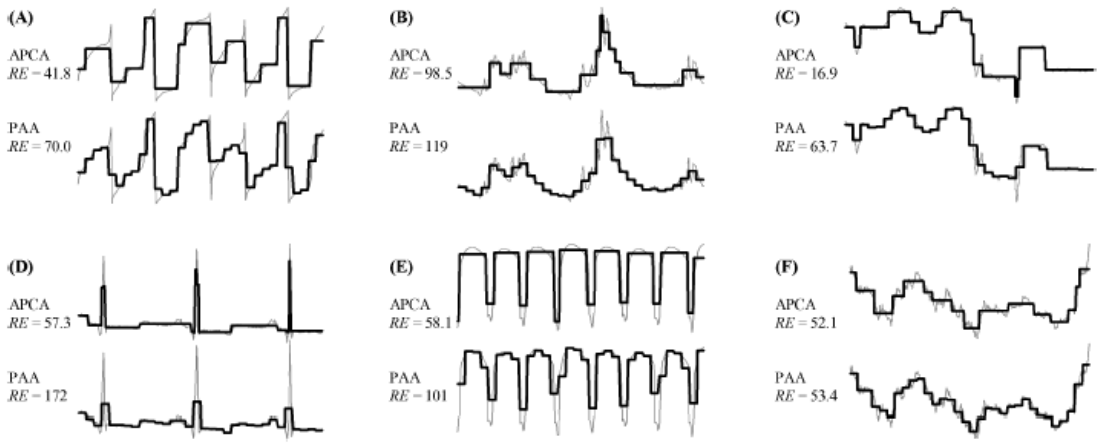

Figura 2.16. Comparación entre PAA y APCA [Chakrabarti et al. 2001]

De acuerdo con sus autores, con ese solo cambio se logra obtener una importante mejora en la dimensionalidad de la serie transformada, con respecto a PAA. La figura 2.16 - tomada de [Chakrabarti et al. 2001] - muestra una comparación entre las dos técnicas donde además se

declara que los resultados son mejores con APCA que con PAA en términos del error global obtenido al comparar las series transformadas con las originales

Aproximación Agregada Simbólica (Symbolic Aggregate approXimation SAX)

Este método para la transformación de series temporales fue introducido formalmente en [Lin et al. 2003], a pesar de que en trabajos anteriores ya se mencionaba como procedimiento para la aproximación de series temporales, por ejemplo en [Lin et al. 2002]. Ha sido mencionado numerosas veces en la literatura, y aparece descrito en diferentes trabajos con gran nivel de detalle ([Lin et al. 2007], [Lin, Li 2009], [Ordóñez et al. 2008], [Ordóñez, Jardins 2011], [Pham et al. 2010], [Senin, Malinchik 2013], [Gamero 2012] y [Cassissi et al. 2009]). Por otro lado, innumerables variantes del método han sido ideadas y usadas en diferentes estudios; algunos ejemplos pueden encontrarse en [Lkhagva et al. 2006], [Pham et al. 2010], [Malinowski et al. 2013] y [Shied 2008].

El objetivo de SAX es transformar una serie temporal en una secuencia de símbolos que forman una única palabra con una longitud predeterminada y puede ser descrito como un procedimiento de dos pasos.

En primer lugar, se aplica un método general de aproximación a la serie temporal, esto es, transformarla a una secuencia de segmentos mediante Aproximación Agregada por Partes (PAA) para luego transformarla a símbolos discretos con puntos de ruptura determinados y una cantidad finita de símbolos.

La figura 2.17 muestra la serie temporal original y su aproximación mediante PAA. Al transformar la serie a una secuencia de intervalos, se cumple con la primera fase, mencionada en el párrafo anterior. La segunda fase se ilustra en la figura 2.18.

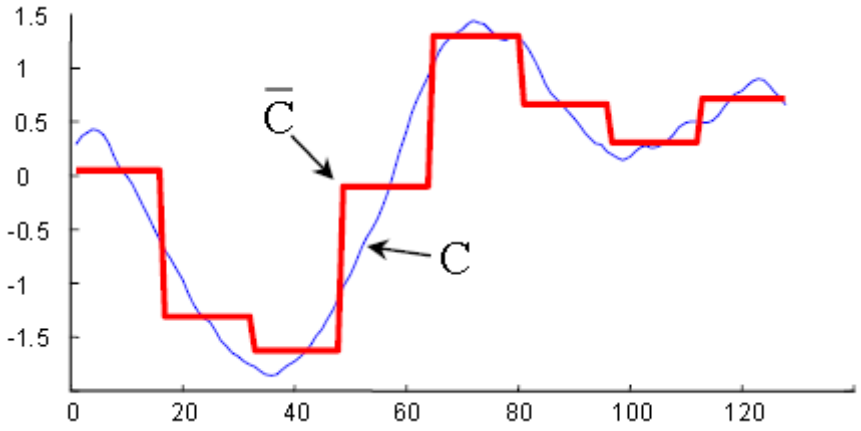

Figura 2.17. Transformación PAA: C (original) y $\bar{C}$(Aproximada).

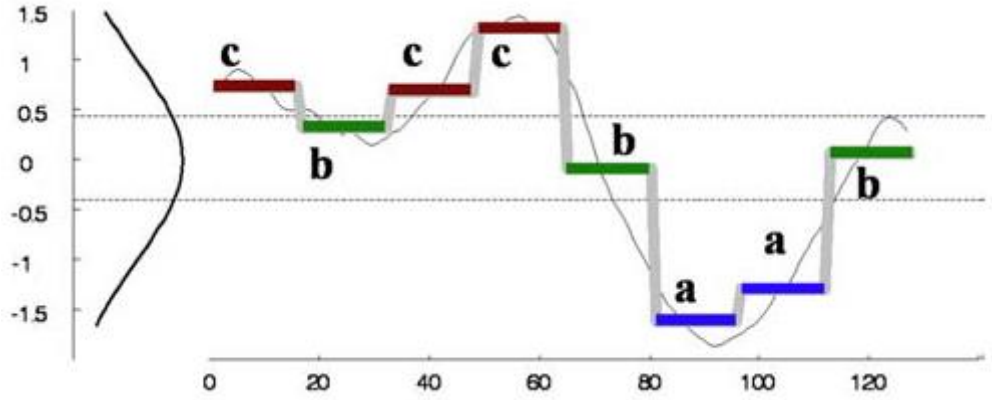

Figura 2.18. Conversión de la serie temporal a símbolos usando SAX.

La idea principal de SAX está en dividir el espacio en regiones con igual probabilidad, conforme a la distribución de Gauss, lo cual se logra mediante la aplicación de una función que recibe dos parámetros: w, que corresponde al número de dimensiones de la serie transformada, denominada longitud de la palabra (*word*); y a, que es el número de regiones, lo cual implica que se requieren a - 1 puntos de corte. La figura 2.18 constituye un ejemplo concreto, cuyos puntos de corte se encuentran en la matriz de búsqueda que se muestra en la figura 2.19. Con estos valores, para este ejemplo, se obtiene una palabra con 8 símbolos, cada uno de los cuales depende de la región en la que se encuentre el segmento al que representa.

β_i \ a	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

Figura 2.19. Matriz de Búsqueda de los puntos de corte para diferentes valores de *w* y *a*.

Al final del procedimiento, para el caso del ejemplo de la figura 2.18 se obtiene la transformación de la serie temporal original de 128 puntos a una palabra de ocho símbolos: ccbccaab.

Los métodos de transformación de series temporales descritos hasta aquí, tienen como objetivo principal reducir la dimensionalidad de las series, sin tomar en cuenta la morfología de las mismas, esto es, no las transforman manteniendo íntegras las formas básicas que conforman la serie temporal. Pero en ciertos dominios la integridad de tales formas resulta ser importante, razón

por la cual ciertos autores han propuesto métodos de transformación que intentan mantener la morfología de la serieoriginal en la serie transformada. A continuación se presentan algunos métodos cuyo objetivo es mantener las formas básicas de la serie, pues ellas podrían contener información potencialmente valiosa que se podría perder al aplicar los métodos anteriormente descritos.

[Batal et al. **2009]**

El algoritmo STF-Mine es propuesto en [Batal et al. 2009], para extraer patrones abstractos de un conjunto de series temporales y a partir de ellos aprender un modelo de clasificación. El punto que interesa, por ahora, es la segmentación de las series temporales para obtener una descripción cualitativa de las mismas.

Los símbolos abstractos constan de dos partes: Abstracciones de valor (Alto, Normal y Bajo) y Abstracciones de tendencia (Creciente, Decreciente y Estable). El alfabeto de símbolos que se obtendrá corresponderá a la combinación de las dos abstracciones; por ejemplo, un tramo de la serie podría ser considerado un símbolo que sea creciente, en cuanto a tendencia. y que sea normal en lo que a su valor se refiere. Esta combinación es un buen avance hacia la reducción de dimensionalidad tomando en cuenta la forma de las series temporales,

[Kumar, Kalia 2011]

El objetivo del trabajo realizado en [Kumar, Kalia 2011] consiste en establecer diferencias entre los enfoques numérico y simbólico en la búsqueda de patrones en series temporales, llegando a la conclusión de que el enfoque numérico proporciona resultados más precisos pero, en compensación, el enfoque simbólico es más fácil de interpretar y ayuda en la búsqueda de un patrón de conjunto.

Tres símbolos son utilizados para representar la serie transformada: Up, Down y Neutral.

[Santamaría 2011]

En este trabajo se propone un marco de descubrimiento de conocimiento en series temporales numéricas aplicando métodos simbólicos. El punto más importante que se trata en esta publicación es la transformación de series temporales a secuencias de caracteres, donde cada carácter representa un símbolo que incorpora parte de la semántica de la serie y que tiene significado para el experto en el dominio.

Los símbolos que se proponen son *Subida, Bajada, Pico, Hundimiento, Transición* y *Curvatura*. Cada uno de los símbolos puede tener categorías, como por ejemplo, en el caso de un Pico: *Grande* y *Pequeño*. Para la categorización de los símbolos se utilizan, como elementos discriminatorios, tanto la duración como la amplitud del mismo.

Una de las principales motivaciones para esta propuesta es la gran ventaja que este tipo de transformación tiene a la hora de interactuar con los expertos en el dominio para la incorporación de su conocimiento al marco de trabajo. Este trabajo fue realizado en el ámbito del área médica de la isocinesia.

[Batal et al. **2012]**

Un método diferente de transformación de series temporales es el presentado en [Batal et al. 2012], que usa un *marco para minería de patrones recientes*. Este marco convierte la serie temporal en secuencias de abstracciones temporales basadas en intervalos de tiempo y luego construye patrones temporales más complejos mediante el uso de operadores temporales.

Cada serie temporal es transformada en una representación $(v_1[s_1, e_1], ..., v_n[s_n, e_n])$, donde $v_i \in \Sigma$ es una abstracción que abarca desde el instante s_i hasta el instante e_i y Σ es el alfabeto de abstracción que representa un conjunto finito de abstracciones permitidas.

Los estados abstractos en los cuales son categorizados los datos de las series temporales son: Muy Bajo *(VL)*, Bajo *(L)*, Alto *(H)*, Muy Alto *(VH)*, lo cual significa que el alfabeto en este caso es $\Sigma = \{VL, L, N, H, VH\}$.

2.2.3. Medidas de similitud en series temporales simbólicas

Las medidas de similitud son necesarias, en el *data mining* de series temporales, a la hora de establecer el nivel de coincidencia entre ellas o entre partes de ellas. La aplicación más obvia se da en la tarea de Búsqueda y Recuperación, en la cual se busca un patrón Q dentro de un conjunto de series temporales $S = \{s_1, s_2, ..., s_n\}$ y el resultado obtenido es un subconjunto de S formado por las series que contengan una secuencia igual o similar a Q. Todos los métodos de búsqueda y recuperación de series temporales adoptan alguna medida de similitud compatible con el tipo de transformación que se realice a las series temporales, lo cual significa que las series con transformaciones simbólicas, cuyos resultados son secuencias de símbolos, usarán medidas de similitud simbólicas.

A continuación se presentan dos de las medidas de similitud numérica más utilizadas según la literatura y, posteriormente, se presentan las medidas de similitud simbólicas.

Distancia Euclidiana

Esta medida de similitud es la más utilizada dentro del ámbito del *data mining* pues, es una medida de distancia métrica y por lo tanto, garantiza el cumplimiento de las cuatro propiedades mostradas en la expresión 2.26.

$$\begin{aligned} &f(X,Y) \geq 0 && (no\ negatividad)\\ &f(X,Y) = 0\ \ iff\ \ X = Y && (Identidad)\\ &f(X,Z) \leq f(X,Y) + f(Y,Z) && (Desigualdad\ Triangular)\\ &f(X,Y) = f(Y,X) && (Simetría) \end{aligned}$$

Expresión 2.26. Propiedades de las distancias métricas

Muchos artículos en la literatura científica la avalan como una forma muy eficiente para calcular la distancia entre dos series temporales de igual tamaño, como [Chan, Fu 1999], [Cassissi et al. 2012], [Valente, López 2000], etc.

La distancia Euclidiana se calcula extrayendo la raíz cuadrada de la suma de los cuadrados de las distancias punto a punto entre las series temporales, lo cual se ilustra en la figura 2.20.

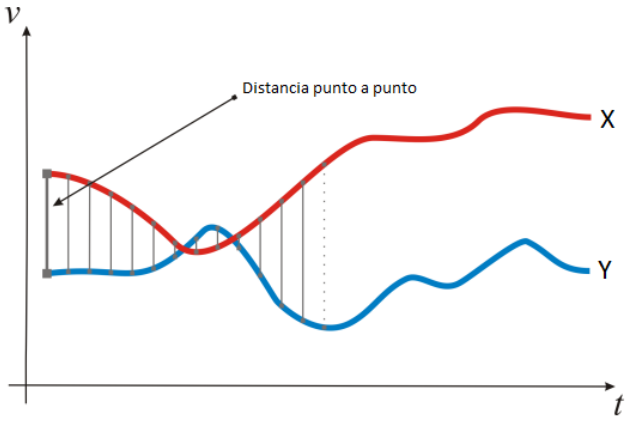

Figura 2.20. Ilustración de la distancia euclidiana entre dos series temporales X e Y

Sean X = {x_1,x_2,...,x_n}, Y = {y_1,y_2,...,yn} dos series temporales de tamaño n. La Distancia Euclidiana entre X e Y se calcula conforme a la expresión 2.27.

$$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Expresión 2.27. Distancia euclidiana

Entre las limitaciones de esta medida de similitud se consideran las siguientes:

1. Las series en cuestión tienen que ser del mismo tamaño.
2. No soporta desplazamiento en el tiempo.
3. No soporta escalamiento en amplitud.

Por otro lado, tanto el escalamiento en amplitud como el desplazamiento en el tiempo pueden ser solucionados con la llamada normalización de las series temporales.

Distorsión Temporal Dinámica (Dynamic Time Warping)

La medida de similitud DTW [Sakoe 1971] es preferida a la distancia euclidiana en muchas aplicaciones pues es capaz de soportar tanto el escalamiento en amplitud como el desplazamiento en el tiempo, y es también una medida de distancia métrica. Al igual que la distancia euclidiana ha sido ampliamente mencionada en la literatura científica como una opción válida para establecer la distancia entre dos series temporales con diferentes tamaños, [Park, Chu 2001], [Cassissi et al. 2012], [Gunopulos, Das 2001], [Papetrou et al. 2011], [Toyoda et al. 2013], [Keogh et al. 2000b], [Keogh, Ratanamahatana 2002], [Keogh, Ratanamahatana 2004], [Chen et al. 2005a], [Chen et al. 2005b].

En la figura 2.21 se puede observar la diferencia entre la distancia euclidiana y DTW. Lo más notorio es la relación entre los puntos de las dos series: en la primera existe una relación 1:1, mientras que en el DTW la relación es 1:n en ambos sentidos, razón por la cual no se requiere que ambas series sean del mismo tamaño.

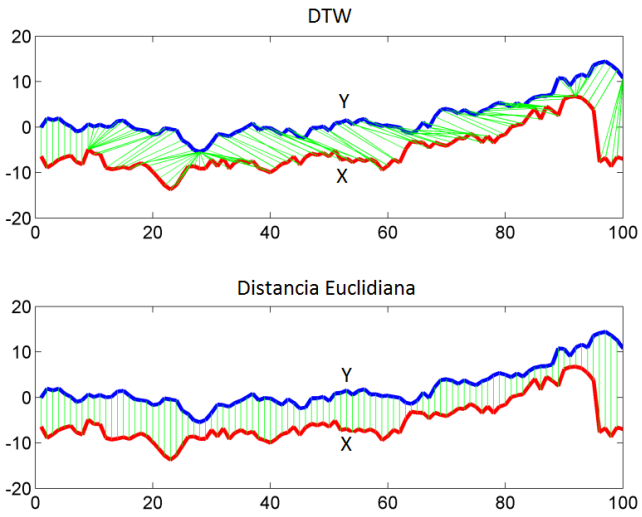

Figura 2.21. DTW vs. Distancia Euclidiana

Muchas veces resulta ser un problema el hecho de que muchos puntos de una serie temporal se correspondan con un solo punto de la otra, generándose un serio desequilibrio; esto se resuelve

restringiendo el camino de deformación, aplicando la parada de la recursión a una cierta profundidad mediante el establecimiento de un umbral δ para la diferencia entre los índices i y j.

$$\gamma(i,j) = \begin{cases} d(X_i, Y_j) + \min\{\gamma(\mathrm{i}-1,\mathrm{j}-1), \gamma(\mathrm{i}-1,\mathrm{j}), \gamma(\mathrm{i},\mathrm{j}-1)\} & |\mathrm{i}-\mathrm{j}| < \delta \\ \infty \quad \text{en otro caso} \end{cases}$$

Expresión 2.28. Cálculo de los elementos de la Matriz de Distorsión

Para el cálculo de la distancia DTW es necesario contar primero con la matriz de distancias acumuladas, que se obtiene a partir de la expresión 2.28 y que se encuentra ejemplificada gráficamente en la figura 2.22.

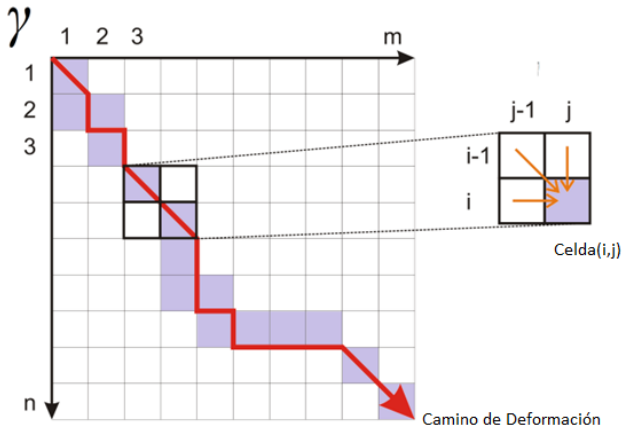

Figura 2.22. Matriz de distancias acumuladas.

Según [Gunópulos 2011], el camino de deformación debe obedecer a ciertas restricciones:

1. *Monotonicidad*: el camino no puede avanzar hacia abajo o hacia la izquierda.
2. *Continuidad:* ningún elemento de las series puede ser saltado.
3. *Umbral* o *ventana de distorsión*: $|i - j| <= \delta$.

Distancias de Edición

Las distancias de edición son medidas de similitud entre secuencias de caracteres, ideadas por [Damerau 1964] y [Levenshtein 1966] y se implementan como funciones que reciben como parámetro un par de secuencias de caracteres y devuelven un valor numérico. El valor devuelto por la función refleja las operaciones que se realizan para transformar una secuencia de caracteres s_1 en otra s_2.

Son medidas de distancia en el sentido que, dadas tres secuencias de caracteres s_1, s_2 y s_3, se cumplen las siguientes condiciones:

1. No-negatividad: $d(s_1, s_2) >= 0$.

2. Cero si y solo si igual: $d(s_1, s_2) = 0$ *si y solo si* $s_1 = s_2$.

3. Simetría: $d(s_1, s_2) = d(s_2, s_1)$.

4. Desigualdad triangular: $d(s_1, s_2) + d(s_2, s_3) >= d(s_1, s_3)$.

En *data mining*, las distancias de edición han sido muy usadas para calcular la distancia entre series temporales simbólicas que, generalmente, son convertidas a secuencias de caracteres, de acuerdo con un alfabeto predefinido.

Distancia de Levenshtein

Es la más simple de las distancias de edición (ED), publicada por vez primera en [Levenshtein 1966]. Cuenta con tres operaciones que son inserción, borrado y substitución de dos caracteres adyacentes. El valor de la distancia de Levenshtein es igual al número mínimo de ediciones que hay que realizar para transformar una secuencia en otra.

La expresión 2.29 muestra las fórmulas utilizadas para el cálculo de la distancia de Lievenstein.

$$d_{ij} = \begin{cases} d_{i-1,j-1} & a_j = b_i \\ min \begin{cases} d_{i-1,j} + 1 \\ d_{i,j-1} + 1 \\ d_{i-1,j-1} + 1 \end{cases} & a_j \neq b_i\,, \qquad for\ 1 \leq i \leq m, 1 \leq j \leq n \end{cases}$$

Expresión 2.29. Cálculo recursivo de la distancia de Levenshtein

Ejemplo: s_1 = hola y s_2 = cola. La distancia de edición es 1 puesto que la única operación que se debe realizar para transformar s_1 en s_2 es la substitución de la letra **h** por la letra **c**.

Distancia de Edición Ponderada

A diferencia de la distancia de edición Damerau-Levenshtein en la cual todas las operaciones tienen coste 1, la distancia de edición ponderada (WED) considera costes diferentes para las distintas operaciones. El hecho de ponderar las operaciones responde a necesidades del dominio [Kurtz 1996]; por citar un ejemplo, si se trata de un corrector ortográfico, unos pares de letras son más probables de ser intercambiados que otros, debido a la disposición de las letras en el teclado del ordenador.

$$d_{i0} = \sum_{k=1}^{i} w_{del}(b_k), \quad for\ 1 \leq i \leq m$$

$$d_{0j} = \sum_{k=1}^{j} w_{ins}(a_k), \quad for\ 1 \leq j \leq n$$

$$d_{ij} = \begin{cases} d_{i-1,j-1} & a_j = b_i \\ min \begin{cases} d_{i-1,j} + w_{del}(b_i) \\ d_{i,j-1} + w_{ins}(a_j) \\ d_{i-1,j-1} + w_{sub}(a_j, b_i) \end{cases} & a_j \neq b_i \end{cases}, \quad for\ 1 \leq i \leq m, 1 \leq j \leq n$$

Expresión 2.30. Cálculo recursivo de la distancia WED

Distancia de Hamming

La distancia de Hamming, introducida en [Hamming 1950], permite únicamente substituciones, con coste 1, se aplica únicamente a un par de secuencias de caracteres de igual tamaño y el valor de la distancia es el número mínimo de substituciones que se tienen que realizar para convertir una secuencia en la otra. La restricción referida al tamaño de las series temporales hace que la utilidad de la distancia de Hamming sea muy limitada.

Distancia de Edición Extendida

La Distancia de Edición Extendida (EED) [Fuad, Marteau 2008a] [Fuad, Marteau 2008b], fue concebida para calcular el grado de similitud entre cadenas de caracteres, argumentando que la Distancia de Edición (ED) no toma en cuenta la cantidad de caracteres diferentes (NC) contenidos en las cadenas a ser comparadas, ni la frecuencia de los mismos. La expresión 2.30 muestra la función para el cálculo de EED, donde: X e Y son dos cadenas de caracteres; fi(X) es la frecuencia del carácter i en X y fi(Y) es la frecuencia del carácter i en Y; |X| y |Y| son las longitudes de X e Y respectivamente; λ es el factor de frecuencia, λ >= 0 (λ ∈ R); i es el número del elemento dentro de NC.

$$EED(X,Y) = ED(X,Y) + \lambda \left[|X| + |Y| - 2 \sum_i \min(f_i^{(X)}, f_i^{(Y)}) \right]$$

Expresión 2.30. Fórmula para el cálculo de la distancia *EED*

Cuando *λ = 0, EED(X,Y) = ED(X,Y)*, lo cual significa que *ED(X,Y)* es el límite inferior de la distancia *EED.*

Subsecuencia Común más Larga (LCS)

Este método para calcular la distancia fue introducido en [Needleman, Wunsch 1970] y un estudio más reciente [Apostolico, Guerra 1985] añade mejoras de rendimiento al algoritmo. El problema consiste en encontrar la subsecuencia, más larga, común a dos secuencias dadas.

La distancia de Subsecuencia Común más Larga, admite solamente eliminaciones e inserciones, todas con coste 1. El nombre de esta distancia se debe a que mide la longitud del emparejamiento

más largo de caracteres respetando el orden y, consecuentemente, la distancia será el número de los caracteres que no emparejaron. La distancia es simétrica y se cumple la expresión 2.31:

$$0 \leq d(XY) < |y| - |x|$$

Expresión 2.31. Límites de la Distancia de Subsecuencia común más larga

La Expresión 2.32 corresponde a la función recursiva que permite calcular la distancia LCS para dos secuencias de caracteres X e Y. En la tercera línea se puede apreciar que se calculan dos distancias entre los caracteres más cercanos y se selecciona la mayor de ellas, para continuar con el proceso recursivo.

$$LCS(x_i, y_j) = \begin{cases} 0 & if\ i = 0\ o\ j = 0 \\ LCS(x_{i-1}, y_{j-1}) + 1 & if\ x_i = y_j \\ másLarga\left(LCS(x_i, y_{j-1}), LCS(x_{i-1}, y_j)\right) & if\ x_i \neq y_j \end{cases}$$

Expresión 2.32. Distancia de Subsecuencia común más larga

Distancia de Episodios

Dadas dos secuencias de caracteres, un texto T de longitud n y un episodio P[1] de longitud m, el problema de coincidencia de episodios consiste en encontrar todas las subsecuencias de T, con longitud mínima, que contengan P como una subsecuencia [Das et al. 1997]. El problema de optimización subyacente consiste en encontrar un número mínimo w tal que el texto T contenga una subsecuencia de caracteres de longitud w que a su vez contenga el episodio P.

La distancia de Episodios solo admite inserciones con coste 1. En la literatura, el problema de búsqueda que usa esta distancia es denominado coincidencia de episodios. Al no cumplir con la característica de simetría, por lo cual podría ser imposible convertir una secuencia de caracteres X en otra Y, en tal caso, el coste d(x, y) puede ser |y| - |x| o ∞.

Distancia de Edición con Penalización Real

En [Chen et al. 2005a] se proponen dos medidas de similitud entre secuencias temporales: la Distancia de Edición con Penalización Real o ERP y la Distancia de Edición en Secuencias Reales o EDR. Ambas soportan desplazamiento en el tiempo.

La distancia ERP entre dos secuencias X e Y, con longitudes m y n respectivamente, se basa en el número de operaciones de edición que se han de realizar sobre una serie para transformarla en la otra. En la propuesta realizada por Chen, se define esta distancia sobre la base de la distancia

[1] Un episodio es un conjunto de eventos totalmente ordenados, y la frecuencia de un episodio es la medida de qué tan frecuentemente éste ocurre dentro de una secuencia [Ao et al. 2015].

de edición simple, ED, y la distancia DTW. Son tres las operaciones admitidas, todas ellas con coste 1. Una penalización real se aplica cuando x_i e y_i no son gaps y una penalización constante cuando uno de los dos es un gap, entendiéndose como gap un símbolo añadido a una secuencia. La operación de borrado puede ser tratada como una inserción de un símbolo en la otra secuencia.

La expresión 2.33 muestra la función recursiva para el cálculo de la Distancia ERP donde g es un gap, al que se le asigna un valor constante.

$$ERP(X,Y) = \begin{cases} \sum_1^m |x_i - g| & if\ n = 0 \\ \sum_1^n |y_i - g| & if\ m = 0 \\ \min\{ERP\big(Rest(X), Rest(Y)\big) + dis_{erp}(x_1, y_1), & Caso \\ ERP(Rest(X), Y) + dist_{erp}(x_1, g), & contrario \\ ERP(X, Rest(Y)) + dist_{erp}(y_1, g)\} \end{cases}$$

Donde,

$$dist_{erp}(x_i, y_i) = \begin{cases} |x_i - y_i| & si\ x_i, y_i\ no\ son\ gaps \\ |x_i - g| & si\ y_i\ es\ gap \\ |y_i - g| & si\ x_i\ es\ gap \end{cases}$$

Expresión 2.33. Distancia de Edición con Penalización Real

Distancia de Edición en Secuencias Reales

Esta medida de similitud, propuesta en [Chen et al. 2005a], al igual que la distancia ERP también soporta desplazamiento en el tiempo, con la diferencia de que soporta además la presencia de ruido en los datos.

Al igual que en la distancia ED, EDR se calcula de acuerdo con el menor número de inserciones, substituciones y borrados que se requieren para convertir una secuencia en otra.

La expresión 2.34 representa la fórmula recursiva para el cálculo de la Distancia EDR, donde subcost = 0 si match(x_i,y_i) es verdadero, caso contrario subcost = 1.

$$EDR(S,Y) = \begin{cases} m & si\ n = 0 \\ n & si\ m = 0 \\ \min\{EDR\big(Rest(X), Rest(Y)\big) + subcost, & \\ EDR(Rest(X), Y) + 1, EDR(X, Rest(Y)) + 1\} & Caso\ Contrario \end{cases}$$

Expresión 2.34. Distancia de Edición de Secuencias Reales

2.2.4. Descubrimiento de patrones en series temporales simbólicas

Las tareas de *data mining* en series temporales pueden ser agrupadas de la siguiente manera [Laxman, Sastry 2006]: predicción, clasificación, agrupamiento, búsqueda y recuperación, y

descubrimiento de patrones. De esta clasificación podemos deducir que el descubrimiento de patrones en sí ya es una tarea de *data mining* a pesar de que, por lo general, se lo requiere de forma previa a la realización de otra.

Un patrón frecuente es una sub-secuencia que aparece de forma repetitiva en un conjunto de datos, con una frecuencia igual o superior a un umbral predeterminado [Han et al. 2007].

La búsqueda de patrones frecuentes es un paso fundamental para la realización de otras tareas de *data mining* como clasificación, agrupamiento y predicción. Además se la realiza con mucha frecuencia en búsqueda y recuperación. Por otro lado, la búsqueda de patrones frecuentes se ha convertido en una tarea de *data mining* extremadamente importante, razón por la cual se ha puesto mucho interés en ella como tema de investigación, en los últimos años. A continuación se describen algunos métodos de descubrimiento de patrones que destacan en la literatura.

2.2.4.1. Apriori

Una propiedad importante de los conjuntos de ítems fue observada por Agrawal y Srikant, a la cual denominaron Apriori: Un conjunto de ítems de longitud k es frecuente solamente si todos sus subconjuntos de ítems son frecuentes [Agrawal, Srikant 1994]. Si usamos esta propiedad en el sentido contrario, para la búsqueda de patrones frecuentes en series temporales, se puede afirmar que los patrones frecuentes de longitud 1 pueden ser usados para encontrar los patrones frecuentes de longitud 2 y éstos a su vez para encontrar los de longitud 3 y así sucesivamente hasta que no exista más posibilidad de obtener patrones más largos.

2.2.4.2. FP-Growth

El algoritmo de FP-Growth [Han et al. 2000] (crecimiento de patrones frecuentes) fue concebido por una necesidad de reducir significativamente la costosa tarea de encontrar y contrastar los patrones candidatos que se generan en los algoritmos basados en Apriori. Esta idea se basa en la creación de una estructura en árbol, denominado FP-tree, que almacena los patrones frecuentes partiendo de los patrones frecuentes de longitud 1 que son encontrados en el primer escaneo del conjunto de datos.

2.2.4.3. Eclat

El algoritmo para descubrimiento de patrones frecuentes Eclat [Zaki 2000] (Equivalence Class Transformation), se basa en la representación de las transacciones en formato vertical. Apriori y FP-Growth se basan en una representación horizontal de los datos, esto es, (Tid, {Items}) donde Tid es el identificador de la transacción y {Items} es el conjunto de ítems asociado a la misma.

Sin embargo, Eclat usa la representación vertical que consiste en reagrupar los datos del conjunto de datos en el formato (Item, {Tids}), durante el primer escaneo. Después de realizar esta transformación, los patrones de nivel 1 habrían sido encontrados y entonces se aplican intersecciones entre los {Tids} para la determinación de los patrones de nivel más alto, usando la propiedad Apriori.

2.2.4.4. Conjuntos de datos cerrados y máximos

Uno de los problemas que se generan al realizar el descubrimiento de patrones, con los métodos mencionados hasta aquí, es la gran cantidad de patrones que satisfacen el umbral, puesto que para cada patrón frecuente todos sus sub-patrones son también frecuentes. Para solucionar este problema, se propone el método de minería del patrón frecuente máximo y del patrón frecuente cerrado.

Un patrón frecuente cerrado p [Pasquier et al. 1999] en un conjunto de datos D es aquel que no tiene un súper patrón q tal que q tenga el mismo soporte de p en D. Pasquier propone un algoritmo, basado en Apriori, denominado A-Close.

Un patrón frecuente máximo p en un conjunto de datos D es aquel que no tiene un súper patrón q tal que $p \subset q$ y q sea frecuente en D. El primer estudio de estos patrones fue realizado en [Bayardo 1998], que propone un algoritmo denominado MaxMiner que consiste en un método basado en Apriori, por niveles, que usa el método de búsqueda primero en amplitud para encontrar el conjunto de ítems máximo, mediante la poda de subconjuntos y la poda de súper conjuntos.

2.2.4.5. Patrones secuenciales

El descubrimiento de patrones secuenciales fue introducido por primera vez en [Agrawal et al. 1995], donde se proponen tres algoritmos que resuelven el problema de la búsqueda de patrones secuenciales en grandes bases de datos transaccionales. Un patrón secuencial es un conjunto de eventos ordenados que ocurren frecuentemente, es decir, una sub-secuencia frecuente.

Sea I = {i_1, i_2, ... , i_k} el conjunto de todos los ítems. Cualquier subconjunto de I es llamado un conjunto de ítems. Una secuencia t = <t_1, t_2, ... , t_m> $t_i \subset I$ es una lista ordenada. Cada conjunto de ítems en una secuencia representa un conjunto de eventos que ocurren en el mismo momento, mientras que diferentes conjuntos de ítems ocurren en diferentes momentos. Se asume que los ítems en cada conjunto de ítems están ordenados de cierta manera.

Una secuencia a = <a_1, a_2, . . . , a_m> es una sub-secuencia de otra b = <b_1, b_2, ... , b_n> $(a \sqsubset b)$, si y sólo si $\exists i_1, i_2, \ldots, i_m$ tal que 1 <= i_1 < i_2 < ...< i_m <= n y $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, ... , $a_m \subseteq$ bim. Se dice además que b es una supersecuencia de a, y que b contiene a a.

Dada una base de datos D = {s_1, s_2, . . . , sn}, el soporte de una secuencia a es el número de secuencias en D que contienen a. Si el soporte de una secuencia *a* satisface un umbral mínimo de soporte min_sup, *a* es un patrón secuencial frecuente.

En [Srikant, Agrawal 1996] se propone GSP, una versión mejorada para el descubrimiento de patrones secuenciales. En ella se incluye la propiedad de cierre hacia abajo, además de las restricciones de tiempo y la ventana deslizante.

Una versión vertical es propuesta en [Zaki 2001]; se trata de SPADE (Sequential PAttern Discovery using Equivalence classes). SPADE, utiliza propiedades combinatorias para descomponer el problema original en sub problemas más pequeños, que pueden ser resueltos por separado en la memoria principal, utilizando técnicas de búsqueda en rejilla y usando operaciones tipo unión (join) simple. Para descubrir todos los patrones secuenciales existentes en el conjunto de datos, realiza tres escaneos de la base de datos. La estructura utilizada para llevar a cabo las operaciones en memoria principal es la rejilla, que es dividida en sub rejillas en las que se realizan dos tipos de búsqueda: en amplitud y en profundidad.

2.2.5. Clasificación basada en patrones

El uso de patrones en modelos predictivos es un asunto que ha acaparado mucho la atención de los investigadores en los últimos años [Bringmann et al. 2009].

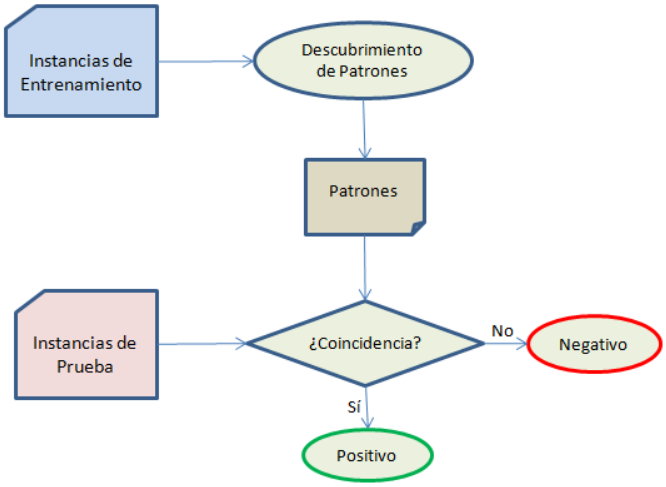

Figura 2.23. Modelo General de Clasificación Basada en Patrones

El descubrimiento de patrones en series temporales puede ser de mucha ayuda para la obtención de modelos más precisos e interpretables. La clasificación basada en patrones (figura 2.23) es un proceso aprendizaje de un modelo en el cual los patrones son usados como características [Kim et al. 2010]; por tal razón, la clasificación basada en patrones es un claro ejemplo de la utilidad de éstos en el ámbito del *data mining* y del éxito alcanzado mediante su uso en diversos métodos.

A continuación se describen algunas de las propuestas.

2.2.5.1. Patrones Emergentes

En [Dong, Li 1999] se introduce un nuevo tipo de patrones, denominados Patrones Emergentes (EPs), que se caracterizan por ser conjuntos de ítems cuyo soporte crece significativamente de un conjunto de datos a otro. Por definición, estos patrones pueden capturar tendencias emergentes en bases de datos temporales, o contrastes útiles entre clases de datos, como por ejemplo, femenino versus masculino, enfermo versus sano, etc.

Teniendo en cuenta que la propiedad Apriori no es válida para EPs y que, normalmente, hay muchos candidatos en bases de datos de alta dimensionalidad o para pequeños umbrales (por ejemplo 0,5 %), los algoritmos Naive son muy costosos, razón por la cual se propone una solución basada en dos aspectos:

(a) Descripción de grandes colecciones de ítems usando sus bordes, esto es, el par de conjuntos de mínimos y máximos en las colecciones.

(b) Diseñar algoritmos, para descubrir EPs, que manipulen solo bordes de colecciones y que representen los EPs descubiertos usando bordes.

Con los algoritmos mencionados en (b), es posible descubrir todos los EPs que satisfacen una restricción, con grandes conjuntos de ítems como datos de entrada.

2.2.5.2. Patrones Emergentes de Salto

Partiendo del trabajo publicado en [Dong, Li 1999], en [Li, Ramamohanrarao 2000] se realiza una mejora al modelo de clasificación mediante el uso de un nuevo tipo de patrones denominados Jumping Emerging Patterns (JEPs), a partir de los cuales se construye un clasificador denominado JEP-Classifier que usa el concepto de borde para almacenar y manipular los JEPs. Un JEP es un EP modificado de tal manera que sea capaz de realizar discriminaciones entre clases de forma más firme que otros tipos de EPs.

La figura 2.24 ilustra cómo trabajan los JEPs. Considérense dos conjuntos de datos de entrenamiento D1 y D2, tales que todas las instancias de D1 pertenecen a la clase 1 y las de D2 pertenecen a la clase 2. Sea cada instancia un subconjunto de {a,b,c,d,e}. La pregunta es: ¿A qué clase pertenece el conjunto {a,b,c}?

Analizando la figura 2.24 se puede determinar que la instancia de prueba {a,b,c} contiene {a,b} que es un JEP con un soporte de 50% en D_2. Además, los subconjuntos propios de {a,b,c} (que son {a}, {b}, {c}, {a,c} y {b,c}) se encuentran tanto en D_1 como en D_2 pero con iguales soportes. Por consiguiente se puede afirmar, con un buen grado de confianza, que la instancia de prueba {a,b,c} se clasifica como Clase 2, pues la suma de los impactos de cada JEP en ambos conjuntos de datos dan una ventaja a la Clase 2.

D1				
a		c	d	e
a				
	b			e
	b	c	d	e

D2				
a	b			
		c		e
a	b	c	d	
			d	e

Figura 2.24. Conjuntos de datos D_1 y D_2.

2.2.5.3. Clasificación Basada en Múltiples Reglas de Asociación

Esta propuesta, CMAR (Classification based on Multiple Association Rules) [Li et al. 2001], consiste en un método de clasificación asociativa y se motiva, principalmente, en la existencia de un número excesivo de patrones y el sesgo que se produce en la clasificación al basarse únicamente en un patrón de alta confianza.

CMAR se basa en FP-growth, pues divide el proceso en las mismas dos etapas. Construye un árbol de patrones frecuentes, denominado FP-Tree, para realizar el descubrimiento de patrones en la base de datos. Adicionalmente, utiliza una estructura de árbol, CR-Tree, para almacenar y manipular las reglas de asociación encontradas y podar las reglas sobre la base de la confianza, correlación y la cobertura de la base de datos. Finalmente, la clasificación se realiza por medio de un análisis ponderado usando las múltiples reglas de asociación seleccionadas.

2.2.5.4. Clasificación basada en Reglas de Asociación Predictivas

Para eliminar los problemas generados por la clasificación asociativa y evitar la imprecisión y el sobreajuste de la clasificación basada en reglas, la propuesta realizada por Yin, CPAR (Classification based on Predictive Association Rules), combina las ventajas de la clasificación asociativa y la clasificación tradicional basada en reglas [Yin et al. 2003]. En lugar de generar una gran cantidad de reglas, como lo hace la clasificación asociativa, CPAR genera directamente

un conjunto más pequeño de reglas predictivas de alta calidad, a partir de los datos de entrenamiento. Adicionalmente, CPAR genera y prueba más reglas que los clasificadores tradicionales basados en reglas, para evitar la pérdida de reglas importantes. Para eliminar el sobreajuste, CPAR utiliza la precisión esperada para evaluar cada regla y usa las k mejores reglas para predicción de la etiqueta de clase de un ejemplo de prueba.

2.2.5.5. Minería de Patrones Discriminativos Numéricos

Las características numéricas de un conjunto de datos son aquellas que toman en cuenta la cantidad de veces que un patrón se repite en una estructura, al contrario de las características binarias que solamente tienen en cuenta si existe o no el patrón. El método NDPMine (Numerical Discriminative Pattern Mining) [Kim et al. 2010] intenta solventar dos grandes problemas existentes al momento de encontrar patrones como características de un conjunto de datos, los cuales tienen que ver con el alto coste que implica encontrar las características numéricas y los problemas de precisión que implican las características binarias.

Esta técnica emplea un método de programación matemática que explora directamente patrones discriminativos como características numéricas y se fundamenta en dos elementos: una medida de la potencia discriminativa de los patrones y un límite teórico de la medida para la poda del espacio de búsqueda.

2.2.5.6. Patrones Temporales Predictivos Mínimos

El método que propone [Batal et al. 2011] para clasificación de series temporales multidimensionales se basa en dos técnicas: primero, una abstracción temporal de los datos y, a continuación, una minería de patrones temporales para la obtención de las características de clasificación.

La principal contribución de este método radica en la capacidad del mismo para obtener solamente un conjunto de patrones relevantes para la clasificación. Estos patrones son capaces de describir relaciones temporales entre múltiples series temporales.

El marco de software creado para el fin descrito se denomina minimal predictive temporal patterns (MPTP), que se basa en pruebas estadísticas para filtrar, de forma efectiva, los patrones no predictivos y espurios.

2.2.5.7. Clasificación Asociativa

Mientras que en el descubrimiento de reglas de asociación no existe un objetivo predeterminado, en el descubrimiento de reglas asociativas de clasificación sí que existe una clase

objetivo. En esta propuesta, Liu integra los dos métodos para extraer ciertas reglas de asociación específicas llamadas reglas de asociación de clase, denominadas CARs [Liu, Hsu 1998].

El método propuesto consta de dos partes. En la primera, el algoritmo publicado en [Agrawal, Srikant 1994] es modificado para descubrir solamente las asociaciones relacionadas con la clase objetivo, es decir, las CARs. Este algoritmo es denominado CBA-RG (Classification Based on Associations - Rule Generator). Y en la segunda parte, se presenta el algoritmo para construir el clasificador sobre la base de las CARs, denominado CBA-CB (Classification Based on Associations - Classifier Builder).

CAPÍTULO 3. PLANTEAMIENTO DEL PROBLEMA

3.1. Motivación

Existe mucho conocimiento oculto en las bases de datos de los sistemas que se encuentran operando en las empresas e industrias del planeta, conocimiento éste que en muchos casos no ha sido descubierto por falta de decisión o de herramientas que permitan hacerlo de la forma menos complicada. La extracción de ese conocimiento puede suponer el logro de mejoras significativas en la planificación, administración y operación de tales empresas; por ejemplo, si en una siderúrgica logramos descubrir la razón del desgaste repetitivo de un rodamiento de una máquina importante, se podría resolver el problema atacando la causa y con ello la empresa se ahorraría las interrupciones abruptas, el tiempo que pierde en realizar el cambio de la pieza y además aseguraría la continuidad en la producción.

Problemas como el descrito en el párrafo anterior se pueden solucionar mediante un proceso de análisis exhaustivo de los registros de funcionamiento de la máquina en cuestión, para reconocer las causas del fallo que está generando el desgaste y asociarlo con el nivel de rendimiento del equipo. Lo menos que se puede lograr con este procedimiento será la capacidad de solucionar el problema a tiempo, gracias a un diagnóstico temprano.

Muchos son los dominios en los cuales el diagnóstico es un fin. Un ejemplo típico es la medicina, en el cual los síntomas que muestra el paciente se asocian a un diagnóstico y éste a su vez permite determinar el tratamiento. En muchas ocasiones, conocer los síntomas no es suficiente y los galenos tienen que recurrir a exámenes muy sofisticados de alta tecnología, que dejan un registro que, al ser analizado por el experto, permite despejar las dudas que pudieran existir. Algunos de los exámenes más conocidos son los Electrocardiogramas, Electroencefalogramas, Potenciales Evocados, Exámenes Isocinéticos, Emisiones Otoacústicas, entre otros. Todos ellos generan series temporales, que son susceptibles de ser analizadas usando técnicas de *data mining* para encontrar patrones, los cuales podrían ser de gran ayuda para los médicos a la hora de elaborar un diagnóstico.

Habitualmente, el experto no se fija tanto en los valores de la serie en cada punto sino en el comportamiento de ésta en determinadas zonas (ciertos cambios de trayectoria o cambios en la forma de la serie, la ocurrencia de determinados eventos, o cosas así). Por tanto, mejor que analizar la serie numérica completa sería centrarse sólo en aquellas cosas que son relevantes para el experto. Esto es lo que conseguimos mediante el proceso de transformación de la serie numérica en una secuencia simbólica. Este proceso de simbolización conlleva, además, una interesante reducción de dimensionalidad.

3.2. Descripción del problema

En casi todas las áreas del quehacer humano, la experiencia es el mejor medio para acumular conocimiento acerca de las actividades propias del dominio, con la particularidad de que ese conocimiento, por lo general, lo posee únicamente el experto, quien lo usa para diagnosticar problemas relacionados con ese dominio. La inclusión de tal conocimiento en los algoritmos de *data mining* se vuelve una tarea compleja, en gran parte debido a que no se maneja un lenguaje único, tanto en el área de *data mining* como en el área del dominio, que facilite la creación de una base de conocimiento con la cual implementar algoritmos de *data mining* eficientes. Por lo tanto, la clave para lograr una captación plena de las características del dominio radica en la creación de herramientas que permitan una interacción natural entre el equipo técnico de *data mining* y el experto en el dominio específico.

En el siguiente capítulo se presenta un método desarrollado para el descubrimiento de patrones en series temporales y la clasificación de nuevos ejemplos basada en tales patrones. La idea principal radica en crear representaciones simbólicas de las series temporales, que integren conocimiento basado en la forma de la serie, esto es, que represente las series tal cual son observadas por el experto en el dominio, mediante el uso de elementos básicos como picos, valles, subidas, bajadas, constantes, etc. y otros más complejos como crestas, intervalos, ondas, etc. Las formas mencionadas se representan como secuencias de símbolos, que pueden ser cualificados conforme a sus características básicas como tamaño, orientación, localización, altura, etc., por ejemplo: grande, bajo, retardado, horizontal,... La implementación de esta idea permitirá al experto en el dominio la cualificación de los símbolos y la creación de nuevos símbolos más descriptivos del dominio específico; con ello se logrará incluir en el método un componente semántico consistente y significativo.

La representación de la serie de la manera descrita anteriormente permitirá una intervención más activa y certera del experto en el dominio, pues una de las grandes dificultades para incluir ese criterio en los sistemas radica en la distancia que existe entre el método conceptual y la realidad del dominio del problema; si se minimiza esa distancia, la interacción con los usuarios del dominio será más directa y servirá de gran ayuda para resolver uno de los inconvenientes anotados en [Padmanabhan, Tuzhilin 1998] y en [Olszewsky 2001], que tiene que ver con la dificultad que se presenta al intentar incluir tal conocimiento en los métodos de *data mining*.

Por lo expuesto, el método que se propone ha de proveer:

- un método apropiado para transformar las series temporales en secuencias de símbolos, logrando así un nivel de abstracción conveniente;

- un método para el descubrimiento de patrones simbólicos (conjuntos de símbolos consecutivos que caracterizan clases), a partir de las secuencias simbólicas; y,
- un método para clasificar instancias nuevas, sobre la base de los patrones descubiertos.

3.3. Restricciones del método

El alcance que se pretende dar al presente trabajo es el de servir como un marco para la clasificación de series temporales, sobre la base de patrones frecuentes basados en subsecuencias, o sea, patrones que representen sub secuencias que aparecen en una secuencia temporal un número de veces superior a un umbral predefinido. Por lo tanto, el método se circunscribe al ámbito de las series temporales de dominios en los cuales se cumplan las siguientes restricciones:

- por tratarse de un proceso de clasificación supervisado, es necesario que, para cada nueva clase definida, se cuente con series temporales previamente clasificadas para la extracción de los patrones que permitirán la posterior clasificación de nuevos ejemplos;
- el método trabaja con series almacenadas en bases de datos, por lo tanto no se aplica para streams en línea;
- las series temporales deben caber en la memoria principal del ordenador, pues los algoritmos han sido diseñados sobre la base de estructuras de datos en memoria.

CAPÍTULO 4. MÉTODO PROPUESTO

La finalidad del método que se propone en la presente tesis, al que se ha llamado SPC (Symbolic Pattern-based Classification), es clasificar individuos (series temporales) de una población, sobre la base de patrones representativos de los grupos de individuos de tal población. Primero, mediante un proceso de simbolización, se transforman las series temporales en un conjunto de series simbólicas, denominadas *secuencias simbólicas*; luego se identifican los patrones representativos de un conjunto de casos de entrenamiento que pertenecen a una clase (grupo de población, diagnóstico específico, etc.) y finalmente, estos patrones son usados para clasificar nuevos casos.

Dado que, en la presente tesis, a fin de viabilizar el análisis, las series temporales son transformadas en secuencias simbólicas, es importante aclarar que el término patrón se refiere a una subsecuencia de símbolos que aparecen en un porcentaje relativamente alto de secuencias simbólicas de la misma clase. Si podemos estar seguros de que un patrón ocurre en un alto número de casos de entrenamiento del grupo objetivo y que no ocurre en los casos de entrenamiento del grupo de control, o viceversa, la presencia/ausencia del patrón respectivo puede ser usada como atributo para clasificar individuos del grupo objetivo. Este proceso tiene una importancia indiscutible en medicina y en otras áreas, donde el diagnóstico es primordial. Por ejemplo, en medicina, si el grupo objetivo se compone de pacientes con una enfermedad conocida y el grupo de control se compone de pacientes sanos, el patrón puede ser un síntoma útil para el diagnóstico de la enfermedad.

El método ha sido configurado en torno a tres procesos: transformación simbólica, descubrimiento de patrones y clasificación. El punto clave del método radica en el uso de patrones simbólicos para representar de la mejor manera el conocimiento contenido en cada serie temporal y cerrar la brecha existente entre los expertos del dominio y las aplicaciones computacionales. Se ha determinado que esto es útil para explicar los resultados a los expertos en su propia terminología [Alonso et al. 2012], además de que, adicionalmente, la transformación de series numéricas a simbólicas reduce drásticamente la dimensionalidad de las series temporales y esto permite el ahorro de tiempo y recursos computacionales para la ejecución de los otros procesos.

El método SPC consta de los tres procesos que se resumen a continuación y que se detallan en las secciones 4.1, 4.2 y 4.3. Para comprender estos procesos, es necesario que se considere que se cuenta con un conjunto de datos de entrenamiento y que cada clase es representada de igual forma, esto es, como un subconjunto de casos de entrenamiento debidamente etiquetados con respecto a un atributo preestablecido. En el caso que nos ocupa, los datos se etiquetan por el atributo diagnóstico. El esquema de funcionamiento del método SPC es el siguiente:

	PARA CADA CLASE, REPRESENTADA POR UN CONJUNTO DE CASOS DE ENTRENAMIENTO,
Proceso de transformación simbólica	**1. Transformar cada serie temporal numérica en una secuencia temporal simbólica que recoja de forma subyacente el conocimiento del dominio. Éste es un proceso de dos pasos:** 1.1. Transformar cada serie temporal numérica en una secuencia de **símbolos independientes del dominio** (ejemplo: picos, ascensos, etc.) que representan las formas básicas contenidas en las series temporales. 1.2. Transformar los símbolos independientes del dominio obtenidos en 1.1 usando conocimiento experto. Esto incluye eliminar los símbolos irrelevantes y cualificar los demás símbolos según su tipo. El tipo dependerá del tamaño, forma, posición relativa, etc., del símbolo en la secuencia temporal. Este paso genera la secuencia formada por los **símbolos dependientes del dominio** y es la secuencia temporal simbólica con la que se trabajará en los demás procesos.
Proceso de descubrimiento de patrones	**2. Encontrar los patrones frecuentes en el conjunto de secuencias temporales simbólicas:** Buscar subsecuencias que aparecen frecuentemente en las secuencias temporales simbólicas pertenecientes a una misma clase. Estos patrones pasan a caracterizar a la clase.
Proceso de clasificación	**3. Obtener los patrones exclusivos de cada clase:** Para cada clase del grupo objetivo, quitar del conjunto de patrones de la clase aquellos patrones que también pertenecen al grupo de control (pacientes sanos). **PARA CADA INDIVIDUO NO CLASIFICADO,** **4. Clasificar al individuo.** Para ello, proceder como sigue: 4.1. Transformar la serie temporal (individuo) en una secuencia temporal simbólica, siguiendo el proceso del paso 1. 4.2. Clasificar la secuencia temporal simbólica. Para cada clase, buscar la presencia de patrones exclusivos de ella en la secuencia simbólica; si existieran, asignar el individuo a esa clase.

El funcionamiento del método se muestra en la figura 4.1, en la que se pueden ver dos fases bien diferenciadas: construcción del clasificador y clasificación. La fase de construcción del clasificador contiene los procesos de transformación de los casos que conforman el conjunto de entrenamiento y de descubrimiento de patrones sobre el conjunto de casos simbolizados, mientras que la de ejecución del clasificador construido abarca los procesos de transformación de los nuevos casos y clasificación de estos casos simbolizados.

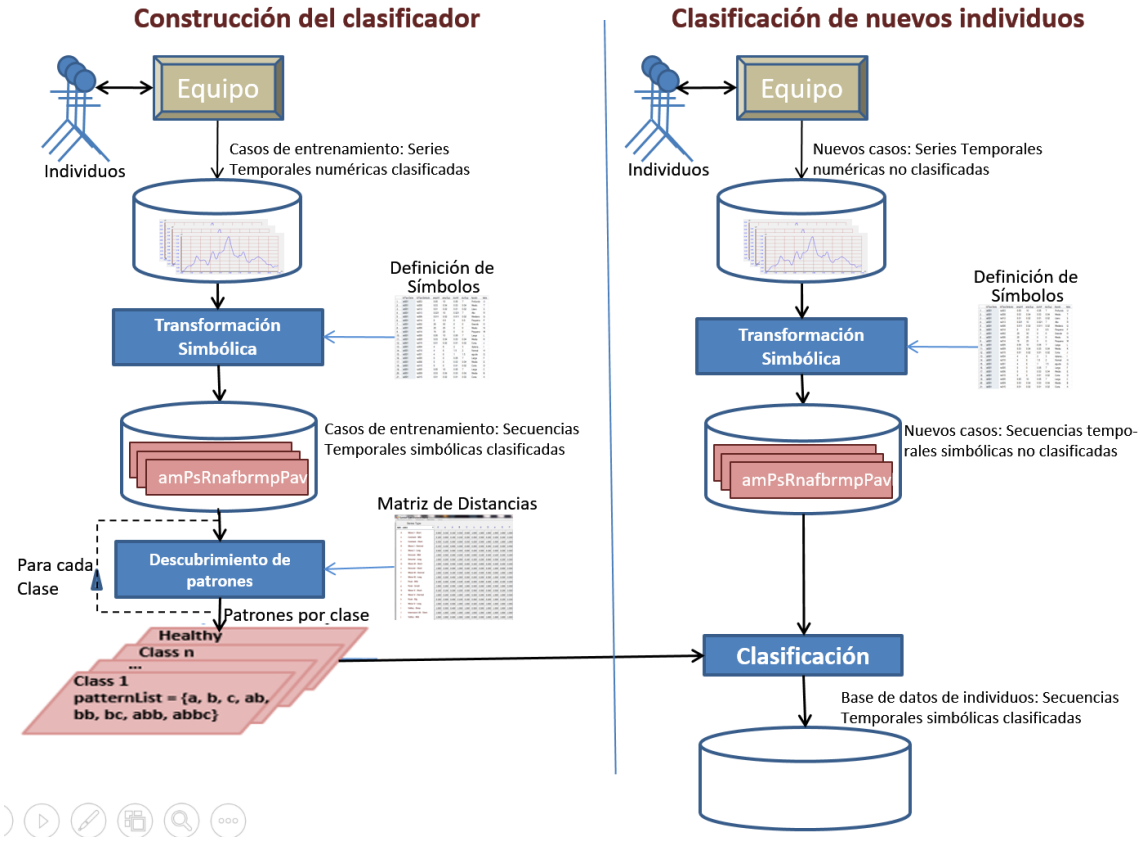

Figura 4.1. Método de clasificación basada en patrones simbólicos (SPC).

4.1. Transformación simbólica

Durante el ejercicio de su profesión, los expertos del dominio adquieren una gran pericia para la interpretación y el tratamiento de los datos que se generan en su trabajo del día a día, lo que les permite resolver problemas cada vez más complejos. En muchos casos, esta pericia está fuera del dominio público. La inclusión de este conocimiento en los algoritmos de *data mining* resulta una tarea compleja [Olszewski 2001], principalmente porque los expertos en *data mining* y los expertos en el dominio hablan diferentes lenguajes técnicos. Ellos tienden a no entenderse entre sí porque utilizan diferentes tecnicismos para referirse al mismo tema y, además, razonan de forma diferente sobre éste, debido a sus diferentes puntos de vista.

A fin de superar esos obstáculos y de contar con el conocimiento experto para garantizar el éxito del método, se ha pensado en transformar las series temporales numéricas en secuencias temporales simbólicas, diseñadas para representar explícitamente el conocimiento que el experto habría abstraído de la serie temporal. Este es un proceso de dos fases, correspondientes a los pasos 1.1 y 1.2 del método propuesto SPC, y que son descritas en las secciones 4.1.1 y 4.1.2, respectivamente.

4.1.1. Transformación simbólica independiente del dominio

La primera fase de la transformación identifica las formas básicas de la serie temporal, conservando la información relevante, usando símbolos independientes del dominio (por ejemplo: picos, ascensos, etc.). Los símbolos independientes del dominio, tienen un cierto número de propiedades cuyos valores son extraídos de la serie temporal numérica. Para esta transformación, usamos un conjunto de símbolos independientes del dominio que corresponden a formas genéricas. La figura 4.2 muestra un ejemplo de los símbolos usados en el método, mientras que en la figura 4.3 se presentan los datos que tienen que ser almacenados para caracterizar un pico (valor inicial, valor final y máximo, amplitud y duración). Esos datos podrían ser diferentes para cada símbolo.

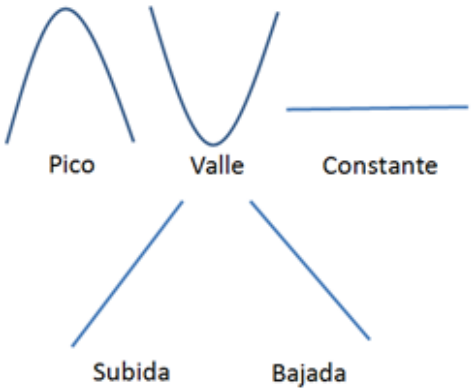

Figura 4.2. Ejemplos de símbolos independientes del dominio (basados en la forma).

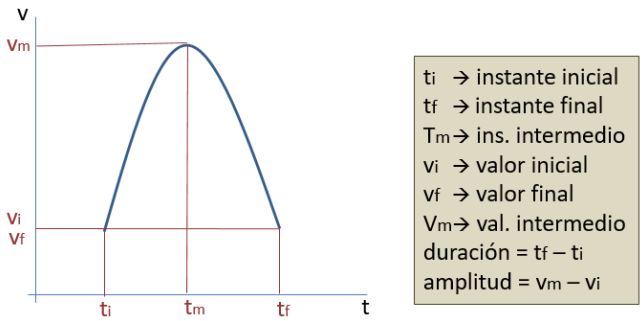

Figura 4.3. Datos requeridos para caracterizar un pico.

Durante esta primera fase, se analiza cada serie temporal numérica en busca de formas básicas contenidas en ella, a fin de transformarla en una secuencia temporal de símbolos independientes del dominio. Se recorre la serie temporal identificando, en una pasada, qué símbolos corresponden a cada segmento de la serie temporal. Para recorrer la serie se utilizan unos punteros auxiliares, con los que se marcan hitos que permiten identificar los diferentes símbolos. Por ejemplo, el puntero *cp* se mueve a lo largo de la serie temporal en busca de cambios de tendencia y puede detectar tres estados diferentes: ascendente, descendente y constante. Un cambio en el estado

(cambio de tendencia) indica la presencia de un pico, de un valle o de un cambio de constante a ascenso o descenso, o viceversa. Cuando un símbolo ha sido detectado, los datos requeridos para caracterizar ese símbolo se calculan y esta información es almacenada en la tabla de símbolos, dentro de la base de datos. Este proceso se detalla en el algoritmo 4.1.

```
Entrada: Serie temporal numérica
Salida: Secuencia temporal simbólica independiente del dominio, SymbolList

SymbolList = ∅
Initialize (cp)                                   // cp apunta al primer punto de ts
While (not end of (ts))
    lp=cp
    status = Advance(cp)        // cp se mueve y se determina el estado (ascendente, descendente, constante)
    cp = NextChangeOfDirection (cp);              // cp se mueve al siguiente cambio de dirección
    Case status = 'Descending'
        s=FindDescentSymbols (cp, lp, mp);        // busca (Descenso || Valle || Descenso + Valle)
    Case status = 'Ascending'
        s=FindAscentSymbols (cp, lp, mp);         // busca (Ascenso || Pico || Ascenso + Pico)
    Case status = 'Constant'
        s=FindConstantSymbols (cp, lp);           // busca (Constante)
    AddIdentifiedSymbols (SymbolList, s, cp, lp, mp)  // añade los símbolos encontrados a SymbolList
End_While
```

Algoritmo 4.1. Transformación simbólica independiente del dominio.

4.1.2. Transformación simbólica dependiente del dominio

Una vez transformada la serie numérica en una secuencia de símbolos independientes del dominio, los símbolos de esta secuencia serán procesados para transformarlos en símbolos dependientes del dominio con sus tipos. Los símbolos dependientes del dominio tienen como objetivo representar los conceptos específicos del dominio contenidos en la serie temporal. A estos símbolos transformados se les atribuirá una cualificación para representar las características relevantes del dominio, entre las que destacan el tamaño, la forma, la ubicación relativa, etc., de los símbolos en la secuencia temporal. Estos símbolos dependientes del dominio serán de utilidad para la toma de decisiones y, además, para explicar los resultados en la terminología propia del dominio.

Los expertos serán capaces de entender fácilmente las explicaciones del proceso de razonamiento y los resultados ya que se utilizan los conceptos del dominio con los que ellos están familiarizados. Esto es especialmente importante en ámbitos como la medicina, donde el diagnóstico es de exclusiva responsabilidad del médico y tiene que ser debidamente justificado. La brecha entre el modelo conceptual y los conceptos que se usan en el dominio será reducida, lo que resulta en una interacción más directa y menos compleja con los expertos del dominio. Esto ayudará a resolver algunos de los problemas mencionados en [Olszewski 2001], [Padmanabhan, Tuzhilin 1998], relativos a la inclusión de experiencia en métodos de minería de datos.

El proceso que se corresponde con el paso 1.2 del método SPC y se detalla en el algoritmo 4.2, utiliza un conjunto de reglas heurísticas para determinar cómo transformar cada símbolo independiente del dominio en un símbolo dependiente del dominio. Téngase en cuenta que éstos pueden tener una correspondencia uno-a-uno o muchos-a-uno, de forma que uno o más símbolos independientes del dominio se transforman en un solo símbolo dependiente del dominio. Un ejemplo simplificado de la heurística (para el dominio PEATC - Potenciales Evocados Auditivos de Tronco Cerebral) se describe en la parte inferior del algoritmo 4.2.

```
Entrada: una secuencia temporal simbólica independiente del dominio T_in
Salida: la secuencia temporal simbólica dependiente del dominio respectiva T_out

For i=1 to Length (T_in)
        H = FindHeuristics (T_in[i])        // Comprueba las heurísticas para el símbolo T_in[i]
        h = SelectHeuristic (H, T_in[i])    // Selecciona la mejor heurística para la subsecuencia T_in[i]..T_in[i+n]
                                            // donde n depende de la heurística
        ds = ApplyHeuristic (h, T_in[i])    // Retorna ds, el símbolo dependiente del dominio resultante, y n
    If ds ≠ null
        Then
            typed_ds = SpecifyType (ds)     // Basándose en la información de la tabla de límites simbólicos
            Add typed_ds to T_out;          // Va construyendo la secuencia temporal dependiente del dominio
            i = i + n;                      // n es el número de símbolos independientes del dominio que
                                            // se corresponden con el símbolo dependiente del dominio ds
        Else i = i + 1                      // Descarta los símbolos independientes del dominio irrelevantes
End For

// Fragmento de heurística en el dominio PEATC
    If T_in[i] = Peak
        Then    If T_in[i].tm ≥ 0.75 ms AND T_in[i].tm ≤ 3.0 ms
                    Then    ds.symbol = WaveI
                            ds.tm = T_in[i].tm
                            ds.amplitude = T_in[i].vf - T_in[i].vi
                            ds.duration = T_in[i].tf - T_in[i].ti
                            n=1
        Else    If T_in[i].tm ≥ 3.01 ms AND T_in[i].tm ≤ 5.0 ms
                    Then    ds.symbol = WaveIII
                            …
    …
```

Algoritmo 4.2. Transformación simbólica dependiente del dominio.

La Figura 4.4 muestra un ejemplo del dominio de la electrocardiografía, donde se puede observar la correlación entre la serie temporal numérica, la secuencia simbólica independiente del dominio y la secuencia simbólica dependiente del dominio. Se esquematiza cómo algunos de los símbolos dependientes del domino coinciden con un solo símbolo independiente del dominio (por ejemplo, el pico de la onda T) o con varios símbolos independientes del dominio (por ejemplo, valle-pico-valle del complejo QRS), y otros son directamente ignorados (por ejemplo, el valle entre la onda T y la onda U). Se debe tener en cuenta que los símbolos dependientes del dominio no sólo coinciden con eventos relevantes que ocurren en la serie temporal, sino que toda la serie

temporal se transforma en una secuencia de símbolos que deben ser equivalentes a la serie temporal numérica original. En otras palabras, con esta transformación se ha realizado un efectivo proceso de abstracción en el cual la serie temporal numérica original ha ganado en riqueza semántica al tiempo que ha disminuido drásticamente en dimensionalidad, conservando, al mismo tiempo, la información relevante contenida en ella.

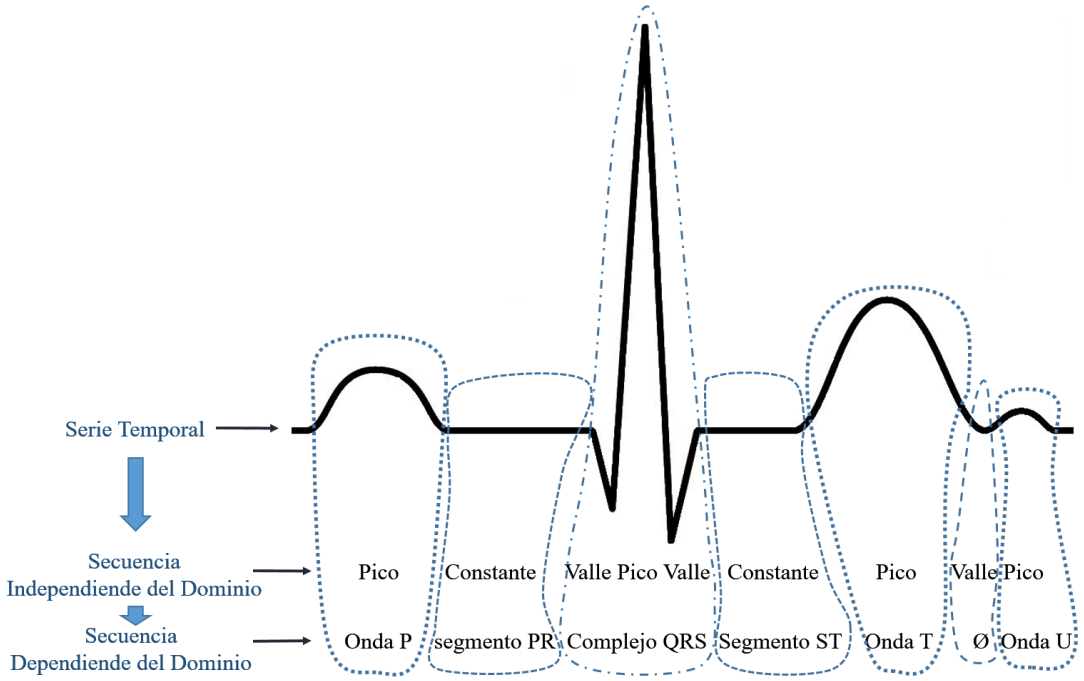

Figura 4.4. Serie temporal de ECG con sus correspondientes secuencias simbólicas.

Los símbolos dependientes del dominio deben ser cualificados. La función SpecifyType() (ver el algoritmo 4.2) toma un símbolo dependiente del dominio y devuelve el respectivo símbolo tipado; para ello usa la información almacenada en la tabla de límites. Esta tabla almacena las condiciones que cada símbolo debe cumplir a fin de que se le pueda asignar un tipo en particular. Por ejemplo, la transformación simbólica dependiente del dominio podría, bajo ciertas circunstancias, convertir la secuencia valle-pico-valle, obtenida por la transformación simbólica independiente del dominio, en un símbolo llamado complejo QRS. Hay varios tipos de complejo QRS; por lo tanto, este símbolo requiere cualificación para definir su tipo. Por ejemplo, para que un complejo QRS sea cualificado como positivo, deberá cumplir ciertas condiciones como son: tener una duración entre 60 y 100 milisegundos, un voltaje no mayor a 3.5V y producirse después de un símbolo P. Otras condiciones más complejas podrían aplicarse a otros símbolos.

Finalmente, la lista de símbolos dependientes del dominio, o sea, la secuencia temporal simbólica, es almacenada en una tabla relacional denominada Symbolic Time Sequences. Para fines de simplicidad, se ha representado cada símbolo tipado como un carácter (una letra minúscula o mayúscula). Por ejemplo, el símbolo tipado complejoQRS.positivo podría ser el

carácter K y complejoQRS.negativo podría ser representado por el carácter L. El conjunto de caracteres utilizado para representar los símbolos dependientes del dominio constituye el alfabeto de símbolos dependientes del dominio.

La adquisición del conocimiento experto se fundamenta en el uso de ciertas técnicas que permiten incluir en el método el conocimiento del dominio, requerido para realizar la transformación simbólica dependiente del dominio. El proceso de adquisición del conocimiento llevado a cabo en el dominio de los PEATC, se describe en la sección 5.1.

4.2. Descubrimiento de patrones

El descubrimiento de patrones en un conjunto de secuencias temporales simbólicas es el núcleo del método. Los patrones que se encuentren en este proceso pueden ser interpretados como características que definen las diferentes clases formadas por los individuos del dominio. Los patrones son subsecuencias que se repiten con alguna frecuencia en un conjunto dado de secuencias temporales. Si estos conjuntos son representativos de grupos relevantes de la población, la presencia repetida de los patrones podría caracterizar a tales grupos, y los patrones podrían, por lo tanto, ser usados en procesos de clasificación para identificar nuevos miembros de esos grupos.

Para caracterizar a un grupo específico de una población, esta técnica propone el descubrimiento de patrones en todas las secuencias temporales de esa población, lo que significa encontrar subsecuencias repetidas tomando en cuenta que una misma secuencia encontrada podría estar presente en diferentes posiciones relativas dentro de las secuencias temporales simbólicas.

Obviamente, es muy improbable que una subsecuencia se repita con, exactamente, los mismos valores en una serie temporal numérica. Inconvenientes tales como los errores inherentes a los sensores de medida, ruido o incluso pequeños cambios en los valores pueden producir diferencias en las series. Sin embargo, esas diferencias pueden carecer de importancia, desde el punto de vista del dominio, en cuyo caso las subsecuencias respectivas deben considerarse iguales. Este problema será considerado parcialmente resuelto una vez que las series temporales numéricas hayan sido transformadas en secuencias simbólicas, gracias a los rangos que existen para caracterizar los símbolos. Sin embargo, se ha pensado que aún es útil considerar algún margen de disimilitud. Por un lado, tales diferencias numéricas en valores cercanos a los puntos que separan un símbolo de otro, podría conducir a secuencias simbólicas distintas y, por otro lado, no tiene sentido requerir patrones largos, compuestos por muchos símbolos, que coincidan perfectamente. Por ello, se ha definido una distancia entre secuencias simbólicas, para lo cual se ha utilizado una matriz de costes *CM* que especifica la distancia entre cada par de símbolos tipados. Esta matriz

de costes permite calcular la distancia entre cada par de secuencias simbólicas *S* y *T de longitud n*. Las distancias se definen de acuerdo con la expresión 4.1.

$$Distancia(S,T) = \sum_{i=1}^{n} CM\ (S_i, T_i)$$

Expresión 4.1. Distancia entre dos secuencias simbólicas.

El descubrimiento de patrones es el núcleo del método propuesto, pues se trata de la fase en la que se extrae el conocimiento de los conjuntos de series temporales. Se define un patrón como una subsecuencia simbólica que aparece frecuentemente en cualquier posición dentro de las secuencias temporales pertenecientes a un conjunto. Para determinar si una secuencia simbólica aparece o no "frecuentemente" dentro de una secuencia temporal, se define un umbral llamado *minsupport*, que denota el porcentaje de secuencias temporales dentro del conjunto en las cuales los patrones aparecen por lo menos una vez. Además, para definir cuando dos subsecuencias se consideran suficientemente similares, se utiliza el parámetro denominado *maxdist*, que define la distancia máxima permitida entre dos subsecuencias que son consideradas iguales.

Se usa un algoritmo incremental para encontrar los patrones que aparecen en un conjunto de secuencias temporales. Este algoritmo fue inspirado en otras técnicas existentes y la propiedad Apriori, pero que necesitaron grandes cambios para cumplir con los objetivos particulares de la presente investigación, debido principalmente a lo siguiente:

- La necesidad de encontrar subsecuencias similares y no solo idénticas, y
- La necesidad de encontrar patrones en un conjunto de secuencias temporales (no solamente en una secuencia temporal), y teniendo en cuenta que dichos patrones pueden aparecer en cualquier posición de las series temporales.

Para representar la información de los patrones de la secuencias temporales simbólicas, el algoritmo usa un cubo de datos (una matriz tridimensional de bits a la que se ha llamada cubo de datos SPC) inspirado en el *cubo de trabajo* de Han [Han et al. 1998]. La idea es construir un cubo de trabajo en el cual un 1 representa la existencia de un valor dado en una cierta posición de una serie temporal mientras que un 0 representa que no existe ese valor en dicha posición. En el ejemplo que se muestra en la figura 4.5, Han divide cada serie temporal en cuatro periodos (Q0 a Q3) de tres meses (mo0 a mo2) y, como el nivel de ganancia para el mes 1 del Q0 fue 683, la celda correspondiente (nivel de ganancia 0-1000) se llena con 1. Esta representación permite realizar de forma eficiente la búsqueda de patrones periódicos, esto es, de segmentos de las series temporales repetidos a intervalos de longitud fija.

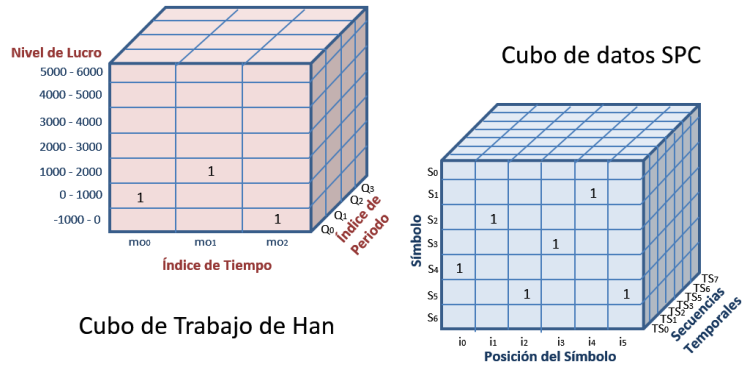

Figura 4.5. Adaptaciones propuestas al cubo de trabajo de Han.

Se ha adaptado este cubo a las necesidades del problema de descubrimiento de patrones en secuencias simbólicas que se aborda en el presente trabajo de investigación. Las diferencias principales entre un caso y otro son:

- En el trabajo de Han se usa el cubo para conseguir patrones que aparecen en un número determinado de periodos contenidos en solamente una serie temporal, mientras que en el método aquí propuesto el cubo se ha adaptado para posibilitar la obtención de patrones presentes en un conjunto indeterminado de series temporales.
- Los patrones periódicos de Han deben estar siempre en la misma posición relativa dentro de la serie temporal; sin embargo, los patrones que se descubren mediante el método aquí propuesto pueden estar localizados en cualesquiera posiciones dentro de las series temporales. Por lo tanto, la idea de abordar el problema de la misma manera que Han, considerando las series temporales diferentes como los diferentes periodos de la serie única de Han, no es aplicable.
- Como consecuencia de lo expuesto, las dimensiones del cubo SPC son posición del símbolo, secuencia temporal y símbolo en vez de índice de tiempo, índice de periodo y valor de la serie, respectivamente. Estos cambios implican grandes diferencias en el proceso de gestión de la información.

Como se ha mencionado anteriormente, la técnica de descubrimiento de patrones es incremental. Primeramente se buscan patrones de longitud 1. Una vez encontrados los patrones de longitud 1, se buscan los patrones de longitud 2. Los patrones de longitud 2 son usados como base para buscar los patrones de longitud 3, y así sucesivamente hasta que ya no sea posible encontrar nuevos patrones. En la presente investigación, un patrón de longitud k se denomina kpatrón.

La técnica de descubrimiento de patrones del método usa la propiedad Apriori; esto significa que si una subsecuencia no es un patrón, ninguno de sus superconjuntos (subsecuencias que la

contienen) puede ser un patrón; en este caso, la respectiva rama del árbol de búsqueda puede ser podada, con lo cual se simplifica el proceso. Sobre la base de la propiedad Apriori, la lista de los 2patrones candidatos se construye mediante la combinación de los 1patrones. El algoritmo entonces revisa cuáles de esos candidatos son suficientemente frecuentes para ser considerados 2patrones. La lista de los 3patrones candidatos se construye combinando cada uno de los 2patrones con cada uno de los 1patrones, y así sucesivamente hasta que la lista de patrones candidatos esté vacía.

El umbral *minsupport* usado para determinar si una subsecuencia es frecuente, puede ser usado para mejorar la eficiencia del algoritmo, usando la propiedad apriori. Como se desea descubrir patrones frecuentes, entonces el valor de *minsupport* es usualmente cercano a 1; se puede por lo tanto usar el complemento a uno de su valor (1-*minsupport*) para descartar las subsecuencias que no pueden ser patrones. De esta manera, cuando *minsupport* es mayor que 0.5, se cuenta el número de secuencias que no tienen el patrón y, cuando este número es superior a 1 - *minsupport*, la rama se descarta (se termina entonces la búsqueda de más ocurrencias para esta subsecuencia). De esta forma se ahorra tiempo computacional para la ejecución del algoritmo de búsqueda de patrones.

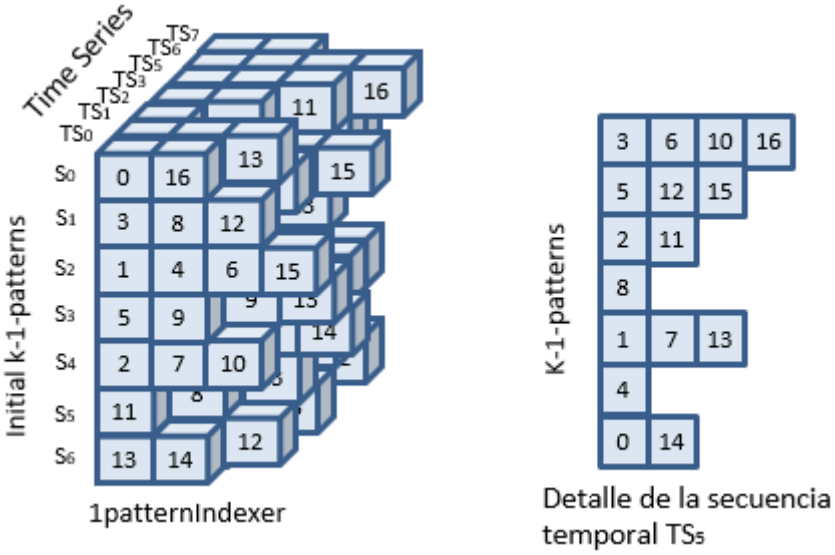

Figura 4.6. El kPatternIndexer para *k*=1.

A fin de acceder con mayor efectividad a las posiciones de la secuencia temporal en las que aparecen los patrones de longitud *k* (kpatrones), se diseñó una estructura de datos llamada kPatternIndexer, que consiste en una matriz bidimensional de listas de enteros, donde cada lista almacena las posiciones de la secuencia temporal TS_i en que comienza cada patrón de longitud *k*. Esta información se usa para encontrar patrones de longitud *k*+1. La figura 4.6 muestra un ejemplo de la estructura de un kPatternIndexer para *k*=1, es decir, para subsecuencias frecuentes compuestas por un solo símbolo S_i. La figura 4.6 muestra también el detalle de un *corte* de esta

estructura, que almacena la información correspondiente a una secuencia temporal, es decir, las posiciones en las que se encuentran sus 1patrones.

El proceso de descubrimiento de patrones propuesto se detalla en el algoritmo 4.3.

Entrada: conjunto de secuencias temporales simbólicas para un grupo poblacional; alfabeto de símbolos dependientes del dominio; *minsupport; maxdist*.
Salida: el conjunto de patrones *patternList*.

```
    Crear el cubo de datos SPC    /* Si el símbolo S_m aparece en la posición i de la secuencia temporal
                                  TS_j, la respectiva celda (i,j,m) se rellena con 1; si no su valor es 0 */
    patternList = ∅
// INICIALIZACIÓN: Construcción de patternList para k=1 y del 1PatternIndexer
    k=1
    For each time sequence
      For each symbol s in the time sequence
        Save its position in the 1PatternIndexer
        If any symbol s is not present in at least (1-minsupport) *n  secuencias        /* n =
                                                          número total de secuencias temporales */
        Then prune symbol s    // s no será considerado 1patrón y se elimina del 1PatternIndexer
        Else  patternList = patternList ∪ {s}        // añadir el 1patrón s a patternList
      End For
    End For

/* ITERACIÓN: Con las estructuras de datos obtenidas de la iteración k-1, encontrar los kpatrones y
construir el kPatternIndexer */
    Repeat
      k=k+1                                  //en la primera iteración k será 2
      kPatternIndexer = ∅
      For each k-1pattern
        For each 1pattern
          kpc = k-1pattern ⊕ 1pattern        // kpc es un candidato a kpatrón
          NotFound_kpc_Count = 0;
          For each time sequence TS_j
            For each position p of the k-1pattern registered in the k-1PatternIndexer
              If located (kpc, p)            /* comprueba si la distancia entre kpc y los símbolos de
                                                la posición p a p+k-1 es menor que maxdist */
                Then
                  include TS_j and p in tempIndex;
                  found = true
            End For
              If not found                   // el candidato kpc no está en la secuencia temporal TS_j
                Then
                  NotFound_kpc_Count ++
                  If NotFound_kpc_Count > (1 – minsupport) * n
                    Then Break                                  // Poda: kpc no puede ser frecuente
          End For
          If NotFound_kpc_Count > (1 – minsupport) * n  // si el candidato kpc no es frecuente
            Then Break
            Else
```

```
            patternList = patternList ∪ {kpc}
            Add tempIndex to kPatternIndexer for kpc
        End For
    End For
Until (kPatternIndexer is empty)        /* No hay patrones de longitud k así que no se
                                           pueden encontrar más patrones nuevos */
```

Algoritmo 4.3. Descubrimiento de patrones.

La figura 4.7 ilustra cómo trabaja el algoritmo para el descubrimiento de patrones. La entrada es un conjunto de secuencias temporales simbólicas que pertenecen al mismo grupo poblacional y el objetivo es encontrar patrones que caractericen al grupo.

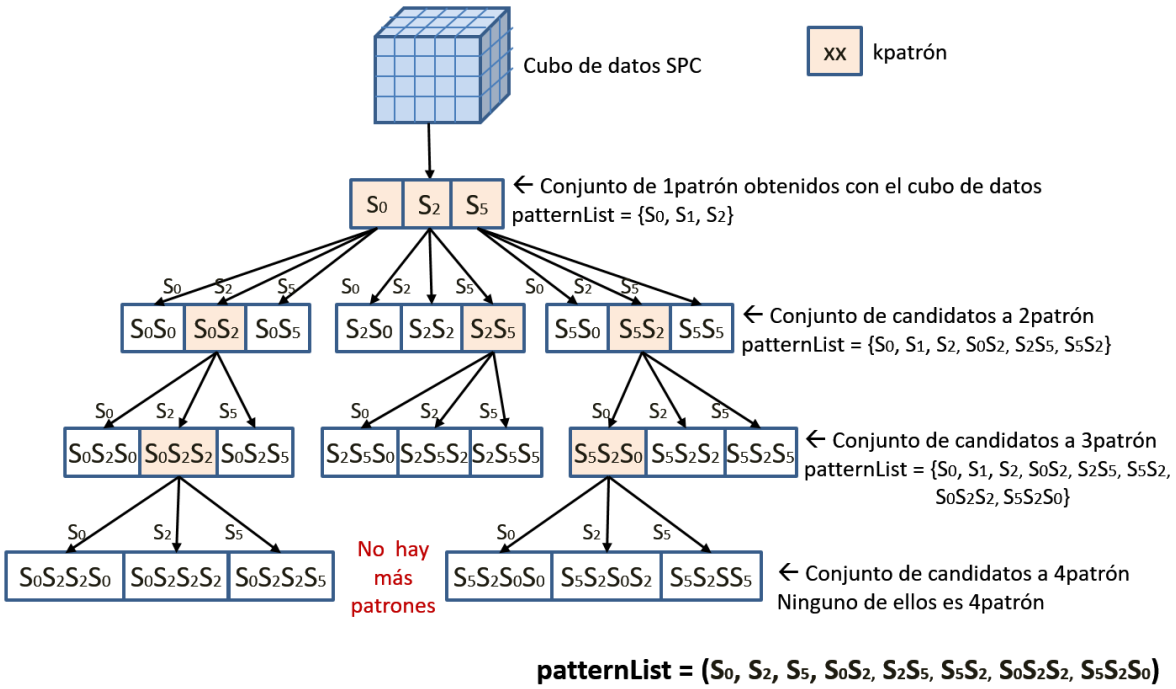

Figura 4.7. Ejemplo de descubrimiento de patrones.

En este ejemplo, se asume que existen siete símbolos en el dominio, tres de los cuales (S_0, S_2 y S_5) aparecen frecuentemente en las secuencias temporales. Por lo tanto, esos son 1patrones y son añadidos a *patternList*. Los 2patrones candidatos se generan mediante la combinación de cada 1patrón con otro 1patrón. Se supone que los 2patrones frecuentes, de esta lista de candidatos a 2patrones, son lo que se han resaltado en la figura 4.7. Se añaden estos 2patrones frecuentes a la *patternList*. Este procedimiento se repite hasta que no se encuentren más kpatrones. En este ejemplo, se ha supuesto que esto ocurre en el nivel 4, donde el *kPatternIndexer* está vacío y no se harán más cambios a la lista de patrones (*patternList*). El proceso de descubrimiento de patrones termina en este punto, y la lista que contiene los ocho patrones diferentes es la salida del algoritmo. Cabe destacar también que los 2patrones S_2S_5 y S_5S_2 son diferentes, ya que se trata de

símbolos con una secuencia temporal relevante. Esto distingue también a este método de otras técnicas donde los patrones se consideran "bolsas" de eventos donde el orden no importa.

Para que el descubrimiento de patrones pueda ayudar a los expertos del dominio en el proceso de diagnóstico, los patrones que pueden caracterizar a un grupo particular de individuos tienen que ser separados de los patrones que son comunes a todos los individuos. La idea es aplicar el método de descubrimiento de patrones a un conjunto de series temporales que representa una población objetivo determinada (en un dominio relacionado con la medicina, se podría referir a una población de pacientes con un trastorno de salud específico) y al conjunto de series temporales que corresponde a un grupo de control (en el caso de la medicina, los individuos sanos sin ningún trastorno). Los patrones que se encuentran en el grupo de control representan el comportamiento normal. Por tanto, si esos patrones aparecen también en el conjunto de patrones de la población objetivo, serán eliminados con el objeto de determinar cuáles son los patrones exclusivos de la clase objetivo. De esta forma, se puede garantizar que esos patrones están asociados con el comportamiento que se intenta caracterizar y, por lo tanto, pueden ser usados como atributos para clasificar futuros individuos, aún no clasificados. Si hubiera más de una clase objetivo, este proceso de obtención de los patrones exclusivos de la clase debe ser repetido para todas y cada una de las clases.

4.3 Clasificación

El proceso de clasificación puede comenzar una vez que se hubieren generado los patrones exclusivos de cada clase. Para clasificar una secuencia simbólica TS, se explorará dicha secuencia (ver el algoritmo 4.4) en busca de los patrones exclusivos de cada clase.

```
Entrada: secuencia temporal a ser clasificada TS
Salida: la(s) clase(s) respectiva(s)

For each class C
  For each Pattern p in ExclusivepatternList[C]
    If present (p, TS, maxdist)          /* Busca en TS un patrón similar a p (la diferencia
                                            entre ambos no puede superar el parámetro maxdist) */
    Then Add p to List;
    If Coincident(List, ExclusivepatternList[C], ppc)     /* Verifica si la lista de patrones
                                                             encontrados en TS coincide con los patrones
                                                             exclusivos de la clase (al menos en un ppc%) */
    Then Assign TS to class C;
  End For
End For
If TS not assigned to any class
Then Assign TS to unclassified sequences
```

Algoritmo 4.4. Clasificación de una secuencia temporal

Con el uso del parámetro ppc (porcentaje de patrones para clasificación), el método propuesto permite ajustar el porcentaje de coincidencia entre los patrones exclusivos de una clase y los patrones encontrados en la TS que es necesario para asignar la secuencia a esa clase. Este proceso debe repetirse para cada clase. Como resultado de este proceso, la TS podría terminar siendo asociada con más de una clase. Otra posibilidad es que la TS no pueda ser asignada a ninguna clase; en tal caso, será etiquetada como "no clasificada". El algoritmo que explora la TS también usa el parámetro maxdist. Esto hace la búsqueda de coincidencias en la TS más flexible, es decir, que permite que se considere que una subsecuencia de la TS coincide con un patrón exclusivo de una clase aunque ambos no sean idénticos. El nivel de flexibilidad puede ser ajustado por el usuario, dependiendo de las características del dominio. El parámetro maxdist representa la máxima distancia admisible para decidir si la TS contiene o no el patrón. Si maxdist = 0, la TS debe contener una coincidencia exacta del patrón.

La figura 4.8 ilustra el proceso de clasificación para una clase objetivo en un dominio médico. El lado izquierdo de la figura muestra los patrones encontrados para un trastorno en particular ("trastorno x") y los patrones para el grupo de control (pacientes "sanos"). Aquellos patrones que se repiten en ambos conjuntos no servirán para caracterizar a los individuos, es decir, para decidir a qué clase pertenecen. Por lo tanto, los patrones exclusivos de clase "trastorno x" son los que se usan para clasificar un nuevo paciente (la secuencia del nuevo paciente, como se ilustra en la figura, contiene todos los patrones del trastorno estudiado y es, por lo tanto, etiquetada como "trastorno x").

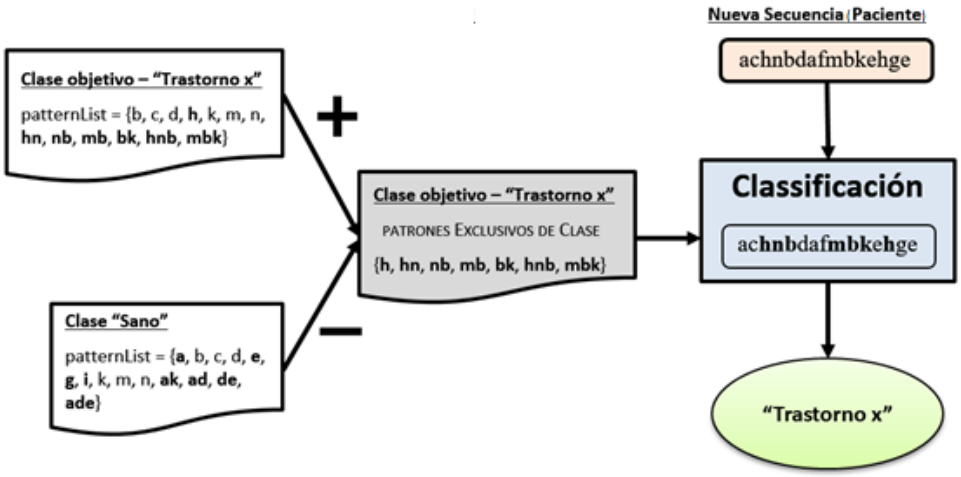

Figura 4.8. Proceso de clasificación basada en patrones para una clase objetivo.

Dependiendo de las características del dominio, puede ser suficiente o aún mejor usar solamente los patrones más largos para el proceso de clasificación, ya que éstos usualmente contienen a la mayoría de los patrones más cortos. Esto es verdad cuando los símbolos tienen

algún tipo de relación entre sí (en el caso que nos ocupa, la relación es dada por la línea temporal), y los patrones más largos contienen más conocimiento. En otras palabras, en los patrones más largos se tienen en cuenta tanto la presencia de los patrones temporales más cortos como las relaciones temporales entre dichos patrones. Los patrones más largos pueden no tener ninguna ventaja en otros dominios, como el carrito de compras, en el que los ítems no se relacionan.

CAPÍTULO 5. EXPERIMENTACIÓN

5.1. Del dominio de conocimiento y la recogida de datos

Para evaluar el método propuesto se ha elegido el dominio de los potenciales evocados auditivos de tronco cerebral (PEATC). La meta principal en este dominio es diagnosticar los problemas auditivos y tumores relacionados con el tronco cerebral, a través del análisis de las respuestas de los pacientes a estímulos acústicos. Estas respuestas consisten en potenciales electrofisiológicos (PEATC) registrados por equipos de alta precisión, conectados al paciente mediante sensores fijados a su cabeza en tres puntos, y almacenados finalmente en la memoria secundaria de un ordenador en forma de series temporales con una duración entre 0 y 15 ms. Se denominan Respuestas Auditivas de Tronco Cerebral de latencia corta (Short Latency Auditory Brainstem Response, o ABR), pues es en este corto periodo de tiempo cuando la zona del tronco cerebral relacionada con el proceso de audición emite respuesta a un estímulo sonoro. Conforme transcurre el tiempo, son otras zonas del cerebro las que irán respondiendo al estímulo hasta llegar al córtex cerebral, donde se emite la respuesta de más alto nivel. En la experimentación realizada para el presente trabajo se usan ABRs de 12 ms de duración con puntos muestreados cada 16 microsegundos.

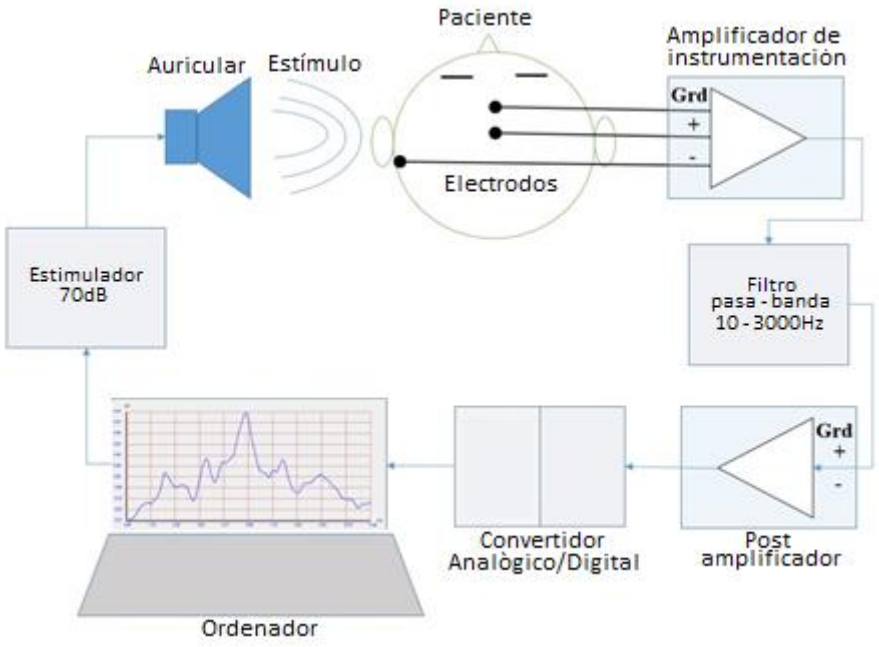

Figura 5.1. Esquema de recogida de datos de potenciales evocados auditivos de tronco cerebral.

5.1.1. Los datos para la experimentación

La figura 5.1 muestra el montaje de los dispositivos para la captura y medición de los PEATC. Típicamente, el estímulo acústico que se envía al sistema auditivo del paciente es un clic. La respuesta es un conjunto de potenciales cuyos valores están asociados con instantes de tiempo.

Puesto que hay innumerables contaminantes en la señal de respuesta, el paciente en realidad es estimulado con alrededor de 2000 clics (el número exacto de clics dependerá de las condiciones del medio en que se realiza el examen y de la calidad de los equipos usados para la captura de las señales). En la presente experimentación se estimuló a los pacientes con exactamente 2000 clics, en todos los casos. A partir de los 2000 ABRs obtenidos, se calcula un ABR promedio, con lo cual el ruido también se promedia, de modo que se torna poco significativo y se obtiene una señal con mayor nivel de pureza.

Una característica importante del clic es su intensidad. Normalmente, un equipo especializado puede aplicar un estímulo de hasta 120 dB (decibelios) y un mínimo de 30 dB. Pero en la práctica, las pruebas de PEATC son ejecutadas, típicamente, como es el caso de la presente experimentación, con intensidades de 70 dB. Valores alternativos podrían ser aplicados, dependiendo de las condiciones del paciente; por ejemplo, un paciente con niveles auditivos cercanos a la normalidad podrá ser examinado con una intensidad de 70 dB, mientras que para un paciente con pérdida auditiva se considerará la posibilidad de incrementar la intensidad dependiendo del nivel de la pérdida. Se conocen casos de pacientes con pérdidas auditivas severas, que fueron examinados con intensidades de 110 y 120 dB. Es importante recalcar que la exposición prolongada del paciente a estímulos sonoros con intensidad elevada puede acarrear mayores daños al sistema auditivo del paciente.

En la investigación desarrollada, con el aval del profesional experto en el dominio se decidió introducir solamente tres de estas ondas (onda I, onda III y onda V) con los correspondientes intervalos entre ellas (intervalo I-III, intervalo I-V e intervalo III-V) como nuevos símbolos dependientes del dominio. En realidad, un ABR de latencia corta contiene cinco ondas con sus respectivos intervalos, pero los expertos en el dominio utilizan solamente las ondas I, III y V para efectos de diagnóstico; las ondas II y IV, en ese contexto, se consideran redundantes.

La figura 5.2 muestra la serie temporal obtenida con la aplicación de un estímulo acústico de 2000 clics a una intensidad de 70 dB, y en ella se puede distinguir la presencia de las ondas y los intervalos mencionados en el párrafo anterior, los mismos que encajan con los símbolos dependientes del dominio, como se explica más adelante en esta sección.

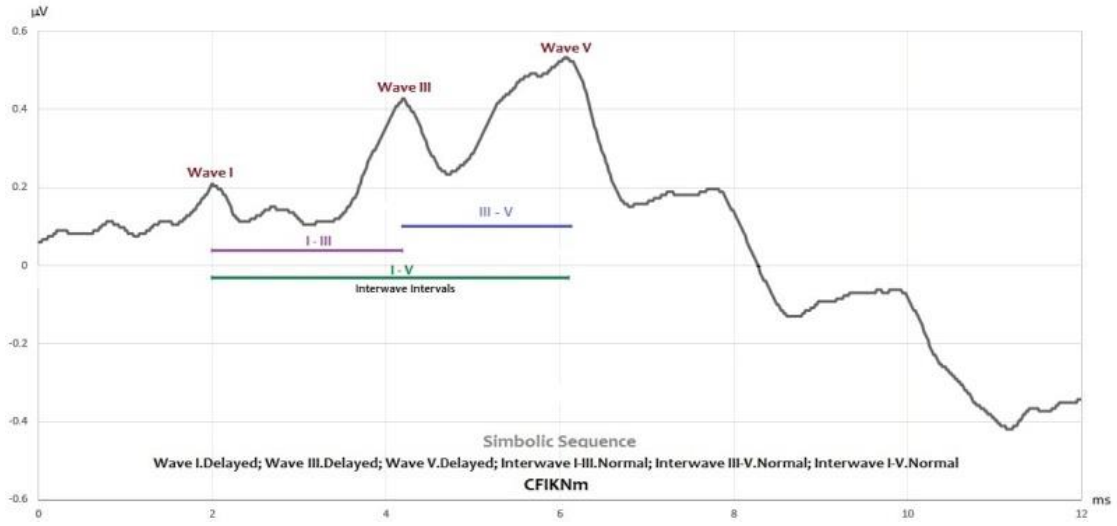

Figura 5.2. Serie temporal de un ABR registrado a una intensidad de 70 dB.

Las entrevistas con el experto revelaron que una serie temporal PEATC tiene unos picos relevantes, denominados ondas, que se etiquetan con números romanos. Para identificar una onda, es necesario conocer el instante en que fue generada, a lo que se denomina latencia de la onda. El intervalo de tiempo entre dos ondas se conoce como intervalo entre ondas y resulta de restar la latencia de la segunda onda menos la latencia de la primera. Por ejemplo:

Intervalo I-III = Latencia(Onda III) – Latencia(Onda I).

A fin de probar el método SPC propuesto, se ha desarrollado una herramienta de software que permite realizar todas las tareas descritas en este capítulo. En particular, permite:

- definir nuevos dominios (símbolos dependientes del dominio, con sus características, y la matriz de costes),
- cargar las series temporales numéricas en la base de datos,
- transformar las series temporales numéricas en secuencias temporales simbólicas, de acuerdo con las características particulares del dominio,
- visualizar gráficamente las semejanzas y diferencias entre las series temporales numéricas y las secuencias temporales independientes y dependientes del dominio,
- definir las clases de las secuencias temporales almacenadas,
- descubrir patrones en las secuencias temporales pertenecientes a la misma clase,
- definir la clase del grupo de control y encontrar los patrones exclusivos de las demás clases,
- clasificar nuevas secuencias simbólicas, y
- generar resultados estadísticos del proceso de clasificación para cada clase.

En lo que resta de este capítulo, se describirá el dominio de los PEATC en el que se ha aplicado el método, se detallará la experimentación realizada para probar tanto el módulo de descubrimiento de patrones como el de clasificación, así como encontrar la mejor configuración de parámetros para el mencionado dominio y, finalmente, se mostrarán los resultados obtenidos para la mejor configuración encontrada.

5.1.2. Estructuración de los datos

La información de dominio es almacenada en una base de datos relacional. La figura 5.3 muestra el modelo conceptual simplificado de la base de datos.

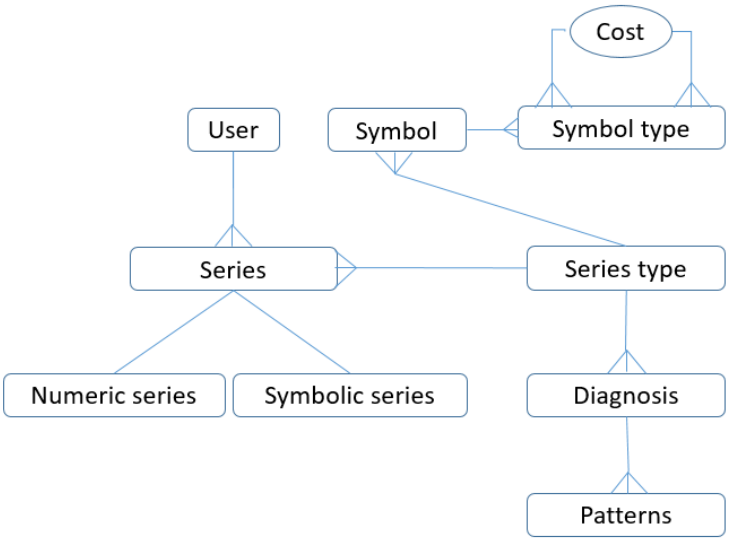

Figura 5.3. Estructura de la base de datos relacional.

Inicialmente, con ayuda de la herramienta de software desarrollada, las tablas de la base de datos deben ser rellenadas con los datos relativos a los símbolos, tipos de símbolos y costes entre los símbolos tipados, que se utilizarán para el cálculo de la distancia entre dos subsecuencias.

En el caso de estudio de los PEATC, el conocimiento del dominio fue recogido, fundamentalmente, a través de entrevistas con el experto, pero también analizando el conocimiento público. Durante las entrevistas, el experto identificó las partes relevantes de un conjunto de series temporales de PEATC (símbolos dependientes del dominio, correspondientes a las ondas e intervalos entre ondas). Un prototipo fue implementado, y las secuencias simbólicas resultantes fueron mostradas al experto para evaluación, a partir de la cual se realizaron algunos cambios en el prototipo. Este proceso se realizó iterativamente hasta cuando el prototipo retornó los símbolos y sus tipos correctos, en un alto porcentaje de casos (98%). Se necesitaron entre tres y cinco iteraciones, dependiendo de la dificultad de identificación de cada símbolo. Los resultados de las últimas iteraciones fueron mostrados también a otros neurofisiólogos para asegurar una

perspectiva más generalista y que no fuera tendenciosa. La tabla 5.1 muestra los símbolos y sus tipos para el dominio PEATC.

Símbolos dependientes del dominio	Tipos de símbolo
Onda I	Retardada Normal Adelantada
Onda III	Retardada Normal Adelantada
Onda V	Retardada Normal Adelantada
Intervalo interondas I-III	Corto Normal Largo
Intervalo interondas III-V	Corto Normal Largo
Intervalo interondas I-V	Corto Normal Largo

Tabla 5.1. Símbolos dependientes del dominio y sus tipos para el dominio PEATC

5.1.3. Selección de datos

Este estudio de los datos de PEATC se realizó en colaboración con el Hospital Homero Castanier Crespo (Ecuador). Los datos fueron recogidos de dos grupos de individuos.

- El primer grupo estuvo compuesto por estudiantes de una institución de educación especial, de edades entre 8 y 14 años. Todos estos estudiantes tienen algún tipo de discapacidad física o intelectual y pertenecen al grupo con riesgo de sufrir algún problema neurosensorial, muy probablemente auditivo. Estas pruebas se realizaron específicamente para esta investigación. Se seleccionaron 36 pruebas PEATC válidas; 22 de las cuales correspondieron a pacientes sanos y 14 a pacientes con algún tipo de afección (ver tabla 5.2).
- El segundo grupo estuvo compuesto por pacientes regulares del hospital, en el mismo grupo de edad, de los cuales 19 eran sanos y 28 tenían algún tipo de afección. Como cabía esperar, el porcentaje de pacientes sanos fue inferior en los datos de los pacientes del hospital. Se tuvo especial cuidado de seleccionar solamente pruebas de PEATC de pacientes con problemas de audición que exhibían formas de onda distinguibles a 70 dB, que pudieran ser analizadas.

Los pacientes del primer grupo fueron analizados y diagnosticados. De todos los casos cuyos diagnósticos tenían que ver con problemas auditivos, se centró la atención en tres diagnósticos

(clases): pérdida auditiva conductiva, schwannoma vestibular con implicación del tronco cerebral y schwannoma vestibular con implicación del 8º nervio. Además de los casos patológicos encontrados para cada uno de los diagnósticos anteriores, también se obtuvo una cantidad de casos de pacientes sanos que conformara el grupo de control.

Después del proceso de limpieza de los datos, se tuvieron un total de 83 casos de pacientes, incluyendo aquellos que fueron examinados exclusivamente para este estudio y que se corresponden con población de riesgo.

Es importante tener en cuenta que este es un dominio muy especializado, donde las pruebas son costosas y sólo los datos de las pruebas efectuadas siguiendo estrictamente el mismo protocolo médico y en las mismas condiciones deben ser seleccionados. Los datos están disponibles en https://app.box.com/Auditory-Evoked-Potential-Data (Accessed: 28 March 2016).

Todas las series temporales fueron etiquetadas con su respectiva clase y todos los datos etiquetados fueron almacenados en la base de datos como series temporales numéricas que, a su vez, fueron convertidas posteriormente en secuencias temporales simbólicas usando el módulo de transformación simbólica implementado (explicado en la sección 4.1). La tabla 5.2 resume la distribución de los datos por clases.

Clase	**Número de casos**		
	Institución de educación especial	Datos históricos del hospital	**Total**
Sanos	22	19	**41**
Pérdida auditiva conductiva	8	15	**23**
Schwannoma vestibular – TC involucrado	2	7	**9**
Schwannoma vestibular – 8º nervio involucrado	4	6	**10**
Total	36	47	**83**

Tabla 5.2. Casos divididos por clases.

5.1.4. Análisis inicial de los datos

Para ilustrar las cuestiones que el clasificador tiene que tener en cuenta, la figura 5.4 muestra ejemplos de formas de onda de PEATC para los tres posibles diagnósticos y el grupo de control. Las bandas resaltadas representan los intervalos de normalidad para las latencias de las ondas I, III y V.

Existen diferencias visibles en el tamaño y duración de las tres formas de onda principales de los PEATC (etiquetadas I, III y V en la Figura 5.4) para las tres afecciones. Adicionalmente, se encontró que las tres ondas se retardan con respecto al rango de audición normal en la pérdida

auditiva conductiva. Por otro lado, la onda I es normal y las ondas III y V son retardadas en los dos casos de schwannoma vestibular. Con respecto a los intervalos entre ondas (que se muestran en la Figura 5.4 etiquetados como IWI por sus siglas en inglés, *Interwave Interval*), se encontró que la duración de los tres intervalos era normal para los pacientes sanos, así como para aquellos con pérdida auditiva conductiva. Por otro lado, los intervalos I-III y I-V son largos para los dos casos de schwannoma vestibular, mientras que el intervalo III-V es largo para el schwannoma con implicación del tronco cerebral y normal para el schwannoma con implicación del 8º nervio.

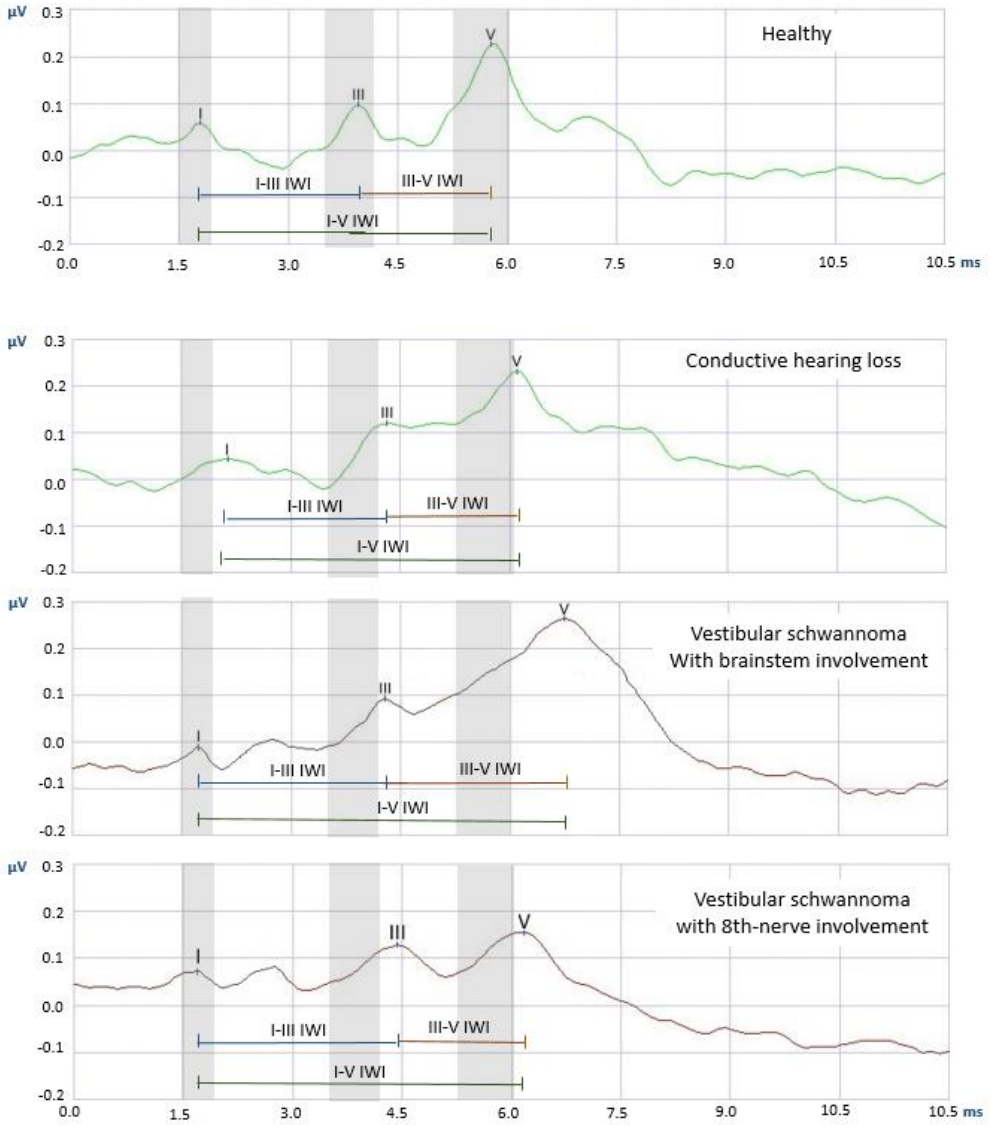

Figura 5.4. Comparación de las formas de onda de todas las clases.

5.2. Experimentos preliminares

Se diseñaron algunos experimentos preliminares con dos objetivos: primero, se desea mostrar que una abstracción temporal que incluye conocimiento dependiente del dominio puede ser más beneficioso que el uso de métodos numéricos en algunos dominios. Para hacer esto, se ejecutó un

sencillo experimento usando las transformadas de Fourier y el algoritmo de *clustering* X-means. En segundo lugar, se quiere comprobar que la técnica de descubrimiento de patrones encuentra patrones relevantes para cada clase (válidos para discriminar entre las clases), así como buscar la mejor configuración de los parámetros de la técnica de descubrimiento de patrones con el fin de lograr los mejores resultados posibles en la clasificación.

5.2.1. Sobre la utilidad de los métodos de extracción temporal

Los conceptos más relevantes para diagnóstico en el dominio BAEP son las tres ondas (wave I, wave III, wave V) y los intervalos entre ellas (IWI I-III, IWI I-V) que se han explicado en la sección 5.1.1. Otras ondas o picos que ocurren en la serie temporal no son relevantes para el proceso de toma de decisiones. Sobre esta base, parece razonable pensar que los métodos numéricos no tendrán un comportamiento idóneo a la hora de comparar o clasificar las series temporales en este dominio. Para comprobarlo, se utilizó la transformada discreta de Fourier (DFT) para transformar las series temporales en vectores de coeficientes, compararlos utilizando la distancia euclidiana y agruparlos utilizando un método de agrupación. Este es, probablemente, el procedimiento más utilizado para el análisis de series temporales mediante técnicas numéricas. Se seleccionaron 43 series temporales divididas en cuatro clases de tamaños similares (es decir, las clases están balanceadas):

- Sanos (12 casos)
- Pérdida auditiva conductiva (12)
- Schwannoma con implicación del tronco cerebral (9)
- Schwannoma con implicación del 8º nervio (10).

El algoritmo de agrupamiento X-means, descrito en [Pelleg, Moore 2000], fue aplicado al conjunto de series transformadas en vectores de coeficientes, usando la transformada discreta de Fourier DFT, y se obtuvieron cuatro *clusters* de salida. El resultado se muestra en la Figura 5.5; en ella, cada punto representa una serie temporal y su color corresponde a la clase a la cual pertenece, tal como se indica en la leyenda. La intención de este experimento es la de verificar si el algoritmo de agrupamiento consigue dividir a los individuos en las cuatro clases originales.

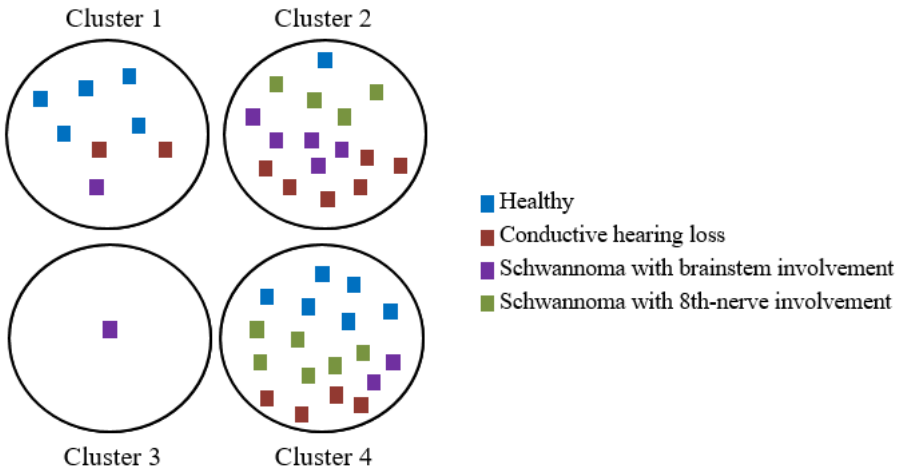

Figura 5.5. Distribución de las instancias agrupadas por X-means.

Al analizar los resultados de este experimento, se puede concluir que ni la distribución resultante de los grupos ni la asignación de los individuos a ellos se asemeja a las clases a las que las series temporales realmente pertenecen. Ninguno de los grupos contiene una mayoría de individuos de una sola clase, y, con la excepción trivial del grupo 3, no hay ningún grupo que agrupe a una sola clase. La razón de estos resultados tan pobres es que el uso conjunto de la DFT y la distancia euclidiana se centra en el valor absoluto de las series temporales en cada punto y no en la forma global de las series. Además, el algoritmo considera no sólo las ondas relevantes sino todas las ondas de las series, con lo que las ondas relevantes pasan a tener una menor importancia relativa. A pesar de que se trata de un experimento simple, los resultados son tan pobres como para descartar el uso de este tipo de métodos numéricos en el dominio que nos ocupa.

5.2.2. Parametrización de la técnica de descubrimiento de patrones

El método de descubrimiento de patrones propuesto tiene un conjunto de parámetros que pueden ser ajustados de acuerdo con las características del dominio para mejorar su rendimiento. Se han llevado a cabo algunos experimentos preliminares para conseguir la mejor configuración de parámetros para el dominio PEATC.

Los parámetros que definen el comportamiento de la técnica de descubrimiento de patrones son minsupport y maxdist. Para encontrar la mejor configuración para estos dos parámetros, se aplicó el método a la tarea de clasificación más simple, considerando sólo dos clases: una clase para los pacientes no saludables y el otro para el grupo de control (pacientes sanos). En concreto, se usaron todos los casos (un total de 83) para el experimento. De esos casos, 42 corresponden a individuos no saludables (agrupando todos los casos de enfermedad en una sola clase) y 41 a individuos sanos.

Cinco configuraciones diferentes de la técnica de descubrimiento de patrones se aplicaron a los datos variando los parámetros minsupport y maxdist. Para cada configuración se aplicó una validación cruzada, como sigue:

1. Cargar esta configuración de parámetros (*minsupport*, *maxdist*).
2. Usando un proceso de validación cruzada.
   2.1. Aplicar la técnica de descubrimiento de patrones para las clases seleccionadas: Clase positiva "no saludable" y clase negativa "saludable".
   2.2. Clasificar los datos según los patrones exclusivos identificados.
   2.3. Calcular las medidas de rendimiento que se utilizarán para evaluar los resultados.

La tabla 5.3 resume los resultados de este experimento para cada una de las cinco configuraciones, que se han calculado utilizando las medidas de rendimiento adecuadas para la clasificación binaria en dominios médicos [Cios, Moore 2002], [Sokolova, Lapalme 2009]. Como se desprende de la tabla 5.3, el rendimiento del clasificador mejora con valores altos de minsupport y valores bajos de maxdist. Por lo tanto, la mejor configuración de parámetros es minsupport = 0,95 y maxdist = 0. Esto indica que la configuración más confiable en nuestro caso de estudio, busca patrones idénticos y no sólo similares. Es importante tener en cuenta que esto se refieres a las secuencias simbólicas, en las que ya existe un mayor nivel de tolerancia en relación a las series numéricas debido al procedimiento de transformación a símbolos.

Configuración de parámetros	tp	fp	fn	tn	Exactitud (Accuracy)	Sensitividad (Recall)	Especificidad (Specificity)	F_1 score
minsupport = 0.90 maxdist = 0.20	32	12	10	29	0.735	0.762	0.707	0.744
minsupport = 0.90 maxdist = 0.00	37	6	5	35	0.867	0.881	0.854	0.870
minsupport = 0.95 maxdist = 0.10	34	11	8	30	0.771	0.810	0.732	0.782
minsupport = 0.95 maxdist = 0.05	38	6	4	35	0.880	0.905	0.854	0.884
minsupport = 0.95 maxdist = 0.00	42	2	0	39	0.976	1.000	0.951	0.977

Tabla 5.3. Rendimiento del clasificador para las cinco configuraciones

Del presente experimento se pueden sacar, también, conclusiones positivas con respecto a la eficacia y la robustez de la técnica de descubrimiento de patrones debido a que los resultados de la clasificación, para cualquiera de las configuraciones utilizadas, están consistentemente por encima del 70% de exactitud. Como las clases en este experimento están balanceadas, la exactitud es una buena medida de rendimiento global del clasificador.

Un tercer parámetro que se utiliza en el proceso de clasificación es el porcentaje de los patrones

exclusivos de una clase que un individuo debe contener a fin de ser clasificado dentro de esa clase. En el dominio de PEATC, cada diagnóstico utilizado en los experimentos se asocia con un número relativamente alto de patrones. Esto plantea la cuestión de si un individuo puede asignarse a una clase incluso si no contiene todos y cada uno de los patrones exclusivos de esa clase. Para abordar esta cuestión se decidió utilizar un parámetro, llamado ppc (porcentaje de patrones para la clasificación), mediante el que se indica el porcentaje de patrones de una clase que necesitan ser identificados en un individuo para que pueda ser considerado miembro de esa clase.

Se ha diseñado un experimento considerando las cuatro clases y utilizando todos los datos disponibles para las cuatro clases con el fin de determinar empíricamente el mejor valor para este parámetro. Se realizaron varias ejecuciones variando este parámetro dentro del rango de 20% a 100% con el fin de determinar los valores que proporcionan los mejores resultados para la clasificación. Mediante este experimento se encontró que hay un límite inferior para cada clase, por debajo del cual el rendimiento del clasificador ya no mejora. Por debajo de esa cota inferior, aumenta el número de falsos positivos. La tabla 5.4 muestra el intervalo de funcionamiento óptimo para el parámetro ppc por clase; a partir de estos intervalos, se encuentra que el clasificador proporciona los mismos resultados globales para los valores de ppc entre el 70% y el 100%. A la vista de esto, se ha optado por el valor más restrictivo en este rango (ppc = 100). Esta decisión está justificada por el hecho de que, por lo general, si se usara un valor menos restrictivo el número de falsos positivos (fp) tendería a aumentar y el número de verdaderos negativos (tn) tendería a disminuir; y en tal caso, la tasa de sensibilidad quedaría sin cambios, pero la tasa de especificidad disminuiría y este comportamiento no es deseable en dominios médicos.

Clase	**Rango óptimo para *ppc***
Sanos	50 a 100%
Pérdida auditiva conductiva	40 a 100%
Schwannoma vestibular – TC involucrado	60 a 100%
Schwannoma vestibular – 8º nervio involucrado	70 a 100%

Tabla 5.4. Rangos óptimos para el parámetro ppc.

CAPÍTULO 6. RESULTADOS Y DISCUSIÓN

Los experimentos preliminares proporcionaron la información requerida para determinar la mejor configuración de parámetros para la aplicación del método SPC en el dominio PEATC. Los datos disponibles de los pacientes consisten en 83 series temporales pertenecientes a cuatro clases:

- 41 corresponden a casos de pacientes sanos, usados como grupo de control,
- 23 corresponden a casos de pérdida auditiva conductiva,
- 9 corresponden a casos de schwannoma vestibular con implicación del 8º-nervio,
- 10 corresponden a los casos de schwannoma vestibular con implicación del tronco cerebral.

Debido al tamaño relativamente pequeño de dos de estas clases, una validación cruzada de tres particiones fue realizada para fines de clasificación. La tabla 6.1 muestra las matrices de confusión para las tres ejecuciones del proceso de validación cruzada. Las filas de las matrices de confusión muestran el número de casos que efectivamente pertenecen a las clases (c1 a c4), y las columnas muestran la clasificación realizada por el sistema (la columna u muestra los casos que el sistema no logra clasificar). Las celdas de la diagonal principal contienen los casos de clasificación correctos, mientras que las otras celdas contienen los casos clasificados erróneamente.

Ejecución 1		Clase pronosticada					Total
		c1	c2	c3	c4	u	
Clase real	c1	13	0	0	0	1	14
	c2	0	7	0	0	0	7
	c3	0	0	3	0	0	3
	c4	0	0	0	4	0	4
Total		13	7	3	4	1	28

Ejecución 2		Clase pronosticada					Total
		c1	c2	c3	c4	u	
Clase real	c1	13	0	0	0	1	14
	c2	0	8	0	0	0	8
	c3	0	0	3	0	0	3
	c4	0	0	0	3	0	3
Total		13	8	3	3	1	28

Ejecución 3		Clase pronosticada					Total
		c1	c2	c3	c4	u	
Clase real	c1	13	0	0	0	1	13
	c2	0	8	0	0	0	8
	c3	0	0	3	0	0	3
	c4	0	0	0	3	0	3
Total		13	8	3	3	1	27

Leyenda:
c1 Sano
c2 Pérdida auditiva conductiva
c3 Schwannoma vestibular – involucra 8º n
c4 Schwannoma vestibular – involucra TC
u No clasificado

Tabla 6.1. Matrices de confusión para las ejecuciones con validación cruzada de tres particiones

Puesto que se trata de un problema de clasificación de múltiples clases, se han usado las medidas de rendimiento apropiadas para calcular el rendimiento del clasificador. La tabla 6.2 describe esas medidas, las cuales han sido adaptadas de [Sokolova, Lapalme 2009]. Hay dos tipos

de medida: el micro-promedio (denotado por µ) el cual es un promedio ponderado que toma en cuenta el tamaño de cada clase, y el macro promedio (denotado por M) que calcula un promedio asignando el mismo peso a todas las clases.

Medida	Fórmula	Enfoque de evaluación
Exactitud promedio (AA)	$\frac{\sum_{i=1}^{n} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{n}$	Exactitud promedio por clase.
Sensitividad$_M$ (SE_M)	$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fn_i}}{n}$	Sensitividad promedio por cada clase. Asocia igual peso a cada clase.
Sensitividad$_\mu$ (SE_μ)	$\frac{\sum_{i=1}^{n} tp_i}{\sum_{i=1}^{n} tp_i + fn_i}$	Similar a Sensitividad$_M$ pero calculando el peso promedio de acuerdo al tamaño de cada clase, mientras que asocia igual peso a cada clasificación individual.
Especificidad$_M$ (SP_M)	$\frac{\sum_{i=1}^{n} \frac{tn_i}{tn_i + fp_i}}{n}$	Especificidad Promedio de cada clase. Asocia igual peso a cada clase.
Especificidad$_\mu$ (SP_μ)	$\frac{\sum_{i=1}^{n} tn_i}{\sum_{i=1}^{n} tn_i + fp_i}$	Similar a Especificidad$_M$ pero calcula el peso promedio conforme el tamaño de cada clase.

Tabla 6.2. Medidas de rendimiento para la clasificación con múltiples clases

	Clase	tp	fp	fn	tn	AA	SPµ	SP_M	SEµ	SE_M
Ejecución 1	c1	13	0	1	14	0.991	1.000	1.000	0.964	0.982
	c2	7	0	0	20					
	c3	3	0	0	24					
	c4	4	0	0	23					
Ejecución 2	c1	13	0	1	14	0.991	1.000	1.000	0.964	0.982
	c2	8	0	0	19					
	c3	3	0	0	24					
	c4	3	0	0	24					
Ejecución 3	c1	13	0	0	14	1.000	1.000	1.000	1.000	1.000
	c2	8	0	0	19					
	c3	3	0	0	24					
	c4	3	0	0	24					
Total	c1	39	0	2	42	**0.994**	**1.000**	**1.000**	**0.976**	**0.988**
	c2	23	0	0	58					
	c3	9	0	0	72					
	c4	10	0	0	71					

AA: Exactitud promedio
SPµ: Micro especificidad
SP_M: Macro especificidad
SEµ: Micro sensibilidad
SE_M: Macro sensibilidad

c1: Sanos
c2: Pérdida auditiva conductiva
c3: Schwannoma vestibular implicación del 8º nervio
c4: Schwannoma vestibular implicación del tallo cerebral.

Tabla 6.3. Rendimiento del método de clasificación en cada ejecución

Los resultados mostrados en la tabla 6.3 son muy prometedores. Una sensibilidad global de 0,97 y una especificidad de 1 (es decir, no hay falsos positivos) son muy buenos indicadores en el ámbito médico. El buen comportamiento tanto de la técnica de descubrimiento de patrones como de la de clasificación, así como también del proceso de transformación simbólica, han propiciado estos resultados. Así mismo, el uso del conocimiento experto del dominio, para la

transformación de series temporales numéricas a simbólicas, es un elemento crucial del modelo para el logro de tales resultados.

Esta investigación se llevó a cabo en pacientes jóvenes de edades comprendidas entre los 8 y los 14 años. Es de esperar que una persona joven con audición cercana a la normal tenga un PEATC con ondas distinguibles. Podrían existir problemas con relación a la generalización de este estudio a otros rangos de edades, pues muchos adultos por encima de cierta edad pueden sufrir algún tipo de trastorno de pérdida auditiva neurosensorial. Hay que tener en cuenta, sin embargo, que la mayoría de los pacientes de nuestro estudio (incluso alguno de los pacientes catalogados como sanos) tienen diferentes grados de pérdida auditiva.

Una vez más, los trastornos de la audición suelen ser más graves en los adultos. Por lo tanto, la atenuación de las ondas PEATC se hacen cada vez más patentes y las latencias de las ondas se retrasan aún más a medida que avanza la edad. En este punto, vale la pena probar una solución con el fin de generalizar el estudio a adultos. Consiste en aumentar el nivel de estímulo a 90 o 100 dB en este tipo de pacientes con el fin de hacer que las ondas I, III y V sean más visibles puesto que, al aumentar la intensidad del estímulo sonoro, las respuestas electrofisiológicas neurosensoriales tienen mayor amplitud. El incremento de la intensidad del estímulo trae consigo la necesidad de que se redefinan los símbolos dependientes del dominio, razón por la cual el estudio para la generalización de la edad es motivo de contribuciones futuras en el ámbito del diagnóstico mediante el uso de potenciales evocados auditivos. Una de las ideas potencialmente interesantes para este propósito sería incluir la edad como un parámetro, con el fin de automatizar el análisis de la prueba PEATC independientemente de la edad.

Otro asunto clave tiene que ver con los datos corruptos por ruido. El ruido aparece en las series temporales PEATC en forma de picos y valles insignificantes. El método aquí propuesto maneja el ruido de dos maneras. En primer lugar, el método es capaz de ignorar estos símbolos ruidosos irrelevantes en la transformación de símbolos independientes del dominio a símbolos dependientes del dominio. En segundo lugar, el método de descubrimiento de patrones busca patrones similares. Esto significa que dos subsecuencias pueden ser consideradas iguales incluso si hay un cierto nivel de ruido. Sin embargo, hemos encontrado, a partir de la aplicación de este método para el dominio PEATC, que el método se comporta mejor cuando busca patrones idénticos, como se discute en la Sección 5.2.2. La razón en el caso particular del dominio PEATC es que el proceso de recogida de datos reduce el ruido mediante la repetición del estímulo acústico alrededor de 2000 veces y al final recoge la respuesta promedio.

CAPÍTULO 7. CONCLUSIONES

Se ha desarrollado un método de clasificación de series temporales (SPC), basado en el descubrimiento de patrones en conjuntos de secuencias simbólicas. El método cuenta con tres procesos bien definidos y cada uno cumple con un gran objetivo:

- El proceso de transformación, en dos niveles, de las series temporales numéricas en secuencias simbólicas recoge los conceptos relevantes del domino y, como efecto paralelo, reduce la dimensionalidad de las series numéricas, lo que facilita, de forma efectiva, el análisis del conjunto de series temporales.

- El proceso de descubrimiento de patrones se realiza mediante el uso de un algoritmo, también contribución de la presente tesis, que optimiza la búsqueda mediante una intensiva aplicación de la propiedad a a priori, y funciona correcta y eficazmente en la consecución de los patrones frecuentes existentes en el conjunto de secuencias temporales.

- El proceso de clasificación realiza el trabajo mediante la búsqueda de los patrones asociados a los diferentes síntomas, dentro de las secuencias simbólicas a ser clasificadas.

Se ha aplicado este método al dominio PEATC. Los resultados muestran que el método es capaz de encontrar patrones en series temporales de PEATC de latencia corta y es muy preciso para predecir correctamente si un paciente tiene un trastorno relacionado con la audición. Estos resultados sugieren que el método de data mining propuesto es apropiado para ese dominio.

El método tiene algunos parámetros (*minsupport*, *maxdist* y *ppc*) que pueden ser ajustados para lograr una configuración óptima para cualquier dominio en particular. Se ha presentado un procedimiento simple para calibrar estos parámetros a fin de conseguir los mejores resultados posibles.

Se han probado cinco configuraciones distintas de estos parámetros para estudiar el comportamiento del clasificador. De acuerdo con los resultados, el rendimiento del clasificador alcanza su máximo valor cuando *minsupport* = 0.95 y *maxdist* = 0.0 (con *ppc*=1), esto es, para un soporte mínimo alto y tolerancia cero para la distancia entre subsecuencias simbólicas. Es importante resaltar que la definición de los símbolos admite un grado tolerable de variación en los valores de éstos, y con ello el método tolera un nivel apropiado de flexibilidad.

El método propuesto es efectivo en cuanto a reducción de la dimensionalidad. Si la transformación simbólica incluye el conocimiento correcto del dominio, podría decirse que el método produce una representación de los datos que refleja claramente los conceptos relevantes

del dominio.

El método propuesto ha necesitado conocimiento público o experto, que puede ser visto como una debilidad; sin embargo, para lo único que el método depende del dominio es para la de definición de símbolos en la fase de transformación simbólica. Este paso puede realizarse con la ayuda de una interfaz gráfica. El médico especialista que participó en esta investigación encontró la interfaz fácil de usar durante el caso de estudio de los PEATC. Como demuestran los resultados, el conocimiento dependiente del dominio mejora sustancialmente la efectividad del clasificador; este también hace que la herramienta sea más aceptable para los médicos, gracias a que los resultados pueden ser explicados usando los mismos conceptos que el médico usa con regularidad. En otras palabras, el método no se comporta como una caja negra, produciendo diagnósticos que el médico podría encontrar difíciles de aceptar; por el contrario, produce una explicación de los diagnósticos resultantes en el lenguaje propio del médico, como por ejemplo: “el trastorno del paciente x puede corresponder con un schwannoma vestibular con implicación del tronco cerebral, pues las ondas III y V están retardadas y el intervalo III-V es largo”.

El trabajo desarrollado en la presente tesis, es base suficiente para el desarrollo de un sistema computacional de apoyo a la decisión que mostraría el diagnóstico junto con una explicación en términos médicos, enfatizando los puntos clave que guiarán al especialista hacia el diagnóstico. Este sistema computacional sería capaz de desplegar los resultados de forma gráfica, resaltando las partes principales de la onda, los desplazamientos, etc., adicionalmente, cada elemento visual estaría acompañado por su respectiva explicación textual.

El método propuesto es, potencialmente, muy útil tanto para el médico experto como para el principiante, debido a que reduce su carga de trabajo con el cálculo automático de la información derivada de los datos de entrada que son relevantes en el dominio y también aportando soporte al proceso de toma de decisiones en lo referente a tareas de diagnóstico en dominios tales como PEATC o cualquier otro dominio con series temporales como datos de entrada.

Los métodos numéricos no son apropiados para muchos dominios donde los conceptos relevantes están directamente relacionados con la forma de las series temporales antes que con los valores instantáneos absolutos. Los PEATC son buenos ejemplos de tales dominios, donde los conceptos importantes son el retraso de picos específicos, así como el intervalo de tiempo entre los picos respectivos. Este es solo un ejemplo de cómo cualquier conocimiento del dominio puede ser capturado con la técnica propuesta de transformación a símbolos dependientes del dominio. Son precisamente esas características del dominio las que vuelven ineficientes a los métodos independientes del dominio, puesto que es casi imposible recoger esta información usando tales

métodos. Esto se ha demostrado en esta investigación aplicando técnicas muy conocidas como DFT y Xmeans.

CAPÍTULO 8. LÍNEAS FUTURAS DE INVESTIGACIÓN

Algunas de las posibles líneas futuras de investigación que han surgido en el desarrollo de esta tesis se detallan a continuación, desglosándolas en líneas relacionadas con el método propuesto, con la herramienta desarrollada y también con la aplicación al dominio de los ABR.

Líneas de Investigación relacionadas con el modelo propuesto

Los símbolos, tanto independientes como dependientes del dominio, que han sido usados en la presente tesis pueden tener limitaciones para el análisis de ciertos problemas asociados con otros dominios en los que fuera aplicable el método. Por ello, es necesario que se trabaje en la creación de una amplia base de símbolos que podrían estar disponibles a manera de colecciones de clases para su uso de acuerdo con las necesidades puntuales de un problema asociado a un determinado dominio.

Como una línea futura de investigación, se planifica la introducción de la distancia de edición para calcular la distancia entre secuencias simbólicas. Esto puede dotar al modelo de una mejor manera de medir la distancia entre secuencias simbólicas. Un inconveniente es, sin embargo, que la propiedad Apriori ya no se mantendría en todos los casos.

Sería importante investigar nuevos dominios para probar el modelo y demostrar su efectividad, mediante experimentos exploratorios que podrían consistir en transformar las series temporales numéricas en conjuntos de secuencias temporales simbólicas e interactuar con el experto en el dominio para conocer si esas secuencias son más fáciles de analizar que las series originales. Este podría ser un punto de partida para iniciar un estudio detallado del nuevo dominio.

Se considera importante encontrar algoritmos que mejoren el rendimiento del método. Se propone como una línea futura de investigación la búsqueda de nuevas alternativas para el descubrimiento de patrones en secuencias temporales simbólicas, para compararlas con el del método propuesto que integra ideas del cubo de Han y el algoritmo Apriori.

El método no se aplica para streams en línea, esto restringe los posibles dominios de aplicación y puede considerarse una desventaja. Sería pues importante que se enriqueciera el modelo para que soporte series temporales adquiridas en streams en línea, manteniendo siempre la misma funcionalidad que fuera de línea.

Los algoritmos han sido diseñados sobre la base de estructuras de datos en la memoria principal del ordenador. Una interesante línea futura de investigación es la de proveer al modelo de la posibilidad de procesar series temporales de tamaños muy grandes, lo cual se puede lograr incluyendo estructuras de almacenamiento para las series temporales, localizadas en la memoria secundaria del ordenador.

Líneas de Investigación relacionadas con la aplicación desarrollada

Para la aplicación del método, se desarrolló un software capaz de automatizar el proceso de diagnóstico a partir de series numéricas. Además, auxilia al experto en las tareas de definición de los símbolos dependientes del dominio. A pesar de que el software desarrollado fue usado por los expertos en el dominio de forma casi intuitiva, para optimizar el desempeño global de la solución sería interesante trabajar en el desarrollo de interfaces gráficas dotadas de inteligencia artificial que minimicen el trabajo del usuario a la hora de definir los parámetros para abordar el análisis de las series temporales. Una posible mejora a la interfaz podría consistir en la inclusión de una herramienta de dibujo para definir gráficamente los símbolos con sus dimensiones posibles, y que el software se encargara de almacenarlos en la tabla de símbolos, quedando así ya disponibles para ser utilizados.

Todas las series temporales fueron almacenadas en la base de datos con un tamaño único, pues ese era el tamaño de la forma de onda arrojada por el equipo usado en nuestra experimentación. Podría ocurrir, sin embargo, que los datos provengan de diferentes equipos, entre los que varíe la duración de la serie temporal. El software deberá ser capaz de discriminar el aspecto de duración y tomar decisiones para simular que tal diferencia no existe, al tiempo que simboliza las series y encuentra correctamente los patrones.

Una línea de investigación que ciertamente tendría un alto impacto en la correcta aplicación del método propuesto, tiene que ver con la posibilidad de dotar al sistema de una herramienta que convierta en líneas de código las recomendaciones del experto, a través de una interpretación que aplique técnicas de inteligencia artificial como la lógica difusa; esto redundaría en que el conocimiento experto estaría disponible prácticamente en línea.

Líneas de Investigación relacionadas con el dominio de aplicación

La experimentación que se presenta en el capítulo 5 de la presente tesis, se basa en los datos pertenecientes a pacientes de un grupo poblacional específico cuya edad oscila entre los ocho y los catorce años. Es importante pues, como un trabajo de investigación futuro, que los resultados experimentales sean generalizados a otros grupos de edad. Una de las ideas interesantes para este propósito sería incluir la edad del paciente como un parámetro para la obtención de los símbolos dependientes del dominio, con el fin de automatizar el análisis de la prueba PEATC independientemente de la edad.

También resulta importante que se generalicen los resultados experimentales para grupos de

pacientes con otros trastornos auditivos; particularmente interesante resulta incluir afecciones que involucren ambos oídos, lo que implica el análisis de al menos dos series por cada paciente, esto es, una por cada oído. El resultado entonces se obtendría después de un análisis comparativo de la serie temporal del oído izquierdo con la del derecho.

La experimentación realizada en este investigación se ha limitado a los potenciales evocados de latencia corta, esto es, hasta quince ms de duración. Hay un vasto camino por andar en el análisis de los potenciales evocados de latencia media y larga, donde se podría esperar que sea totalmente posible la aplicación del método propuesto en la presente tesis.

Un aspecto que merece ser investigado es el análisis de ABRs mediante el método de apilado, que consiste en obtener de un mismo paciente las formas de onda relativas a tres intensidades, que pueden ser 70 dB, 50 dB y 30 dB. Al apilarlas se trazan líneas que unen los picos de las ondas I, de las ondas III y de las ondas V para, de acuerdo con la inclinación de esas líneas, obtener un sobre el estado del sistema auditivo del paciente.

CAPÍTULO 9. BIBLIOGRAFÍA

[Aamodt, Plaza 1994] A. Aamodt and E. Plaza. *Case-Based Reasoning: Foundational Issues, Methological Variations, and System* Approaches. AI Communications, IOS Press, Vol 7, issue 1, pp. 39-59, 1994.

[Agbinya 1996] J.I. Agbinya. *Discrete Wavelet Transform Techniques in Speech Processing*. In IEEE TENCON - Digital Signal Processing Applications, pp. 514–519, 1996.

[Aggarwal, Aggarwal 2012] N. Aggarwal, and K. Aggarwal. *A Mid-Point Based K-means Clustering Algorithm for Data Mining*. In International Journal on Computer Science and Engineering, vol 4, issue 6, pp.1174-1180, 2012.

[Agrawal et al. 1993a] R. Agrawal, T. Imielinski and A. Swami, *Mining Association Rules between Sets of Items in Large Databases*. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Peter Buneman and Sushil Jajodia (Eds.), ACM, New York, NY, USA, vol. 22, issue 2, pp. 207–216, Jun. 1993.

[Agrawal et al. 1993b] R. Agrawal, C. Faloutsos and A. Swami, *Efficient similarity search in sequence databases*. In Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms, David B. Lomet (Ed.), Springer-Verlag London, London, UK, pp. 69–84, 1993.

[Agrawal, Srikant 1994] R. Agrawal and R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases.* In VLDB'94 Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 487-499, 1994.

[Agrawal et al. 1995] R. Agrawal, G. Psaila, E.L. Wimmers and M. Zaït, *Querying shapes of histories.* In Proceedings of the 21th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 502-514, 1995.

[Agrawal et al. 1996] R Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, *Fast Discovery of Association Rules*. In Book Advances in knowledge discovery and data mining, American Association for Artificial Intelligence, Menlo Park, CA, USA, pp. 307-328, 1996.

[Akansu, Haddad 2001] A.N. Akansu, and R.A. Haddad, *Multiresolution Signal Decomposition. Transforms, Subbands, and Wavelets*. Second Edition, Academic Press, San Diego, CA, USA, 516p, 2001.

[AL-Nabi, Ahmed 1013] D. AL-Nabi, and S. Ahmed, *Survey on Classification Algorithms for Data Mining-(Comparison and Evaluation)*. In Computer Engineering and Intelligent Systems (online), vol 4, issue 8, pp. 18-25, 2013.

[Alfares, Nazeeruddin 2002] H. Alfares and M. Nazeeruddin, *Electric Load Forecasting Literature Survey and Classifcation of Methods*. In International Journal of Systems Science, volume 33, issue 1, Taylor & Francis Ltd, pp. 23-34, 2002.

[Alonso et al. 2012] F. Alonso, L. Martinez, A. Perez and J. P. Valente, *Cooperation between Expert Knowledge and Data Mining Discovered Knowledge: Lesson Learned*. In Expert Systems with Applications, Elsevier, vol. 39, issue 8, pp. 7524–7535, Jun. 2012.

[Alonso, García 2012] A. Alonso, and C. García, *Time Series analysis - Time Series and Stochastic Processes*. Universidad Politécnica de Madrid, Universidad Carlos III, Madrid, España, 2012.

[Alvo et al. 2011] M. Alvo, E. Firuzan and A. Firuzan, *Predictability of Dow Jones Index via Chaotic Symbolic Dynamics*. In World Applied Sciences Journal, Vol 12, issue 6, pp. 835-839, 2011.

[An 2006] A. An. *Bayesian Networks*. In Encyclopedia of Data Warehousing and Mining, vol 1, Idea Group Reference, London, UK, pp. 144-149, 2006.

[Ao et al. 2015] X. Ao, P. Luo, C. Li, F. Zhuang and Q. He. *Online frequent episode mining*. In Proceeding of Data Engineering (ICDE), 2015 IEEE 31st International Conference, IEEE, pp 891-902, 2015.

[Apostolico, Guerra 1985] A. Apostolico, and C. Guerra. *The Longest Common Subsequence Problem Revisited*. In Computer Science Technical Reports, Department of Computer Science, Purdue e-Pubs, Purdue University, Paper 462, 41p, 1985.

[Armengol 2007] E. Armengol. *Usages of generalization in case-based reasoning*. In Proceedings of the International Conference on Case-Based Reasoning, Springer, Berlin Heidelberg, pp. 31-45, 2007.

[Ayad et al. 2001] A. Ayad, N. El-Makky and Y. Taha. *Incremental Mining of Constrained Association Rules*. In Proceedings of the SIAM International Conference on Data Mining, Chicago, IL, USA, pp. 1-18. 2001.

[Azevedo et al. 2012] J. Azevedo, R. Almeida, and P. Almeida, *Using Data Mining with Time Series Data in Short Term Stocks Prediction- A Literature Review*. International Journal of Intelligence Science, 2, Scientific Research, pp. 176-180, 2012.

[Babbu et al. 2011] B.H. Babu, N.S. Chandra, and T.V. Gopa, *Clustering AlgorithmsFor High Dimensional Data – A Survey Of Issues And Existing Approaches*. In the Special Issue of International Journal of Computer Science & Informatics (IJCSI), Vol 2, issue 1, 142-148, 2011.

[Barat et al. 2010] C. Barat, C. Ducottet, E. Fromont, A.C. Legrand and M. Sebban, *Weighted Symbols-based Edit Distance for String-Structured Image Classification*. In Proceedings of the European Conference ECML PKDD, Barcelona, Spain, Part I, pp. 72-86, 2010.

[Bashir et al. 2006] Ahmad Bashir, Latifur Khan, and Mamoun Awad, *Bayesian Networks*. In Encyclopedia of Data Warehousing and Mining, vol 1, Idea Group Reference, London, UK, pp. 89-93, 2006.

[Batal et al. 2009] I. Batal, L. Sacchi, R. Bellazzi and M. Hauskrecht, *Multivariate Time Series Classification with Temporal Abstractions*. In Proceedings of the Twenty-Second International FLAIRS Conference, Florida, FL, USA, pp. 344-349, 2009.

[Batal et al. 2011] I. Batal, H. Valizadegan, G.F. Cooper and M. Hauskrecht. *A Pattern Mining Approach for Classifying Multivariate Temporal Data*. In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, Atlanta, GE, USA, pp. 358-365, 2011.

[Batal et al. 2012] I. Batal, D. Fradkin and J. Harrison. *Mining Recent Temporal Patterns for Event Detection in Multivariate Time Series Data*. In KDD '12 Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, pp. 280-288, 2012.

[Bayardo 1998] J. Bayardo, *Efficiently Mining Long Patterns from Databases*. In Proceedings of the ACM-SIGMOD Internatinal Conference on Management of Data, ACM, pp. 85-93, 1998.

[Becker 2005] H. Becker. *A Survey of Correlation Clustering*. In COMS E6998: Advanced Topics in Computational Learning Theory, 10p, 2005.

[Berkhin 2006] P. Berkhin. *A Survey of Clustering Data Mining Techniques*. In Recent Advances in Clustering,Springer, Berlin, Germany, pp 25-71, 2006.

[Bertens, Siebes 2014] R. Bertens, and A. Siebes, *Characterising Seismic Data*. Technical Report, Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands, 2014.

[Boucheham 2012] B. Boucheham. *PLA Data Reduction for Speeding Up Time Series Comparison*. In International Arab Journal of Information Technology, Vol. 9, issue 5, 2012.

[Brakel, Krieg 2008] J. Brakel and S. Krieg. *Estimation of the Monthly Unemployment Rate Through Structural Time Series Modelling in a Rotating Panel Design*, discussion paper, Statistics Netherlands, Voorburg/Heerlen, 2008.

[Breiman et al. 1984] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, NY, USA, 354p, 1984.

[Brida 2000] J. Brida, *Symbolic Time Series Analysis and Economic Regimes*. Working Paper N° 2, Department of Economics IDEE, University of Siena, Milan, Italy, 2000.

[Brillinger 1996] D.R. Brillinger. *An Analysis of an Ordinal-Valued Time Series*. Athens Conference on Applied Probability and Time Series Analysis, Springer, New York, NY, USA, pp. 73-87, 1996.

[Brillinger 2000] D.R. Brillinger. Time Series: General. In International Encyclopedia of Social and Behavioral Sciences, Elsevier, 22p, 2000.

[Bringmann et al. 2009] B. Bringmann, S. Nijssen and A. Zimmermann, *Pattern-based Classification: A Unifying Perspective*. In Proceedings of 'From Local Patterns to Global Models', Second ECML PKDD Workshop, 2009.

[Burkard el al. 2007a] R. Burkard, M. Don and E. Joss, *Auditory Evoked Potentials: Basic Principles and Clinical Application*. Lippincott Williams and Wilkins, Baltimore, MD, USA, p. 731, 2007.

[Burkard et al. 2007b] R. Burkard, M. Don and J.J. Eggermont. *The Auditory Brainstem Response*. In Auditory Evoked Potentials. Basic Principles and Clinical Application, Lippincott Williams and Wilkins, Baltimore, 1st ed. ch. 11, sec. 1, pp. 229–230, 2007.

[Burkom et al. 2006] H.S. Burkom, S.P. Murphy and S. Galit. *Automated Time Series Forecasting for Biosurveillance*. Statistics in Medicine, Forthcoming; Robert H. Smith School Research Paper Nº RHS 06-035, available at SSRN: http://ssrn.com/abstract=923635, 26p, 2006.

[Canelas et al. 2012] A. Canelas, R. Neves, and R. Horta, *A New SAX-GA Methodology Applied to Investment estrategies optimization*. In GECCO'12 Proceedings of the Genetic and Evolutionary Computation Conference, ACM, pp. 1065-1072, 2012.

[Cassisi et al. 2012] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata and A. Pulvirenti, *Similarity Measures and Dimensionality Reduction Techniques for Time Series data*. In Advances in Data Mining Knowledge Discovery and Applications, Adem Karahoka (Ed.), InTech, Rijeka, Croatia, ch. 3, sec. 1, pp.71–96, 2012.

[Cazelles et al. 2008] B. Cazelles, M. Chavez, D. Berteaux, F. Menard, J. Vik, S. Jenouvrier, and N. Stenseth. *Wavelet analysis of ecological time series*. Oecologia Volume 156, Issue 2, Springer, pp. 287-304, 2008.

[Chakrabarti et al. 2001] K. Chakrabarti, E. Keogh, M. Pazzani and S. Mehrotra, *Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases*. In ACM Transactions on Database Systems, vol. 27, issue 2, pp. 188-228, 2002.

[Chan, Fu 1999] K.P. Chan and A. Fu, *Efficient time series matching by wavelets*. In Proceedings of the 15th International Conference on Data Engineering, Masaru Kitsuregawa, Leszek Maciaszek, Mike Papazoglou and Calton Pu (Eds.), IEEE Computer Society, Los Alamitos, CA, USA, pp. 126–133, 1999.

[Chaudhuri, Dayal 1997] S. Chaudhuri and U. Dayal, *An overview of data warehousing and OLAP technology*. SIGMOD Record, Special Section on Environment Information Systems, Michael Franklin (Ed.), ACM, New York, NY, USA, Vol. 26, issue 1, pp. 65–74, 1997.

[Chen, Pham 2001] G. Chen, and T.T. Pham, *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*, CRC Press, Boca Raton, FL, USA, 329p, 2001.

[Chen et al. 2005a] L. Chen, T. Ozsu and V. Oria, *Robust and Fast Similarity Search for Moving Object Trajectories*. In Transactions of the ACM SIGMOD 2005 Conference, Baltimore, Maryland, USA, p. 12, 2005.

[Chen et al. 2005b] Q. Chen, L. Chen, X. Lian, Y. Liu and J. X. Yu, *Indexable PLA for efficient similarity search*. In VLDB ’07: Proceedings of the 33rd international conference on Very large databases, pp. 435–446, 2007.

[Chen et al. 2006] M.S. Chen, J. Han and P.S. Yu, P. S. *Data mining: an overview from a database perspective*. IEEE Transactions on Knowledge and data Engineering, Vol 8, issue 6, pp. 866-883, 2006

[Cheung et al. 1996] Cheung, D. W., Han, J., Ng, V. T., & Wong, C. Y. *Maintenance of discovered association rules in large databases: An incremental updating technique*. In Proceedings of the Twelfth International Conference on Data Engineering. IEEE, pp. 106-114, 1996.

[Chu, Wong 1999] K. Chu, and and M. Wong. *Fast time-series searching with scaling and shifting*. In Proceedings of the 18th ACM Symposium on Principles of Database Systems. ACM, New York, NY, USA, 1999.

[Cios, Moore 2002] K.J. Cios and G.W. Moore, *Uniqueness of medical data mining*. Artificial Intelligence in Medicine, vol. 26, issue 1, pp. 1–24, Sep. 2002.

[Damerau 1964] F.J. Damerau. *A technique for computer detection and correction of spelling errors*. Communications of the ACM, Vol 7, issue 3, pp. 171-176, 1964.

[Das et al. 1997] G. Das, R. Fleischer, L. Gasieniec, D. Gunopulos, J. Kärkkäinen. *Episode matching*. In Proceedings of the 8th Annual Symposium, CPM 97 Aarhus, Denmark, pp. 12-27, 1997.

[Das et al. 2007] R. Das, D. Bhattacharyya1 and J. Kalita. *An Effective Dissimilarity Measure for Clustering Gene Expression Time Series Data*. In www.biotconf.org/2007/2007/Das.pd, Paper 56, 2007.

[David, Balakrishnan 2010] J. David, and K. Balakrishnan. *Significance of Classification Techniques in Prediction of Learning Disabilities*. In International Journal of Artificial Intelligence & Applications, Vol 1, issue 4, pp. 111-120, 2010.

[De Campos, Romero 2009] L. de Campos, and A. Romero. *Bayesian network models for hierarchical text classification from a thesaurus*. In the International Journal of Approximate Reasoning, Vol 50, Elsevier, pp. 932–944, 2009.

[De Jong et al. 1994] K. A. De Jong, W. M. Spears, and D. F. Gordon. *Using Genetic Algorithms for Concept Learning*. In book Genetic Algorithms for Machine Learning, Springer, New York, NY, USA, pp. 5-32, 1994.

[De Reus 1994] N.M. De Reus. *Assessment of benefits and drawbacks of using fuzzy logic, especially in fire control systems*. TNO report FEL-93-A158, TNO Defence Research, 1994.

[Deistler et al. 1986] M. Deistler, O. Prohaska, E. Reschenhofer, and R. Volmer. *Procedure for the Identification of Different Stages of EEG Background and its Application to the Detection of Drug effects, Electroencephalography and Clinical Neurophysiology*, pp. 294-300, 1986.

[Diggs, Povinelli 2003] D. Diggs, and R. Povinelli. *Temporal Pattern Approach for Predicting Weekly Financial Time Series*. In http://www.researchgate.net/publication/228969984_A_temporal_pattern_approach_for_predicting_weekly_financial_time_series, 2003.

[Dixit 2014] J. Dixit, and A. Choubey. *A Survey of Various Association Rule Mining Approaches.* In International Journal of Advanced Research in Computer Science and Software Engineering, Vol 4, issue 3, pp. 651-655, 2014.

[Dong, Li 1999] G. Dong, and J. Li. *Efficient Mining of Emerging Patterns: Discovering Trends and Differences*. In KDD'99 Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, pp. 43-52, 1999.

[Eckley 2001] I. Eckley. *Wavelet Methods for Time Series and Spatial Data*. PhD Thesis, Faculty of Science, University of Bristol, Bristol, UK, 2001.

[Elavarasi 2011] S.A. Elavarasi. *A Survey on Partition Clustering Algorithms*. In the International Journal of Enterprise Computing and Business Systems, Vol 1, issue 1, 14p. 2011.

[Fayyad et al. 1996] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. *From Data Mining to Knowledge Discovery in databases*. AI Magazine, American Association for Artificial Intelligence, pp. 37-54, 1996.

[Finney et al. 1998] C. Finney, J. Green, and C. Daw. *Symbolic Time Series Analysis of Engine Combustion Neasurements*. Paper of Society of Automotive Engineers, pp. 98062-98071, 1998.

[Fisher 1936] R.A. Fisher. *The Use of Multiple Measurements in Solution of Taxonomic Problems*. In Annals of Eugenics, Vol 7, issue 2, pp 179–188, 1936.

[Fuad, Marteau 2008a] M. Fuad and P.F. Marteau, *Extending the Edit Distance Using Frequencies of Common Characters*. In DEXA 2008 Database and Expert Systems Applications, Springer, Berlin, Germany, pp. 150–157, 2008.

[Fuad, Marteau 2008b] M. Fuad, and P.F. Marteau. *The Extended Edit Distance Metric", Content-Based Multimedia Indexing*. In CBMI'08 Contents Based Multimedia Indexing, London, UK, 2008.

[Gamero 2012] I. Gamero, *Pattern Recognition Based on Qualitative Representation of Signals. Application to Situation Assessment of Dynamical Systems*. PhD thesis, Universitat de Girona, Girona, Spain, 2012.

[Greene, Smith 1993] D. P. Greene, and S. F. Smith. *Competition-Based Induction of Decision Models from Examples*. In Journal Machine Learning, 13, pp. 229-257, 1993.

[Goebel, Gruenwal 1999] M. Goebel and L. Gruenwal. *A Survey of Data Mining and Knowledge Discovery Software Tools*. In ACM SIGKDD Exploration Newsletter, Vol 1, issue 1, pp. 20-33, 1999.

[Gortler 1995] S.J. Gortler. *Wavelet Methods for Computer Graphics*. PhD Thesis, Department of Computer Science, University of Princeton, 195p, 1995.

[Grira et al. 2005] N. Grira, M. Crucianu, and N. Boujemaa. *Unsupervised and Semi-supervised Clustering: a Brief Survey*. In A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence, 12p, 2005.

[Groves, Hannan 1968] G.W. Groves, and E.J. Hannan, *Time-Series Regression of Sea Level on Weather*. In Review of Geophysics, 6, pp. 129-174, 1968.

[Gudmunsson 1971] G. Gudmundsson, *Time-series Analysis of Imports, Exports and other Economic Variables*. In Journal of the Royal Statistical Society, Series A, issue 134, pp. 383-412, 1971.

[Gundogan et al. 2004] K. K. Gundogan, B. Alatas, and A. Karci. *Mining Classification Rules by Using Genetic Algorithms with Non-random Initial Population and Uniform Operator*. In Turkish Journal of Electronic Engineering, Vol 12, issue 1, pp. 43-52, 2004.

[Gunopulos, Das 2001] D. Gunopulos, and G. Das. *Time Series Similarity Measures and Time Series Indexing*. In SIGMOD'01 Proceedings of the 2001 ACM SIGMOD international conference on Management of data, ACM, New York, NY, USA, pp. 624, 2001.

[Hamming 1950] R.W. Hamming. *Error Detecting and Error Correcting Codes*. In The Bell System Technical Journal, vol 29, issue 2, American Telephone and Telegraph Company, USA, 1950.

[Han et al. 1998] J. Han, W. Gong and Y. Yin, *Mining Segment-Wise Periodic Patterns in Time Related databases.* In Proceedings of the International Conference on Knowledge Discovery and Data Mining, Rakesh Agrawal and Paul Stolorz (Eds.), AAAI Press, Menlo Park, CA, USA, pp. 214–218, Aug. 1998.

[Han et al. 2000] J. Han, J. Pei, and Y. Yin, *Mining Frequent Patterns without Candidate Generation*. In Proceedings of the ACM-SIGMOD International Conference on Management of Data SIGMOD Record, Weidong Chen, Jeffrey Naughton and Philip A. Bernstein (Eds.), ACM New York, NY, USA, Vol. 29 issue 2, pp. 1-12, Jun. 2000.

[Han et al. 2007] J. Han, H. Cheng, D. Xin and X. Yan, *Frequent pattern mining: current status and future directions*. Data Mining and Knowledge Discovery, Vol. 15, issue 1, pp 55–86, 2007.

[Han et al. 2012] J. Han, M. Kamber, and J. Pei. *Data Mining Concepts and Techniques*. 3rd Edition, Elsevier, Waltham, MA, USA, 517p, 2012.

[Hassleman et al. 1963] K. Hassleman, W. Munk, and G. MacDonald, *Bispectrum of Ocean Waves*. In Time Series Analysis, chapter 8, John Wiley and Sons, New York, NY, USA, 125-139, 1963.

[Ke, Wu 2010] C.K. Ke, M.Y. Wu. *Adaptive Support for Student Learning in an e-Portfolio Platform by Knowledge Discovery and Case-based Reasoning*. JSW, Vol 5, issue 12, pp. 1355-1362, 2010.

[Hidber 1998] C. Hidber. *Online Association Rules Mining*. In Technical Report TR-98-033, International Computer Science Institute, Berkeley, 1998.

[Hilderman, Hamilton 1999] R. Hilderman and H. Hamilton. *Knowledge Discovery and Interesting Measures: A Survey*. Technical Report, 27p. 1999.

[Hipp et al. 2000] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. *Algorithms for Association Rule Mining - a General Survey and Comparison*. In ACM SIGKDD Explorations Newsletter, Vol 2 issue 1, ACM New York, NY, USA, pp. 58-64, 2000.

[Homayounfard, Kennedy 2009] H. Homayounfard and P. Kennedy. *HDAX: Historical Symbolic Modelling of Delay Time Series in a communication network*. In Proceedings of the 8th Australasian Data Mining Conference, pp. 129-137, 2009.

[Homayounfard 2013] H. Homayounfard. *Packet-Loss Prediction Model Based on Historical Symbolic Time Series Forecasting*. PhD Thesis, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia, 207p, 2013.

[Hruschka et al. 2009] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, A.C.P.L.F. de Carvalho. *A Survey of Evolutionary Algorithms for Clustering*. In IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 2009.

[Jain, Dubes 1988] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Upper Saddle River, NJ, USA, 334p, 1988.

[Janikow 1993] C. Z. Janikow. *A knowledge-intensive genetic algorithm for supervised learning*. In Journal of Machine Learning, Vol 13, issue 2-3, pp. 189-228, 1993.

[Kameshwaran, Malarvizhi 2014] K. Kameshwaran, K. Malarvizhi. *Survey on Clustering Techniques in Data Mining*. In International Journal of Computer Science and Information Technologies, vol 5, issue 2, 2014.

[Kanth et al. 1998] K.V.R. Kanth, D. Agrawal, A. El Abbadi, and A. Singh. *Dimensionality Reduction for Similarity Searching in Dynamic Databases*. In Proceedings of the ACM SIGMOD international conference on Management of data, ACM, New York, NY, USA, pp. 166-176, 1998.

[Karczmacsuk 1998] J. Karczmarczuk. *Wavelets in Computer Graphics- a Tutorial for Ambitious Beginners*. DESS – Images, Université de Caen, 34p, 1998.

[Kaufman, Rousseeuw 1990] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, NY, USA, 355p, 1990.

[Keogh et al. 2000a] E. Keogh, K Chakrabarti, M. Pazzani, and S. Mehrotra, *Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases,* in KAIS - Long paper submited, 2000.

[Keogh et al. 2000b] E. Keogh, and M. Pazzani, *Scaling up Dynamic Time Warping for Datamining Applications*. In Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, ACM, Boston, MA USA, pp. 285-289, 2000.

[Keogh, Ratanamahatana 2002] E. Keogh, and C.A. Ratanamahatana, *Exact Indexing of Dynamic Time Warping*. In Proceedings of the 28th international conference on Very Large Data Bases, pp. 406-417, 2002.

[Keogh, Ratanamahatana 2004] E. Keogh, C. A. Ratanamahatana, *Exact Indexing of Dynamic Time Warping*. In Knowledge and Information Systems, Springer-Verlag London Ltd. 29p. 2004.

[Kim, Loh 2003] H. Kim, and W. Y. Loh. *Classification Trees with Bivariate Linear Discriminant Node Models*. In the Journal of Computational and Graphical Statistics, Vol 12, pp. 512–530, 2003.

[Kim et al. 2010] H. Kim, S. Kim, T. Weninger, J. Han, and T. Abdelzaher. *NDPMine: Efficiently Mining Discriminative Numerical Features for Pattern-Based Classification.* In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Barcelona, Spain, 16p, 2010.

[Koopmans 2012] M. Koopmans. *Time Series in Education: The Analysis of Daily Attendance in Two High Schools*. In Proceedings of the Annual Convention of the American Educational Research Association, New Orleans, LA, USA, 2012.

[Korn et al. 1997] F. Korn, H.V. Jagadish, and C. Faloutsos. *Efficiently Supporting Ad Hoc Queries in Large Datasets of time Sequences*. In Proceedings of the 1997 ACM SIGMOD international conference on Management of data, ACM, New York, NY, USA, pp. 289-300, 1997.

[Kumar, Kalia 2011] M. Kumar and A. Kalia, *A Comparative Study of Numeric and Symbolic Representation of stock Data.* International Journal of Computer Science and Technology, Vol. 2, issue 4, pp. 469-473, 2011.

[Kurtz 1996] S. Kurtz. *Approximate String Searching Under Weighted Edit Distance*. In Proceedings of the Third South American Workshop on String Processing, Recife, Brazil, Carlton University Press, 15p, 1996.

[Lartillot, Ayari 2006] O. Lartillot, and M. Ayari. *Motivic Pattern Extraction in Music, And Application to the study of the tunisian modal music*. ARIMA/SACJ Joint Special Issue — Advances in end-user data-mining techniques, Vol 36, pp. 16-28, 2006.

[Lavilles, Arcilla 2012] R.Q. Lavilles and M.J. Arcilla. *Enrollment Forecasting for School Management System. In International Journal of Modeling and Optimization*, Vol. 2, issue 5, 2012

[Laxman, Sastry 2006] S. Laxman and P. Sastry, *A Survey of Temporal Data Mining*. Sadhana, Vol 31, issue 2, pp. 173–198, 2006.

[Lee 1990] C. C. Lee, *Fuzzy Logic in Control Systems: Fuzzy Logic Controller -Part 1*. In IEEE Transactions on Systems, 1990

[Levenshtein 1966] V.I. Levenshtein. *Binary Codes Capable of Correcting Spurious Insertions and Deletions and Reversals*. In Soviet Phisics, Doklady Akademii Nauk, SSSR, Vol 163, issue 4, pp. 845-848, 1996.

[Lewis 2000] R. Lewis. *An Introduction to Classification and Regression Tree (CART) Analysis*. In Proceedings of the Annual Meeting of the Society for Academic Emergency Medicine ,San Francisco, CA, USA, 2000.

[Li, Ramamohanrarao 2000] J. Li, G. Dong, and K. Ramamohanrarao, *Making Use of the Most Expressive Jumping Emerging Patterns for Classification*, in Proceedings of the 2000 Pacific-Asia Conference on Knowledge Discovery and Data Mining, Kyoto, Japan, pp. 220–232, 2000.

[Li et al. 2001] W. Li, J. Han, and J. Pei, *CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules*, in Proceedings of the 2001 International Conference on Data Mining, San Jose, CA, USA, pp. 369–376, 2001.

[Liang, Lin 2002] H. Liang, and Z. Lin. *Multiresolution Decompositions and its Application to Electrogastric Signals*, In Recent Research in Novel Biomedical Engineering, Vol 1, Kansas City, KA, USA, pp. 15-31, 2002.

[Lin et al. 2002] J. Lin, E. Keogh, S. Lonardi and P. Patel, *Pattern Recognition in Time Series*. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2002.

[Lin et al. 2003] J. Lin, E. Keogh, S. Lonardi and B. Chiu, *A Symbolic Representation of Time Series with Implications for Streaming Algorithms*. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pp. 2-11, 2003.

[Lin et al, 2007] J. Lin, E. Keogh, L. Wei and S. Lonardi, *Experiencing SAX: a Novel Symbolic Representation of Time Series*. Data Mining and Knowledge Discovery, Vol. 15, issue 2, pp. 107–144, Oct. 2007.

[Lin, Li 2009] J. Lin and Y. Li, *Finding Structural Similarity in Time Series Data Using Bag-of-Patterns Representation.* In SSDBM'2009 Proceedings of the International Conference on Scientific and Statistical Database Management, Springer, Berlin, Germany, pp. 461–477, 2009.

[Lingras, Huang 2005] P. Lingras, X. Huang. *Statistical, Evolutionary, and Neurocomputing Clustering Techniques: Cluster-Based vs Object-Based Approaches*. Artificial Intelligence Review, Vol 23, Issue 1, pp 3–29, 2005.

[Liu, Hsu 1998] B. Liu, W. Hsu, Y. Ma. *Integrating Classification and Association Rule Mining*. In Proceedings of the 1998 International Conference on Knowledge Discovery and Data Mining, New York, NY, pp. 80–86, 1998.

[Lkhagva et al. 2006] B. Lkhagva, Y. Suzuki, and K. Kawagoe. *Extended SAX- Extension of Symbolic Aggregate Approximation for Financial Time Series Representation*. Working paper DEWS2006 4A-i8, Graduate School of Science and Engineering, Ritsumeikan University, 6p, 2006.

[Loh, Vanichsetakul 1986] W.Y. Loh and N. Vanichsetakul. *Tree-Structured Classification Via Generalized Discriminant Analysis*. In Technical Report Nº 786, Department of Statistics, University of Wisconsin, Madison, Wisconsin, USA, 1986.

[Loh et al. 2000] W.Y. Loh, S. Kim, and K. Whang. *Index Interpolation: an Approach to Subsequence Matching Supporting Normalization Transform in Time-Series Databases*. In Proceedings of the 9th International Conference on Information and Knowledge Management, ACM, New York, NY, USA, pp. 480-487, 2000.

[Loh 2008] W.Y. Loh. *Classification and regression tree methods*. Encyclopedia of statistics in quality and reliability, Wiley, 1800p. 2008.

[Lorr 1983] M. Lorr. *Cluster Analysis for Social Scientists: Techniques for Analyzing and Simplifying Complex Blocks of Data*. Jossey-Bass Inc. San Francisco, CA, USA, 1983.

[Maimon 2010] O. Maimon. *Data Mining and Knowledge Discovery Handbook*. Second edition, Springer, New York, NY, USA, 1306p, 2010.

[Malinowski et al. 2013] S. Malinowski, T. Guyet, R. Quiniou and R. Tavenard. *1d-SAX a Novel Symbolic Representation for Time Series*. In Proceedings of the International Symposium on Intelligent Data Analysis, UK, 2013.

[Mallat 1987] S.G. Mallat. *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*. In Technical Reports (CIS), Department of Computer & Information Science, University of Pennsylvania, USA, 1987.

[Mallat 1989] S.G. Mallat. *A Theory for Multiresolution Signal Decomposition- The Wavelet Representation*. In Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol II, issue 7, pp. 674-693, 1989.

[Mannila et al. 1994] H. Mannila, H. Toivonen, and A.I. Verkamo. *Efficient Algorithms for Discovering Association Rules*. In AAAI-94 Workshop on Knowledge Discovery in Databases, pp. 181-192, 1994.

[Moler 2004] C. Moler. *Numerical Computing with MATHLAB*. In SIAM Website, 2004.

[Molina et al. 2009] J. Molina, J. García, A. Bicharra, R. Melo, and L. Correia. *Segmentation and classification of time Series - Real Case Studies*. Intelligent Data Engineering and Automated Learning - IDEAL, Springer, Berlin, Germany, pp. 743-750, 2009.

[Mollazade et al. 2012] K. Mollazade, M. Omid, and A. Arefi. *Comparing Data Mining Classifiers for Grading Raisins Based on Visual Features*. In Journal of Computers and Electronics in Agriculture, Vol 84, Elsevier, pp. 124-131, 2012.

[Morgan, Sonquist 1963] J. N. Morgan and J. A. Sonquist. *Some results from a non-symmetrical branching process that looks for interaction effects*. Young, Vol 8, p. 5, 1963.

[Nason 2006] G. Nason. *Stationary and Non-Stationary Time Series*. Nason, G.P. *Stationary and non-stationary time series*. In Chapter 11 of Statistics in Volcanology, Geological Society of London, London, UK, 29p, 2006.

[Needleman, Wunsch 1970] S. B. Needleman and C. D. Wunsch, *A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins*. In Journal of Molecular Biology, Vol. 48, issue 3, pp. 443-453, 1970.

[Noda et al. 1999] E. Noda, A. Freitas, and H. Lopes. *Discovering interesting prediction rules with a genetic algorithm*. Proceedings of the Conference on Evolutionary Computation (CEC-99), pp. 182-196, Washington, DC, USA, 1999.

[Olszewsky 2001] R. Olszeusky, *Generalized Feature Extraction For Structural Pattern Recognition in Time Series Data*. PhD Thesis, School of Computer Science, Carnegie Mellon University Pittsburgh, PA, USA, 2001.

[Ordóñez et al. 2008] P. Ordóñez, M. des Jardins, C. Feltes, C. Lehmann, and J. Fackler. *Visualizing Multivariate Time Series Data to Detect Specific Medical Conditions*. In Proceedings of the AMIA Annual Sympposium, pp. 530–534, 2008.

[Ordóñez, Jardins 2011] P. Ordóñez, and M. des Jardins. *Multivariate Time Series Analysis of Clinical and Physiological Data*. In http://ccom.uprrp.edu/~pordonez/ghc10_ordonez.pdf, 2011.

[Padmanabhan, Tuzhilin 1998] B. Padmanabhan and A. Tuzhilin, *A Belief-Driven Method for Discovering Unexpected Patterns*. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, Rakesh Agrawal and Paul Stolorz (Eds.), The AAAI Press, Menlo Park, CA, USA, pp. 94–100, 1998.

[Pandey, Satish 1998] S.K. Pandey, and L. Satish. *Multiresolutio Signal Decomposition- A New Tool for Fault Detection in Power transformers During Impulse tests* In IEEE Transactions on Power Delivery. Vol 13, issue 4, pp. 1194-1200, 1998.

[Papetrou et al. 2011] P. Papetrou, V. Athitsos, M. Potamias, G. Kollios, and D. Gunopulos. *Embedding-Based Subsequence Matching in Time-Series Databases*. In ACM Transactions on Database Systems, Vol. 36, issue 3, Article 17, 2011.

[Park, Chu 2001] S. Park,S. Kim, and W.W. Chu. *SBASS: Segment-Based Approach for Subsequence Searches in Sequence Databases*. In proceedings of the 16th ACM Symposium on Applied Computing. Las Vegas, NV, USA, pp. 248-252, 2001.

[Parthasaradhi et al. 2005] S. Parthasaradhi, R. Derakhshani, L. Hornak and S. Schuckers. *Time Series Detection of Perspiration as a Liveness Test in Fingerprint Devices*. In IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol 35, issue 3, pp. 335-343, 2005.

[Pasquier et al. 1999] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal. *Discovering Frequent Closed Itemsets for Association Rules*. In Proceedings of the 7th International Conference on Database Theory, Jerusalem, Israel, pp. 398–416, 1999.

[Pelleg, Moore 2000] D. Pelleg, and A. Moore, *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*. In Proceedings of the 17th International Conference on Machine Learning, Pat Langley (Ed.), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 727–734, Jul. 2000.

[Pelt 2003] J. Pelt. *Astronomical Time Series Analysis*. Lecture Notes, Tartu Observatory, Oulu University, Oulu, Finland, 2003.

[Pham et al. 2010] D. Pham, Q. Le and T. Dang, *Two Novel Adaptive Symbolic Representations for Similarity Search in Time Series Databases*. In Proceedings of the 12th International Asia-Pacific Web Conference, pp. 181-187, 2010.

[Phyu 2009] T. Phyu. *Survey of Classification Techniques in Data Mining*. In Proceedings of the IMECS'09 International MultiConference of Engineers and Computer Scientists, Vol I, 5p, 2009.

[Quinlan 1983] J.R. Quinlan. *Learning efficient classification procedures and their application to chess end games*. In Machine learning: An artificial intelligence approach. R. S., Michalski, J. G., Carbonell, & T. M., Mitchell (Eds.), Morgan Kaufmann, Los Altos, CA, 1983.

[Quinlan 1993] J. Quinlan, *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, USA, 312p, 1988.

[Rafiei 1999] D. Rafiei. *On Similarity-Based Queries for Time Series Data*. In Proceedings of the 15th IEEE International Conference on Data Engineering, IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 410-417, 1999.

[Raikwal, Saxena 2012] J. S. Raikwal, and K. Saxena. *Performance Evaluation of SVM and K-Nearest Neighbor Algorithm over Medical Data Set*. In the International Journal of Computer Applications, Vol 50, issue 14, 2012.

[Rasmussen 1992] E. Rasmussen. *Clustering Algorithms. Chapter 16, in Information Retrieval Data Structures and Algorithms*. Prentice HAll, pp. 401-426, 1992.

[Rice 1963] S.O. Rice, *Noise in FM Receivers*. Chapter 25 in Time Series Analysis. M. Rosenblatt, John Wiley and Sons, New York, NY, USA, 1963.

[Richter, Aamodt 2006] M. Richter and A. Aamodt. *Case-Based Reasoning Foundations*. In The Knowledge Engineering Review, Cambridge University Press, UK, Vol 20, issue 3, pp. 203-207, 2006.

[Ritschard 2010] G. Ritschard. *CHAID and Earlier supervised Tree Methods* In Cahiers du Département d'économétrie, nº 2010.02, Department of Econometrics, University of Geneva, Geneva, Switzerland, 28p, 2010.

[Romani 2010] L. Romani. *Integrating Time Series Miming and Fractals to Discover Patterns and Extreme Events in Climate and Remote Sensing Databases*. PhD Thesis, Instituto de Ciencias Matematicas e de Computaçao, Universidade de Sao Paulo, Sao Carlos, SP, Brasil, 2010.

[Sakoe 1971] H. Sakoe, and S. Chiba. *A Dynamic Programming Approach to Continuos Speech Recognition*. In Proceedings of International Congress on Acoustics, Budapest, Hungary, Paper 20C-13, 1971.

[Samia 2004] M. Samia. *A Representation of Time Series for Temporal Web Mining*. In Proceedings of the 16th GI-Workshop on the Foundations of Databases, pp. 103-107, 2004.

[Santamaría 2011] A. Santamaría, *Modelo de Descubrimiento del Conocimiento en Series Temporales Numéricas Aplicando Métodos Simbólicos.* PhD Tesis - Departamento de Lenguajes, Sistemas Informáticos e Ingeniería de software, Facultad de Informática, Escuela Politécnica de Madrid, 2011.

[Savasere et al. 1995] A. Savasere, E. Omiecinski, and S. Navathe. *An Efficient Algorithm for Mining Association Rules in Large Databases*. In Proceedings of the 21st Very Large Databases Conference, Zurich, Swizerland, 1995.

[Saxena, Gadhiya 2014] A. Saxena, and S. Gadhiya. *A Survey on Frequent Pattern Mining Methods*. In International Journal of Engineering Development and Research, Vol 2, issue 1, pp. 92-96, 2014.

[Senin, Malinchik 2013] P. Senin, and S. Malinchik. *SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model*. In Proceedings of the IEEE 13th International Conference on Data Mining, IEEE, Dallas, TX, USA, pp. 1175-1180, 2013.

[Shannon 1948] C. Shannon. *The mathematical theory of communication*. In The Bell System Technical Journal, vol 27, pp. 379–423, 1948.

[Shied 2008] J. Shieh, and E. Keogh, *iSAX Indexing and Mining Terabyte Sized Time Series*, in KDD'08 Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM New York, NY, USA, Pages 623-631, 2008.

[Shin et al. 2005] S. Shin, A. Ray, and V. Rajagopalan. *Symbolic Time Series for Anomaly Detection: A Comparative Evaluation*. Signal Processing 85, Elsevier, pp. 1859–1868, 2005.

[Silvasan et al. 2014] C. Silvasan, I. Borza and Romania. *Intelligent Cities Interdependent Time Series Graphically Analyzed. Strategies in Traffic Control and Street Lighting*. In Latest Trend in Energy, Environment and Development, Salermo, Italy, pp. 344-348, 2014.

[Singh, Valtorta 1995] M. Singh, and M. Valtorta. *Construction of Bayesian Network Structures From Data: A Brief Survey and an Efficient Algorithm*. In International Journal of Approximate Reasoning, Vol 12, pp. 111-131, 1995.

[Singh et al. 2013] S.K. Singh, S.J. Khan, and M.K. Singh. *Discrete Wavelet Transform: A Technique for Speech Compression & Decompression*. In International Journal of Innovations in Engineering and Technology, Special Issue – ICAECE, 2013.

[Sivanandam, Deepa 2008] S.N. Sivanandam, and S.N. Deepa. *Introduction to Genetic Algorithms*. Springer-Verlag, Berlin, Heidelberg, 453p, 2008.

[Sokolova, Lapalme 2009] M. Sokolova, G. Lapalme. *A Systematic Analysis of Performance Measures for Classification Tasks.* In Information Processing and Management Journal, Elsevier, Vol. 45, issue 4, pp. 427-437, 2009.

[Sreemathy, Balamuguran 2012] J. Sreemathy, and P. Balamuguran. *An Efficient Text Classification Using KNN and Naive Bayesian*. In International Journal on Computer Science and Engineering, Vol 4, pp. 392-396, 2012.

[Srikant, Agrawal 1996] R. Srikant, R. Agrawal. *Mining Sequential Patterns: Generalizations and Performance Improvements*. In Proceeding of the 5th International Conference on Extending Database Technology, Avignon, France, pp. 3–17, 1996.

[Stollnitz et al. 1995] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin. *Wavelets for Computer Graphics- A Primer*. In Journal IEEE Computer Graphics and Applications, vol 15, issue 3, IEEE Computer Society Press Los Alamitos, CA, USA, pp. 76-84, 1995.

[Struhl 1992] S. Struhl. *Classification tree methods- AID, CHAID and CART*. In QUIRK'S Marketing research Media, 1992.

[Suriya et al. 2012] S. Suriya, S.P. Shantharajah, and R. Deepalakshmi. *A Complete Survey on Association Rule Mining with Relevance to Different Domain*. In Internations Journal of Scientific and Technical Research. Vol 1, issue 2, pp. 163-168, 2012.

[Svensson 2011] S.A. Svensson. *Implementing a Fuzzy Classifier and Improving its Accuracy using Genetic Algorithms*. In Proceedings of the Extra Mini-conference on Interesting Results in Computer Science and Engineering, Sweeden, 2011.

[Tee, Wu 1972] L.H. Tee, and S.U. Wu. *An Application of Stochastic and Dynamic Models for the Control of a Papermaking Process*. In Technometrics, Nº 14, pp. 481-496, 1972.

[Thomson 1994] D.J. Thomson. *Jackknifing multiple-window spectra*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol 1, 73-76, 1994.

[Tibshivani 2013] R. Tibshivani. *Clustering 1: k-means, k-medoids*. In Data Mining - Optional Reading, 2013.

[Timofte 2007] R. Timofte. *Short-term Time Series in Authomatic Speech Processing*. MSc Thesis, Department of Computer Science and Statistics, University of Joensuu, Joensuu, Finland, 70p, 2007.

[Toivonen 1996] H. Toivonen. *Large Databases for Association Rules*. In Proceedings of the 22nd Very LArge Databases Conference, Bombay, India, 1996.

[Toyoda et al. 2013] M. Toyoda, Y. Sakurai, and Y. Ishikawa. *Pattern Discovery in Data Streams under the Time Warping Distance*. In The VLDB Journal, Vol. 22, Issue 3, Springer, pp. 295-318, 2013.

[Wu et al. 1996] D. Wu, D. Agrawal, A. El Abbadi, A. Singh, and T.R. Smith. *Efficient retrieval for browsing large image databases*. In Procedures of the 5th International Conference on Knowledge Information, Rockville, MD, USA, pp. 11-18, 1996.

[Wu et al. 2008] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. F. M. Ng, B. Liu, P. S. Yu, Z-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. *Top 10 algorithms in data mining*. In Knowledge Information Systems. Vol 14, issue 1, pp. 1–37, 2008.[Yadav 2013] A. Yadav. *A Survey Of Issues And Challenges Associated With Clustering Algorithms*. In the International Journal for Science and Emerging Technologies with Latest Trends. Vol 10, issue 1, pp. 7-11, 2013.

[Yafee, McGee 2000] R. Yafee, and M. McGee. *An Introduction to Time Series Analysis and Forecasting with Applications of SAS and SPSS*. Academic Press, New York, NY, 555p, 2000.

[Yi, Faloutsos 2000] B.K. Yi, and C. Faloutsos. *Fast Time Sequence Indexing for Arbitrary Lp Norms*. In Proceedings of the 26th Conference on Very Large databases, Cairo, Egypt, pp. 385-394, 2000.

[Yin et al. 2003] X. Yin, and J. Han. *CPAR: Classification Based on Predictive Association Rules.* In Proceedings of the 2003 SIAM International Conference on Data Mining, San Francisco, CA, pp. 331–335, 2003

[Yule 1927] G.U. Yule. *On a Method of Investigating Periodicities in Disturbed Series with Special Reference to Wolfer's Sunspot Numbers*. In Philosophical Transactions of the Royal Society, Series A, Nº 226, pp. 267-298, 1927.

[Yuzuriha 1960] T. Yuzuriha, *The Autocorrelation Curves of Schizophrenic Brain Waves and the Power Spectrum*. In Psychiatria et Neurologia Japonica, Nº 26, pp. 911-924, 1960.

[Zaiane 1999] O. Zaïane. *Introduction to Data Mining*. In Chapter I of CMPUT690 Principles of Knowledge Discovery in Databases, 15pp, 1999.

[Zaki 2000] M.J. Zaki, *Scalable Algorithms for Association Mining.* In Proceedings of the IEEE Transactions on Knowledge Data Engineering, Vol. 12, issue 3, pp. 372–390, 2000.

[Zaki 2001] M.J. Zaki. *SPADE: an efficient algorithm for mining frequent sequences*. Machine Learning, Vol 40, issue 1 - 2, pp. 31–60, 2001.

[Zhao, Bhowmick 2003] Q. Zhao, and S. Bhowmick. *Association Rule Mining: A Survey*. Technical Report, Nº 2003116, CAIS, Nanyang Technological University, Singapore, 2003.