TÉCNICAS DE MINERÍA DE DATOS APLICADAS A SERIES TEMPORALES BURSÁTILES

PRESENTADA POR:

ROSA BASAGOITI ASTIGARRAGA

DIRIGIDA POR:

TIM SMITHERS

MONDRAGÓN, 2007

AGRADECIMIENTOS

La autora desea dar las gracias a Elizabet Juaristi, compañera y constante apoyo en los sucesivos estadios de esta tesis.

Al doctor Eamonn Keogh por su ayuda y por sus estupendos tutoriales,sin los cuales jamás hubiese podido tener una visión tan genérica del problema al cual se enfrentaba.

Agradecimiento al Ministerio de Educación y Ciencia por la financiación del proyecto DPI2006-03060 dentro del Programa del Plan Nacional de Investigaciones Científicas.

Y por último, a su familia, por su apoyo y confianza.

RESUMEN

Analizar grandes cantidades de datos con rapidez es uno de los mayores retos a los que se enfrentan las empresas hoy en día. Los sistemas informáticos ofrecen la posibilidad de almacenar gran cantidad de datos pero es necesario encontrar la manera de analizarlos en un tiempo razonable. Las bases de datos temporales son bases de datos excepcionalmente grandes cuyo análisis resulta particularmente laborioso cuando no imposible. Reducir la dimensionalidad de los datos para que el tiempo necesario para dicho análisis sea admisible es la primera prioridad. La definición de una buena medida de la similitud que permita comparar las series temporales entre sí será la segunda. Si nos aproximamos a un dominio en concreto, el de las series temporales económicas, vemos que las series temporales bursátiles son estudiadas con el fin de detectar posibles cambios de tendencia que avisen de las oportunidades de compra o de venta de títulos. El análisis técnico observa la formación de ciertos patrones y realiza un seguimiento de los mismos utilizándolos como referencia a la hora de tomar las decisiones oportunas. Es posible utilizar los fundamentos del análisis técnico como base para la representación de las series temporales para estudiar,.a continuación, usando distintas medidas de la distancia, resultados obtenidos por algoritmos de clustering. Se analizarán los resultados tanto desde el punto de vista de la eficacia como desde la eficiencia.

TABLA DE CONTENIDO

LISTA DE ILUSTRACIONES

LISTA DE FIGURAS

LISTA DE TABLAS

CAPÍTULO 1

INTRODUCCIÓN

En las últimas décadas ha aumentado considerablemente la posibilidad de generar y guardar gran cantidad de datos. El uso extendido de los códigos de barras, el hecho de que muchas de las transacciones a nivel científico, gubernamental o empresarial se hagan de manera electrónica, junto con los avances en las herramientas de captura de datos, tanto de tipo texto como de imágenes recolectadas por satélites, sumado al uso masivo de la Web, han hecho necesario el uso de nuevas técnicas que puedan ayudarnos a transformar estas inmensas cantidades de información en conocimiento e información útil.

La minería de datos es el conjunto de conceptos y técnicas que procuran explotar los datos contenidos en grandes bases de datos de manera automática,

extrayendo de ellos los patrones que puedan estar representados implícitamente. Es un área multidisciplinar donde, además de incluir áreas tales como la inteligencia artificial, la estadística o las redes neuronales, debemos tener en cuenta el dominio de aplicación en el que estemos trabajando.

El análisis de bases de datos de series temporales despierta interés en áreas tan diversas como la ingeniería, la medicina o las finanzas. Los datos en forma de serie temporal son difíciles de procesar y analizar. La búsqueda de similitudes entre series temporales es fundamental para poder realizar cualquiera de las labores de minería de datos, algunas de las cuales se enumeran a continuación: clasificación, obtención de reglas, clustering, y búsqueda en grandes bases de datos de la serie más similar a una serie dada. En el caso de las series temporales y debido a la dimensionalidad inherente, cualquier algoritmo de minería debería usar una aproximación a los datos de manera que, a la vez que capture sus características más esenciales, obtenga una representación reducida que facilite su posterior uso. La representación obtenida constituirá, por lo tanto, un mecanismo de compresión.

La minería de tipos de datos complejos tales como datos espaciales, datos multimedia, series temporales, datos textuales, y datos extraídos de la web es una tarea que está ganando en importancia. Las técnicas clásicas de minería de datos han de ser desarrolladas para que puedan ser utilizadas con este tipo de datos de manera que sea posible extraer información fructífera de estos repositorios de datos complejos. Todos los mecanismos mostrados a lo largo de esta tesis, con excepción del uso de los patrones específicos del análisis técnico, serían

aplicables a cualquier otro problema en el que los datos a tratar fuesen series temporales.

Por otra parte, los datos financieros recolectados en los bancos o instituciones financieras son siempre completos, fiables y de gran calidad. Esto facilita el análisis sistemático de los datos, además del uso de algoritmos de minería para resolver, entre otras, algunas de las problemáticas que a continuación se presentan:

- Predicción en el pago de los préstamos.
- Análisis de las políticas en la concesión de créditos a clientes.
- Clasificación y clustering de los clientes para marketing personalizado.
- Detección del fraude.

El dominio de aplicación seleccionado en esta tesis para desarrollar los principios y avances sobre los que progresa la minería de datos son los mercados financieros, concretamente el análisis técnico.

El análisis técnico es el estudio de los movimientos del mercado usando para ello gráficos con el propósito de pronosticar las futuras tendencias de los precios (Murphy. 2000). Cuando nos referimos a movimientos del mercado, se incluyen las tres fuentes principales de información disponibles: el precio, el volumen y el interés abierto. En este caso, para realizar el trabajo que a continuación se va a mostrar, la única información a la que se va a prestar atención es la concerniente al movimiento de precios.

Estos precios constituyen secuencias de valores que cambian con el tiempo y son medidas a intervalos regulares. El resultado obtenido son series temporales,

para las cuales se han desarrollado mecanismos específicos dentro de la minería de datos.

La peculiaridad que presenta el análisis técnico, que lo hace adecuado para el uso de herramientas computerizadas, es el primer principio en el que se sustenta: "los movimientos del mercado lo descuentan todo". El analista técnico supone que todo lo que pueda afectar al mercado se encuentra ya reflejado en el propio precio. Los movimientos de precios deberán, por lo tanto, reflejar los cambios de la oferta y la demanda. Si la oferta sube, los precios deberían subir y, si baja, los precios deberían bajar. Todo ello, sin preocuparnos de las razones por las cuales los precios suben o bajan.

Surge entonces el concepto de tendencia. El análisis técnico representa los precios gráficamente con el fin de identificar tendencias. Identificadas las tendencias, el objetivo es que las transacciones que se realicen en el mercado vayan en la dirección de dichas tendencias. Suponemos que una tendencia en movimiento seguirá en la misma dirección hasta que comience a volver atrás.

El análisis técnico ha identificado y clasificado ciertos patrones extraídos a partir de los gráficos. Estos patrones revelan la sicología alcista o bajista del mercado y dado que dichos patrones han funcionado bien en el pasado, se asume que seguirán funcionando bien en el futuro. Presuponemos, entonces, que la comprensión del futuro está en el estudio del pasado.

La teoría de Dow es la piedra angular del análisis técnico. Sus principios básicos son los que a continuación se exponen:

Las medias lo descuentan todo: la suma y tendencia de las transacciones de la Bolsa representan la suma de todo el conocimiento del pasado, el inmediato y el remoto, aplicado el descuento del futuro.

El mercado presenta tendencias ascendentes y descendentes. Se define una tendencia ascendente como una situación en la que cada sucesiva recuperación cierra más alto que la recuperación previa y cada sucesivo nivel bajo cierra mas alto que el nivel bajo de la recuperación previa. Según Dow, es posible aplicar a los mercados los mismo efectos de acción reacción que se aplican al universo físico. Es posible observar que "cuando un valor llega a lo más alto, a continuación tiene un moderado descenso y luego vuelve a subir para aproximarse a las cifras más altas. Si después de un movimiento tal, la cotización vuelve a retroceder, probablemente bajará una cierta distancia".

Dow consideraba que una tendencia tenía tres partes: primaria, con duración superior a un año, secundaria; que representa correcciones en la tendencia primaria y que durará de tres semanas a tres meses; y tendencia menor con una duración inferior a 3 semanas. Además, las tendencias principales, según Dow, tienen tres fases bien diferenciadas: una fase de acumulación o compra bien informada de los inversores más astutos, una fase de participación pública en la que comienzan a participar la gran mayoría de los que siguen tendencias, y una fase de distribución en la que los periódicos publican noticias sobre subidas progresivas.

Las medias deben confirmarse entre ellas: refiriéndose a las medias Industriales y a las de los Ferrocarriles. Dow pensaba que ambas medias debían confirmar el

comienzo o la continuación de una media alcista. El volumen debe, también, confirmar la tendencia, creciendo o decreciendo en la dirección de la tendencia principal.

Además, se presume que una tendencia está en vigor hasta que da señales definitivas de que una fuerza externa le ha hecho retroceder.

Antes de pasar a explicar el contenido de esta tesis, será necesario mencionar la Teoría del Paseo Aleatorio, la cual dice que los cambios en los precios son serialmente independientes y que el histórico de precios no es un indicador de confianza de la futura dirección de los mismos.

El funcionamiento de los mercados puede, ciertamente, parecer aleatorio. Esperamos que dicha aleatoriedad desaparezca en cierta medida según vayamos introduciendo los mecanismos implementados.

Lo ideal sería trasladar a algoritmos de minería de datos la capacidad de observación empírica y la experiencia práctica de los analistas técnicos, de manera que sea posible anticiparse a los mercados. No es esta una tarea fácil, los mercados descuentan toda la información rápidamente, tan rápidamente que, a veces, no hay manera de sacar partido de dicha información.

Se tratará a lo largo de la tesis de aunar las necesidades de los analistas técnicos con los avances que se han obtenido como resultado de la investigación realizada en la minería de series temporales en la última década. Para poder lograr dicho objetivo, se observarán las series temporales bursátiles desde la perspectiva del analista técnico. Se desarrollará una nueva representación con patrones usados por dicha disciplina. El más conocido será el patrón de cambio de tendencia,

hombro-cabeza-hombro. En la Figura 1.1 podemos observar un ejemplo de una serie temporal en el cual existe un patrón de este tipo. Se muestran gráficamente una serie temporal, la serie suavizada y los extremos que constituyen el patrón. El mostrado es uno de los patrones de más confianza y el más importante. De hecho, sería posible pensar que la mayoría del resto de los patrones es sólo una variación suya.

En la gráfica se muestra la serie original, la misma serie suavizada mediante el uso de un estimador kernel que eliminará el ruido y el patrón detectado. Una vez detectado el patrón, se debe almacenar cierta información referente a él. Almacenar esta información supone adoptar una representación.

Figura 1. 1. Una serie temporal, la serie suavizada correspondiente y un patrón hombro-cabeza-hombro.

Para cualquier representación, los aspectos más importantes a tener en cuenta son los siguientes (Shatkay and Zdonik. 1996).

- Desde el punto de vista del espacio ocupado, la nueva representación debe ser más eficiente que la representación original
- Si tengo dos secuencias *a* y *b* y *características (a), características (b)*, y sus respectivas representaciones en el nuevo alfabeto, sería deseable que *rep(a)=rep(b)⇔ características (a)=características (b)*. Ésta es una condición bastante fuerte que se puede relajar.
- Las representaciones similares deben corresponderse con características similares. Por otra parte, se deben preservar características importantes de la secuencia.
- Sería aconsejable que se pudiese generar un índice convenientemente además de soportar consultas sobre patrones generales en lugar de consultas sobre valores concretos.
- Por último, es importante que la representación pueda ser utilizada para predecir y/o deducir valores de la secuencia que no se han muestreado.

Estrechamente relacionada con la representación adoptada para las series temporales, se encuentra la necesidad de definir una medida de la distancia que use dicha representación en labores de minería de datos tales como la búsqueda de la secuencia o subsecuencia más parecida a una secuencia dada. Según la caracterización ofrecida en este trabajo de las consultas aproximadas generalizadas, podremos afrontar la búsqueda de subsecuencias semejantes desde una perspectiva más general siempre que se contemplen las siguientes características:

- Hay algún tipo de patrón general, independiente de valores concretos que caracteriza los resultados deseados. Definir el patrón supone definir la consulta.
- La consulta denota en realidad un conjunto *S* de subsecuencias en lugar de una sola.
- *S* será un conjunto cerrado ante ciertas transformaciones que preserven su comportamiento tales como traslación en tiempo y en amplitud, dilataciones y contracciones, etc.
- El resultado será una coincidencia exacta si el resultado pertenece a *S*, en otro caso será un resultado aproximado.

Han sido múltiples los intentos para desarrollar sistemas para analizar datos de series temporales correspondientes al dominio bursátil. En 1993, en la Universidad de Wisconsin se desarrolló MIMSY (Roth. 1993). En dicho sistema era posible, mediante un lenguaje parecido al SQL, la búsqueda de patrones temporales en datos bursátiles. Dicho sistema utilizaba mecanismos para poder especificar restricciones temporales a la hora de su recuperación.

Una buena recopilación de las distintas medidas de similitud utilizadas entre series temporales del dominio bursátil fue utilizado para el clustering de series temporales en (Gavrilov et al. 2000).

Estos y otros muchos trabajos han intentado afrontar el problema de la representación, la consulta, y otras labores de minería de datos en el dominio que nos ocupa. Desde que Agrawal, Faloutsos y Swarni presentaran (Agrawal. 1993), un trabajo ampliamente citado, se han desarrollado múltiples técnicas con el

objetivo de que la minería de datos pudiese ser aplicada con éxito en el tratamiento de datos bursátiles. El tratamiento de estos datos ha tenido siempre muy presente el análisis técnico, una muestra de ello es el artículo (Lo et al. 2000) donde se estudiaba la presencia y la información suministrada por los patrones que aquí se van a utilizar. En (Ming Dong. 2002), por otra parte, se estudia la naturaleza difusa de los patrones del análisis técnico utilizando para ello la lógica difusa.

La búsqueda de subsecuencias semejantes considerando los patrones de análisis técnico en datos bursátiles ha sido estudiada en, al menos, (Rafiei. 1999) y (Agrawal et al. 1995) donde se definía un lenguaje de consulta de formas que podía ser utilizado para rescatar patrones del tipo hombro-cabeza-hombro.

Todo lo mencionado anteriormente ha de servir para que, además de detectar patrones, suministre información relevante referente a las decisiones respecto a las inversiones a realizar. La rentabilidad y la eficiencia de los patrones encontrados a la hora de decidir las inversiones diarias serán fundamentales. Lo ideal sería ayudar a los inversores a identificar oportunidades para realizar operaciones rentables en tiempo real. Según (Roth. 1993) variables tales como la calidad o grado de pertenencia de una determinada serie a un cierto tipo de patrón, así como la longitud del mismo, han resultado determinantes a la hora de obtener unos buenos resultados.

Contribuciones

Esta tesis usa los principios del análisis técnico para poder dar soporte al analista de bolsa a la hora de decidir sus inversiones. La aproximación que se va a emplear va a ser la más parecida a la que los propios técnicos utilizan. Cuando éstos observan los gráficos, van trazando rectas que unen máximos y mínimos procurando predecir los movimientos siguientes. Mientras se van trazando dichas rectas, es posible observar algunas de las formaciones que pronostican un comportamiento reconocible y predecible del mercado. Además, de poder trazar dichas rectas, los analistas intentan buscar referencias de precios pasados que hayan tenido un comportamiento semejante. Tratarán , a continuación, de asociar el actual comportamiento al ya reproducido con anterioridad.

Técnicamente, la representación utilizada, es una aproximación lineal a tramos. Es una de las representaciones más utilizadas en minería de series temporales. Representa la serie temporal mediante k segmentos lineales. De hecho, es una buena técnica para reducir la complejidad de los datos originales. Ya fue utilizada en (Ge. 1998) para explorar la coincidencia de patrones propios del análisis técnico, aunque estos patrones se extraían en dicho trabajo mediante un mecanismo distinto. La representación lineal a tramos es una aproximación de los datos y por lo tanto debemos tener presente el error inherente a dicha representación. Tal como se verá a continuación, se han realizado pruebas para poder medir dicho error.

Han sido muchos los algoritmos de segmentación utilizados para obtener representaciones más reducidas de las series temporales. En (Keogh et al. 2004) se puede tener una reciente recopilación. El propio autor desarrolló en (Keogh. 2001a) otro mecanismo que también utilizaremos en las pruebas mostradas en los siguientes capítulos. La mayoría de los algoritmos planteados son generales, es decir, servirán para cualquier tipo de dato, bursátil o de cualquier otro dominio. En esta tesis, la segmentación obtenida viene determinada por aquellos extremos de la serie temporal susceptibles de constituir un patrón previamente determinado. En ese sentido, la representación obtenida, a pesar de ser una representación basada en segmentos lineales, es totalmente dependiente del dominio de aplicación considerado. La misma aproximación ha sido también estudiada desde otra perspectiva en (Chung et al. 2004).

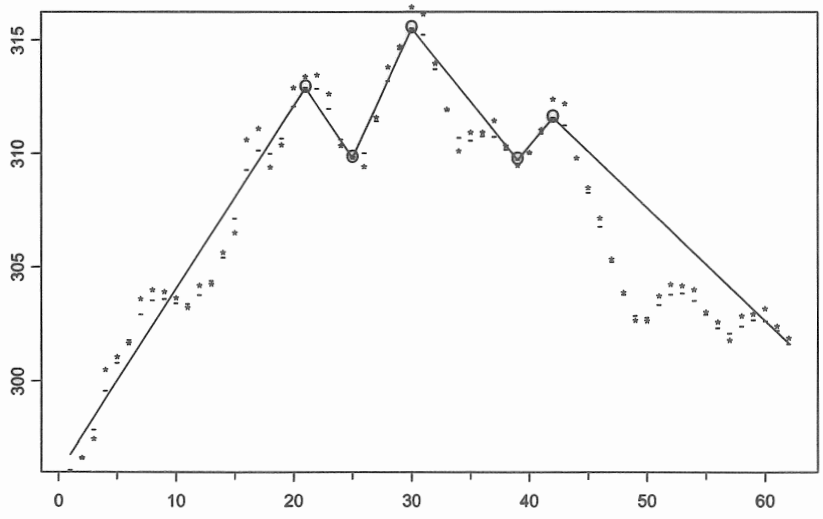


Figura 1. 2. La serie, el patrón detectado y su representación mediante segmentos

En lo que respecta a la medida de la distancia, esta deberá reflejar el hecho de que dos series temporales se están comportando de manera semejante, no necesariamente igual. Se explorarán múltiples medidas de la distancia que, utilizadas con todos o parte de los segmentos constitutivos de la representación, dan la posibilidad de una recuperación rápida y flexible de la información almacenada. Se han explorado distancias entre segmentos en las cuales la posibilidad de poder asignar un peso a cada segmento es, en realidad, una posibilidad que es posible explotar de múltiples maneras (Keogh. 1998).

Otras distancias utilizadas contemplan la posibilidad de que tengamos distorsión en el eje temporal. La posibilidad de que exista dicha distorsión hace que un patrón se pueda reproducir en una serie más lentamente o más rápidamente que el mismo patrón encontrado en otra serie. Estas distancias serán las denominadas distancias con alineación dinámica temporal.

A continuación se muestran las fases de las que constará el proceso de representación y exploración de las series temporales económicas en estudio:

- Selección de los puntos importantes (máximos y mínimos) e identificación de los patrones en el dominio de aplicación.
- Representación de las series mediante segmentos lineales que se ajustan a dichos patrones.
- Valoración de las representaciones obtenidas con respecto a otras representaciones ampliamente utilizadas midiendo para ello los errores obtenidos en las reconstrucciones de las series temporales.

- Establecer medidas de la similitud que se utilizarán en la representación planteada.
- Comparaciones empíricas del comportamiento de la representación y de las distancias planteadas en la resolución de labores de clustering.

Descripción del contenido

Para facilitar la lectura del documento, se resumen los contenidos capítulo a capítulo, que es posible encontrar en esta memoria.

Capítulo 2: revisión de los distintos tipos de problemas planteados en el área de minería de datos en series temporales. Se plantea el problema de la dimensionalidad inherente a los datos temporales y los distintos mecanismos que se han utilizado para sortear este problema. Se plantea la necesidad de usar representaciones que reduzcan la dimensionalidad de los datos además de estudiar las propiedades deseables de la medida de la distancia definida.

Capítulo 3: revisión más exhaustiva de las distintas representaciones que han sido utilizadas para reducir la dimensionalidad de los datos. Se presentan, entre otros, la Transformada Discreta de Fourier y la Transformada Discreta Wavelet además de otros mecanismos que representan las series temporales mediante segmentos. Se harán también breves menciones a aproximaciones simbólicas.

Capítulo 4: estudio de las medidas de la similitud entre series temporales que han sido utilizadas en sucesivos trabajos, medidas de la similitud que pueden, o

no, considerar distorsiones en el eje temporal e incluso que pueden permitir al usuario introducir nociones subjetivas de similitud.

Capítulo 5: propone una representación basada en puntos importantes de las series temporales usando patrones propios del análisis técnico. Se consideran distintas bases de datos y se plantean distintas problemáticas a tener en cuenta, tales como el error de reconstrucción, la calidad de la segmentación obtenida, la media de los patrones encontrados y la capacidad predictiva de estos patrones .

Capítulo 6: propone distintas distancias y presenta los resultados experimentales obtenidos en sucesivos clusterings.

Capítulo 7: muestra una alternativa cada vez más utilizada en tareas de minería de datos costosas, la paralelización.

Capítulo 8: servirá para presentar las conclusiones de los experimentos realizados, además de plantear los trabajos que se van a realizar en el futuro en esta línea de investigación.

CAPÍTULO 2

MINERÍA DE DATOS EN SERIES TEMPORALES

La minería de datos ha atraído una gran atención en la industria de la información debido a la gran cantidad de datos disponibles y a la necesidad de convertir dichos datos en información útil y en conocimiento.

Según una definición de minería de datos "es la extracción no trivial de información implícita potencialmente útil a partir de los datos".

La minería de datos puede ser vista como el resultado de la evolución natural de la tecnología de la información. Dicho proceso comenzó con el desarrollo de las funcionalidades de recopilación, creación, y gestión de bases de datos, para dar paso, alrededor de la mitad de los 80, a los sistemas de bases de datos avanzadas. A finales de esta década, podemos encontrar las bases de datos

multidimensionales y la minería de datos y a continuación, a partir de 1990, podemos encontrarnos con el desarrollo de sistemas de bases de datos basados en la web. La abundancia de datos, junto a la necesidad de disponer de herramientas de análisis potentes, se ha descrito como una situación de riqueza de datos frente a la de pobreza de información.

Las herramientas de minería de datos realizan análisis de datos y descubren patrones de datos importantes, contribuyendo a las decisiones estratégicas en los negocios y en investigaciones médicas y científicas.

El proceso de extracción de conocimiento de las bases de datos es un proceso más amplio que constará de varios pasos.

- La limpieza de los datos, con el objetivo de borrar ruido y datos inconsistentes.
- La integración de los datos, proceso en el cual se pueden combinar datos de varias fuentes.
- La selección de los datos, cuando se extraen los datos relevantes para la tarea de análisis de la base de datos.
- La transformación, cuando los datos son transformados o consolidados en formas apropiadas para el análisis, realizando, por ejemplo, operaciones de suma y agregación.
- La minería de datos, donde se aplican métodos inteligentes para extraer patrones a partir de los datos.

- La evaluación de los patrones, donde se identificarán los patrones realmente interesantes, usando para ello cierta medida de lo interesante que pueda resultar un patrón.
- Presentación del conocimiento, donde se utilizaran técnicas de visualización y técnicas de representación para presentar al usuario el conocimiento extraído.

Cualquiera de estos procesos puede ser aplicado sobre diferentes repositorios: bases de datos relacionales, almacenes de datos, bases de datos transaccionales, ficheros planos, bases de datos espaciales, bases de datos de series temporales, bases de datos textuales, o bases de datos multimedia, entre otros. Los problemas y las técnicas utilizadas pueden ser, además, distintas para cada uno de los repositorios. En la minería de datos se han estudiado algoritmos que facilitan el proceso de extracción de información. Hasta ahora, dicho proceso de extracción de información estaba centrado básicamente en el tratamiento de datos estáticos, datos acerca del mundo en un instante dado de tiempo. En la minería de datos estática, las bases de datos consisten en registros que caracterizan objetos en términos de atributos x_i que no cambian con el tiempo. Son registros relacionados con un objeto en un cierto instante de tiempo.

Disponemos, por lo tanto, de la descripción de un objeto dado en función de dichas variables:

$$x_1, x_2, x_3, \ldots\ldots, x_n$$

En la minería de datos dinámica o temporal, las bases de datos se corresponden con registros que caracterizan a los objetos en distintos instantes de tiempo:

$$x_{t1}, x_{t2}, \ldots\ldots x_{tn} \quad \text{donde } t = t_1, t_2, t_3, \ldots\ldots\ldots, t_n .$$

Muchos de los datos recogidos automáticamente por sensores o monitores son series temporales. Una serie temporal es una secuencia de números reales que representa la medida de una variable real a intervalos regulares de tiempo. Como ejemplo de series temporales, podemos tener los volúmenes de ventas a lo largo del tiempo, la lectura de la temperatura diaria, etc.

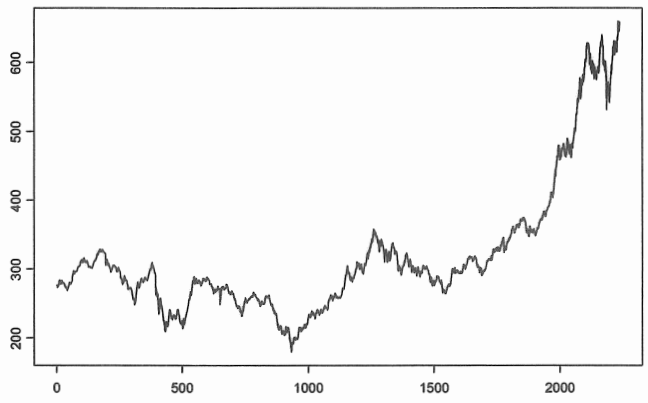


Figura 2. 1. Serie temporal correspondiente a los valores de cierre del Índice General de la bolsa de Madrid en el siguiente período: del 02/01/1990 al 25/5/2004.

A la hora de aplicar distintos algoritmos a datos temporales, se debe tener en cuenta el espacio de búsqueda con el que se va a trabajar. Los espacios discretos son aquellos que tratan con caracteres o patrones coincidentes. Diremos que

estamos en un ambiente contínuo cuando trabajamos con series temporales de números reales.

Una base de datos de series temporales será una larga colección de series temporales. Almacena secuencias de valores o eventos que cambian con el tiempo. Las bases de datos que recogen la variación de un stock serán un ejemplo. Los valores son normalmente medidos a intervalos iguales de tiempo. Las bases de datos de series temporales son muy utilizadas para estudiar las fluctuaciones diarias de los mercados de valores, realizar trazas de un proceso de producción dinámico, experimentos científicos, tratamientos médicos, etc.

Las bases de datos temporales se caracterizan por ser bases de datos muy grandes. Debido a estas dimensiones, el acceso a dichos datos o la aplicación de algoritmos de minería de datos se convierte en una tarea en la que es necesario invertir gran cantidad de tiempo. Cierto porcentaje de dichos datos nunca se estudiarán.

Una secuencia de eventos será, en cambio, una secuencia compuesta por una serie de símbolos nominales a partir de un determinado alfabeto. Una base de datos de secuencias es cualquier base de datos que registra secuencias de eventos ordenados con o sin noción concreta de tiempo.

Será necesario tratar los atributos relacionados con información temporal de manera que los datos se puedan ver como una secuencia de eventos para poder comprender completamente el fenómeno.

Tanto en (Antunes and Oliveira. 2001) como en (Vlachos. 2004) se ofrece una panorámica extensa del área de minería de datos referente a información temporal.

Es de destacar que el conocimiento a extraer a partir de datos temporales varía cuando disponemos de una sola secuencia o de múltiples secuencias. Cuando nos encontramos ante una sola secuencia el conocimiento que nos podría interesar extraer es el siguiente:

- Características importantes que describan a la secuencia.
- Patrones frecuentes y significativos.
- Periodicidades significativas.
- Cambios en el fenómeno que subyace en dicha secuencia y que se mostrará como una desviación significativa.
- Porcentaje de cambio a partir del comportamiento normal.
- Modelo completo que explique el fenómeno.

Cuando nos encontramos ante múltiples secuencias el conocimiento que nos interesa es:

- Clustering.
- Descubrimiento de reglas: "si el stock X sube e Y se mantiene Z bajará pronto".
- Clasificación de las secuencias
- Consultas por contenido.

El estudio de la similitud entre secuencias, y la consiguiente definición de una medida de la similitud o disimilitud, será la base para poder realizar muchas de las labores mencionadas, siendo el objetivo último el de descubrir relaciones entre secuencias y subsecuencias de eventos.

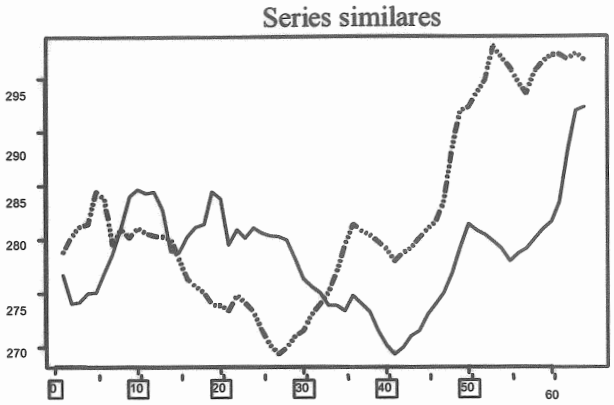


Figura 2. 2. Dos series temporales similares pero no idénticas

Cómo representar el tiempo es un problema fundamental que debe resolverse antes de que se aplique ningún algoritmo de los mencionados anteriormente. Una de las alternativas será una representación implícita, otra opción será la representación explícita.

La representación explícita asocia el orden consecutivo del patrón con la dimensionalidad del vector de patrones. El primer evento temporal está asociado al primer elemento del vector de patrones, el segundo con el segundo y así sucesivamente. Dicha aproximación tiene bastantes problemas, una de ellas es que necesita una interfaz con el mundo que almacene los datos de manera que se puedan presentar de golpe. El segundo de dichos problemas es que dichos registros imponen condiciones muy rígidas sobre la duración de los patrones y además sugieren que todos los vectores de entrada tienen la misma longitud. Es posible tratar con los datos después de que se hayan realizado transformaciones

mínimas, incluso podemos dejar la serie temporal en su forma original. Es posible utilizar una aproximación lineal a tramos para obtener subsecuencias manejables. Otra posibilidad es la de usar transformadas, de manera que podamos disponer de los datos representados en un espacio de representación más manejable. Dicho espacio de representación puede ser continuo o discreto.

La representación implícita es la que opta por dejar que el tiempo se represente por el efecto que tiene sobre el proceso. Se trata de dar al sistema dinámico procesado propiedades que le permitan ser sensible a secuencias temporales. En este tipo de aproximaciones, las representaciones internas tratan de conservar implícitamente el paso del tiempo. Se obtiene un modelo que puede ser visto como un generador de las secuencias obtenidas. Obtendremos un modelo determinístico o estadístico que pueda ser utilizado para resolver preguntas más complejas. En (Elman. 1990) se propone esta alternativa con el fin de dotar de memoria a modelos conexionistas, en concreto, se propone el uso de enlaces recurrentes en redes con el fin de dotar a las redes de memoria dinámica. Se han usado modelos semi-Markov (Ge and Smyth. 2000) para modelar la secuencia y encontrar un algoritmo que pueda calcular la distancia entre dos secuencias calculando la distancia entre los modelos que las generan. Otra alternativa es la de encontrar ordenaciones parciales entre símbolos de manera que sea posible encontrar episodios complejos formados por composiciones en serie o en paralelo de eventos básicos. Estos modelos pueden ser vistos como modelos basados en grafos desde el momento en el que se utilizan uno o más grafos para modelar las secuencias observadas.

Algoritmos de minería para datos temporales

Entre los algoritmos de minería de datos para series temporales, la clasificación no supervisada trata de encontrar agrupaciones naturales de series temporales correspondientes a una base de datos teniendo en cuenta cierta medida de la similitud/disimilitud. La clasificación no supervisada juega un papel importante en el proceso de extracción de información a partir de series económicas y financieras donde las series temporales tienen ruido. El clustering se ha utilizado en (Focardi. 2001-04.) como una herramienta econométrica para estudiar dependencias entre variables que pueden utilizarse para alguno de los siguientes fines:

- Identificar áreas o sectores con fines policiales como es el caso de la detección del fraude.
- Identificar similitudes estructurales en los procesos económicos para realizar predicciones económicas.
- Identificar dependencias estables para la gestión del riesgo y la gestión de las inversiones.

El clustering constituye un proceso de descubrimiento. El hecho de que dichos clusters se puedan descubrir, dependerá de los conjuntos de datos empíricos. Será necesario reflexionar posteriormente sobre la utilidad de los clusters descubiertos. El descubrimiento de clusters útiles depende tanto del algoritmo de clustering como de la distancia utilizada.

Múltiples trabajos, entre los cuales se encuentran (Bozkaya and Ozsoyoglu. 1997), (Bozkaya et al. 1997), (Han. 1999), (Ge. 2000), han estudiado distintos aspectos relacionados con los algoritmos de clustering.

En los algoritmos de clasificación, dada una serie temporal no etiquetada Q, se trata de asignar a dicha serie temporal una de dos o más clases predefinidas. Es necesario disponer de una base de datos previamente clasificada y estudiar la fiabilidad de la clasificación obtenida. En (Keogh. 1998) podemos encontrar referencias a distintos aspectos referentes a la clasificación.

La posible dependencia entre variables ha sido también estudiada usando para ello el coeficiente de correlación. Dicho coeficiente tendrá valores entre [–1 , 1]. Cuando el valor sea –1, sabemos que las variables se mueven en sentidos opuestos. Si el coeficiente de correlación es 1 sabemos que se mueven juntas.

El coeficiente de correlación puede ser interpretado como la existencia de cierta dependencia causal entre las variables o dependencia causal de las variables en cierto factor exógeno común.

Dicho coeficiente se ha usado en (Li et al. 1996). En otros trabajos ha sido también utilizado para comprobar si distintas medidas de la distancia definidas sobre un mismo conjunto de datos estaban correladas o no.

A continuación se plantea el problema de la predicción. Dado $x_{t-1}, x_{t-2}, x_{t-3} \ldots$, una serie de valores reales, se trata de predecir x_t.

Las predicciones lineales tratan de expresar x_t como una combinación lineal del pasado x_{t-1}, x_{t-2} tomando para ello una ventana de longitud w.

Formalmente podemos expresarlo según (1):

$$(1)\quad x_t = a_1 x_{t-1} + \ldots + a_n x_{t-w} + ruido$$

Dentro del área de la predicción, la regresión lineal trata de expresar lo que no conocemos (ó variable dependiente) como una función lineal de lo que sí sabemos (ó variable independiente). Por otra parte, la autoregresión lineal, usando una serie temporal $S[t]$, hace que ésta sea la variable dependiente y toma como variable independiente $S[t-1]$, suponiendo que la ventana de retraso utilizada es $w = 1$.

El análisis de periodicidad es la búsqueda de patrones recurrentes en bases de datos de series temporales. Pueden ser aplicados a áreas tan importantes como patrones de tráfico diarios, programaciones televisivas semanales, consumo de electricidad diaria, etc. Todos ellos presentan patrones periódicos. La extracción de éstos patrones periódicos puede ser vista como la tarea de extracción de patrones secuenciales tomando la duración como un conjunto de secuencias particionadas tales como "cada año" o "cada rebanada antes o después de la ocurrencia de cierto evento", etc.

El análisis de la periodicidad se puede dividir en tres categorías:

- Extracción de patrones periódicos completos: donde cada punto en el tiempo contribuye (precisa o aproximadamente) al comportamiento cíclico de la serie temporal. Por ejemplo, todos los días de un año contribuyen aproximadamente al ciclo estacional del año.
- Extracción de patrones periódicos parciales: especifica el comportamiento periódico de la serie temporal en algunos, pero no

todos, los instantes de tiempo. Serán patrones del tipo: "El lee el periódico de 7 a 7,30 de la mañana pero el resto de sus actividades diarias no presentan mucha regularidad".

- Extracción de reglas de asociación periódicas o reglas de asociación cíclicas: serán reglas que asocian un conjunto de eventos que ocurren periódicamente. Un ejemplo de reglas de asociación cíclicas es el siguiente: "Basándonos en las transacciones día a día, si el aperitivo de la mañana se toma entre las 12 y la 1 de la mañana, la comida será bien recibida, los fines de semana, entre las 14,30 y las15,30".

Se han aplicado técnicas para el análisis de periodicidad completa en el área de análisis de señal y la estadística. La Transformada de Fourier se ha utilizado para representar datos en el dominio de la frecuencia. Dicha transformada no puede ser aplicada a la tarea de extracción de patrones periódicos parciales porque trata la serie temporal como un flujo de valores inseparables. Lo cierto es que muchos de los métodos para la extracción de patrones periódicos completos son inaplicables o demasiado costosos para el proceso de búsqueda de patrones periódicos parciales La extracción eficiente de patrones periódicos parciales en bases de datos de series temporales ha sido estudiada en (Han. 1999). El mismo problema se aborda en (Han et al. 1998), considerándose el problema de la búsqueda de patrones periódicos parciales y no sólo la búsqueda de patrones periódicos completos.

Otro problema estudiado en la minería de datos es el problema de la detección automática de los puntos de cambio (Ge. 2000), (Zeira et al. 2004). A medida que

vamos acumulando datos en grandes bases de datos, es posible pensar que, de alguna manera, los nuevos datos actúan de la misma manera que el conocimiento previamente implementado. En realidad, sería necesario, para la mayoría de los algoritmos de aprendizaje incrementales, refinar o reconstrur los modelos almacenados a medida que lleguen nuevos datos.

Otro de los problemas de minería de datos que más interés ha despertado entre los investigadores en los últimos años es el proceso de búsqueda de la serie o series más similares a una dada. La problemática planteada en este apartado se divide en dos grandes grupos que se exponen a continuación.

Búsqueda de las series más similares a una dada con coincidencia completa, es decir, dada una colección de N objetos $O_a, O_b, ..., O_n$ y un objeto de consulta Q, queremos encontrar todos los objetos que estén dentro de una distancia ε de Q. La consulta y los objetos almacenados son del mismo tipo. Tratándose de secuencias, serían secuencias de la misma longitud. Las técnicas más utilizadas hasta el momento reducen la dimensionalidad de las series temporales, transformándolas. Se obtiene una representación en el espacio transformado y se utilizan posteriormente métodos de acceso espaciales, tales como el árbol R y sus variantes, en dicho espacio transformado.

La medida de la similitud $D(P',Q')$ definida sobre la base de una determinada representación P' y Q' para dos series temporales P y Q sería deseable que fuese una cota inferior de la distancia $D(P,Q)$ entre las dos series originales, de manera que $D_{LB}(P',Q') <= D(P,Q)$.

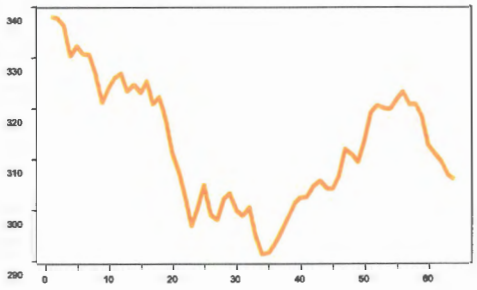


Figura 2. 3. Serie de consulta Q.

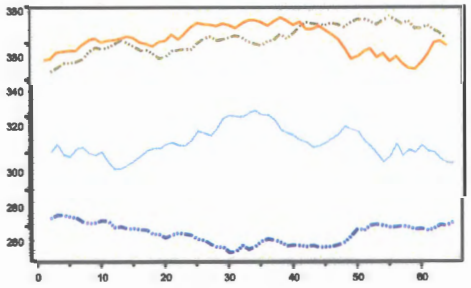


Figura 2. 4. Colección de 4 objetos del mismo tipo que la serie Q dada en la figura anterior.

Cuando se aborda el problema de los subpatrones coincidentes, se permite que la consulta especifique sólo parte del objeto. Dados N objetos $O_a, O_b, ..., O_N$, cada uno de ellos de longitud m, una consulta que busca subsecuencias coincidentes, dado una parte de un objeto Q y una tolerancia ε, debe identificar las partes de los objetos de datos que coinciden con la consulta. Se permite que la secuencia de consulta sea menor que la secuencia sobre la cual se pretende realizar la búsqueda. Se trataría de, dada una secuencia Q y otra secuencia más

larga O_i, de encontrar la subsecuencia en O_i, comenzando por el principio, que mejor coincida con Q y devolver su desplazamiento dentro de O_i. Ésto requiere que la consulta Q se sitúe en cada uno de los posibles desplazamientos dentro de la secuencia O_i. Si la longitud de la secuencia O_i es m y la longitud de la secuencia Q es h, la complejidad del algoritmo que realiza la búsqueda secuencial será $O(mh)$, suponiendo que $m >> h$. Esta opción será el escaneo secuencial y debemos considerarlo como aquella frontera superior o peor caso con el que comparar nuestro algoritmo. La mayoría de las técnicas de reducción de la dimensionalidad no pueden extenderse para que soporten el problema de la búsqueda de subsecuencias coincidentes.

Por otra parte, el proceso de recuperación de las series más similares a una dada, necesita saber el tipo de consulta a realizar. La consulta puede ser una consulta del vecino más próximo, una consulta por rango, todos las consultas de pares de vecinos más próximos, etc. Se han utilizado índices de búsqueda espaciales para poder responder rápidamente a este tipo de preguntas. El problema que presentan estos índices de búsqueda espaciales o multidimensionales es que, al trabajar con espacios de gran dimensionalidad, decrece la respuesta de la estructura de índices con lo que se ralentiza el cómputo de la distancia. Son, por lo tanto, ineficientes en el contexto de las series temporales donde la dimensionalidad es alta.

En (Faloutsos. 1994) se presenta el marco de trabajo GEMINI (Generic Multimedia Indexing Method) y las características generales que debe cumplir cualquier esquema de indexado que siga dicho marco de trabajo.

Los esquemas de indexado que usen dicho marco de trabajo requieren la consecución de los siguientes pasos:

- Establecimiento de una métrica para la distancia
- Utilización de un mecanismo de reducción de la dimensionalidad de manera que se reduzca la dimensionalidad de n a N con el fin de que podamos tratar las secuencias de longitud N con algún mecanismo de acceso multidimensional.
- Producción de una medida de la distancia en el espacio de representación N dimensional y demostrar que $D(F(S),F(T)) <= D(S,T)$, siendo $F(S)$ y $F(T)$ las representaciones correspondientes a las series S y T en el espacio de dimensionalidad reducida.
- Búsqueda de los vecinos más próximos a S en el nuevo espacio de representación.
- Cálculo de las distancias, para los vecinos más próximos, en el espacio de representación inicial y conservación de las más cercanas.

Si F es el mecanismo de reducción de la dimensionalidad, lo deseable es que las distancias sean iguales tanto en la representación original como en el espacio de las transformadas, es decir, que $D(F(S),F(T)) = D(S,T)$, aunque, a veces, esto

no será posible. Si la distancia es distinta en el espacio de transformadas, $D(F(S),F(T)) \neq D(S,T)$, pueden darse los siguientes casos:

- El cálculo de la distancia en el espacio de transformadas genera omisiones incorrectas. Se deben a objetos que no se pueden recuperar puesto que en el espacio de las transformadas se encuentran distantes, es decir, $D(S,T) << D(F(S),F(T))$. Dichas situaciones son, normalmente, inaceptables y para garantizar que no se produzcan, es necesario probar que $D(F(S),F(T)) < aD(S,T)$ para alguna constante a.
- El cálculo de la distancia en el espacio de transformadas genera inclusiones innecesarias: sucede cuando el objeto parece estar cerca en el espacio transformado pero se encuentra en la realidad distante. Se puede expresar como aquellos casos donde $D(S,T) >> D(F(S),F(T))$. Las inclusiones innecesarias pueden borrarse en un proceso posterior, confirmando las distancias en los datos originales. Se toleran si son relativamente poco frecuentes.

Utilizar dicho marco de trabajo aporta las siguientes ventajas:

- Las estructuras de índices trabajan mejor en espacios de dimensionalidad pequeña.
- El cálculo de la distancia se realiza antes.
- El tamaño de la base de datos se reduce, mejorando su tiempo de respuesta a las consultas.

Sería deseable asegurar la completitud del proceso de extracción o que el índice sea compacto, es decir, que garantice que no se produzcan omisiones incorrectas. En el caso de la Transformada Discreta de Fourier, el teorema de Parseval garantiza que la distancia entre dos secuencias en el dominio de la frecuencia es la misma que la distancia en el dominio del tiempo. En lo que respecta a los métodos de indexado multidimensional, los tiempos crecen exponencialmente para grandes dimensiones, quedándose reducidas, eventualmente, al escaneo secuencial. Además, se debe contemplar la posibilidad de realizar transformaciones sobre el índice original para sobrellevar consultas de longitud variable.

A la hora de realizar una valoración del algoritmo de indexado, en (Keogh. 2000a), se recuerda que el tiempo de ejecución puede ser dependiente de la implementación seleccionada. En dicho trabajo, se propone como unidad de medida "la fracción de la base de datos que ha de ser examinada antes de que podamos decir que hemos encontrado la secuencia más coincidente con nuestra consulta".

El cálculo del índice deberá ser posible de calcular en $O(n)$ o $O(nlogn)$ pero no $O(n^2)$. El algoritmo debe ser rápido a la hora de asociar un objeto nuevo a su imagen. Este algoritmo debería ser del orden $O(1)$ o $O(logn)$.

Dos de las transformadas más utilizadas son la Transformada Discreta de Fourier y la Transformada Discreta Haar Wavelet. Para ambas, la distancia Euclídea en el dominio del tiempo, entre dos series, es la misma que la distancia

Euclídea en el dominio transformado. Dentro de las transformadas dependientes de los datos está la descomposición en componentes principales.

La referencia (Faloutsos. 1996) corresponde a un breve pero ilustrativo libro de las técnicas utilizadas tanto en lo que respecta a las bases de datos como a los métodos correspondientes al área de teoría de la señal.

Reducir la dimensionalidad de los datos temporales es, por lo tanto, fundamental para realizar cualquiera de las labores mencionadas, tanto en la búsqueda de las series más similares a una dada, como en labores de clustering, clasificación, y en la búsqueda de reglas de asociación.

En el siguiente capitulo se explorarán un cierto número de técnicas relacionadas con la reducción de la dimensionalidad en datos temporales entre las cuales veremos las siguientes:

- SVD o descomposición en componentes principales (Korn et al. 1997).
- Transformadas discretas de Fourier (Agrawal. 1993), (Faloutsos et al. 1997), (Li et al. 1996), (Rafiei and Mendelzon. 1998), (Rafiei. 1999).
- Transformada Discreta Coseno (Le Gall. 1991)
- Transformadas Discretas Wavelets (King-pong Chan. 1999), (Kahveci and Singh. 2001), (Popivanov and Miller. 2002), (Shahabi et al. 2000), (Wang and Wang. 2000), (Wu et al. 2000), (Pei et al. 2001).
- FastMap (Faloutsos and Lin. 1995) y sus variantes.
- Transformada "linear predictive coding cepstrum".
- Aproximación lineal a tramos.

- Índice PCA o aproximación constante a tramos.
- Índice PAA o aproximación agregada a tramos.
- Índice APCA o aproximación constante adaptativa a tramos.
- Aproximaciones simbólicas.

La comparativa entre distintas técnicas de reducción de la dimensionalidad puede realizarse estudiando:

- Errores en la reconstrucción de las series temporales.
- El tiempo necesario para construir el índice multidimensional para distintas bases de datos y longitud de consulta distintas.
- La eficiencia del indexado, observado como el tiempo que le lleva encontrar la mejor respuesta a nuestra consulta. La mejor manera de medirlo puede ser medir el número de veces que tenemos que recuperar un ítem de disco.
- Características observadas, tales como el hecho de que X permita consultas sopesadas, consultas de longitud variable, que sea fácil de implementar, etc.
- Exactitud en la clasificación.
- Calidad de los clusters encontrados mediante algoritmos de clasificación no supervisada.

La reducción de la dimensionalidad de los datos puede producir una pérdida de información. Dicha pérdida se mide, en el caso de la búsqueda de similitudes, mediante dos medidas que a continuación se detallan.

Si denotamos con A_d un conjunto de puntos d-dimensionales y denotamos como A_k el conjunto de puntos en el espacio k-dimensional, donde $k \leq d$. Si $R(q, A_d)$ denota el conjunto de puntos que satisfacen una consulta en el espacio d dimensional y si $R(q, A_k)$ denota el resultado de la consulta en el espacio de k dimensiones, podemos definir dos medidas de la exactitud en la consulta de la siguiente manera:

$$precision(q,d,k) = \frac{|R(q,A_k) \cap R(q,A_d)|}{|R(q,A_k)|} \quad (2)$$

$$rellamada(q,d,k) = \frac{|R(q,A_k) \cap R(q,A_d)|}{|R(q,A_d)|} \quad (3)$$

Cuando la consulta devuelve los m vecinos, es decir, $|R(q,A_d)| = m$ y $|R(q,A_k)| = m$, las dos medidas anteriores son las mismas.

A continuación se presentan distintos métodos de indexado de bases de datos, empezando por los métodos de indexado de claves primarias, claves secundarias, métodos de acceso a puntos y métodos de acceso espaciales.

Según el modelo relacional, la información se organiza en tablas donde las filas de la tabla corresponden a registros. Si quisiéramos almacenar los datos de los clientes podríamos crear una tabla utilizando el siguiente comando SQL :

CREATE TABLE EMPLEADO (

Cod-emp# entero,

Nombre carácter(50),

Edad carácter (80),

salario entero,

antigüedad fecha)

El acceso a claves primarias (únicas) supone contestar a preguntas del tipo "encontrar el cliente con cod-emp=22" donde se asume que cod-emp no va a tener duplicados. Entre los mecanismos disponibles para el acceso a claves primarias en almacenamiento secundario podemos tener el hashing y los árboles B.

El acceso a claves secundarias, supone contestar eficientemente a consultas sobre cualquiera o incluso sobre todos los atributos disponibles. A continuación se muestra una posible clasificación de dichas consultas.

Consultas de coincidencia total: cuando la consulta especifica todos los valores deseados de los atributos. Por ejemplo, "encontrar los empleados cuyo nombre sea García, la edad sea 40 y el salario sea 3.000 € "

Consulta de coincidencia parcial: cuando se especifican sólo algunos de los atributos. Por ejemplo, "encontrar los empleados cuyo nombre sea García y el salario sea 3.000 € ".

Consultas de rango: donde se especifica el rango para alguno o todos los atributos. Por ejemplo, "encontrar los empleados cuyo salario sea mayor o igual que 3.000 € y menor que 5.000 €".

Consultas booleanas: serán aquellas consultas del tipo "encontrar los empleados cuyo nombre no sea Juan y el salario sea mayor o igual que 4.000 €".

Consulta del vecino más cercano o la mejor coincidencia: donde el usuario especifica algunos de los valores de los atributos y pregunta por la mejor coincidencia de acuerdo a una función distancia predefinida.

Algunos de los métodos usados en este apartado serán los ficheros invertidos y los llamados métodos de acceso a puntos (PAMS). Estos últimos métodos pueden ser también utilizados para acceder a objetos espaciales tales como rectángulos, polígonos, etc. Los árboles k-d serán métodos de acceso PAM y son, estructuralmente, árboles binarios donde cada nodo contiene un registro de datos, un apuntador izquierdo y un apuntador derecho. Divide el espacio de direcciones en regiones disjuntas mediante cortes, según dimensiones/atributos alternativos. En cada nivel del árbol, se utiliza un atributo distinto como discriminante, normalmente con alternancia round-robin.

Métodos de acceso espaciales

Los métodos de acceso espaciales están diseñados para soportar puntos, líneas, rectángulos y otras estructuras geométricas entendiendo que k atributos numéricos pueden ser entendidos como puntos en un espacio de dimensión k.

Disponiendo de un conjunto de objetos espaciales, las siguientes consultas pueden resultar de interés:

Las consultas de rango son, ahora, una generalización de las consultas de rango mencionadas anteriormente. Responden a preguntas del tipo 'encontrar todas las ciudades que estén a menos de 50 Km. de Bilbao' de manera que, se suministra una región y se consulta por todos los objetos que intersecan dicha región. Dada una secuencia, encontrará todas las secuencias similares a la dada dentro de una distancia ε. El parámetro ε, es un parámetro que controla cuándo dos secuencias pueden ser consideradas similares. Dicho valor podría determinarlo el usuario o podría determinarse automáticamente (10% de energía de la secuencia de consulta, por ejemplo).

Las consultas de punto son un caso especial de las consultas de rango donde la región queda reducida a un solo punto. Una consulta de rango podría devolver los objetos espaciales completamente contenidos en la región de consulta o podría devolver los objetos espaciales que contienen completamente la región de consulta.

Las consultas del vecino más cercano serán una generalización de las consultas del vecino más cercano para claves secundarias. Responden a preguntas del tipo "encontrar las 5 oficinas de correo más cercanas a nuestra empresa". En dichas consultas el usuario especifica un punto o una región y el sistema debe devolver los k objetos más cercanos utilizando para devolver dicha respuesta la distancia Euclídea o alguna otra función distancia.

Consultas espaciales o recubrimientos, que corresponderán a consultas del tipo: "encontrar todas las ciudades que estén dentro de un radio de 30 Km. de un parque natural". Dadas n secuencias, se trata de determinar todos los pares de secuencias que estén dentro de una distancia ε una de otra.

Los métodos que utilizan estructuras semejantes a árboles constituyen un gran grupo dentro de los métodos de acceso espaciales y se pueden entender como una extensión de los árboles-B para objetos multidimensionales.

Los árboles-R fueron propuestos en (Guttman, A. 1984). Es una estructura de datos arbórea basada en rectángulos de área mínima definidos por dos puntos y con sus caras paralelas a los ejes de coordenadas (MBR, Minimun Bounding Rectangles), balanceados en altura y almacenadas en disco. Dichos MBR se agrupan recursivamente en otros MBR´s y estos se organizan en un árbol multidimensional.

Los nodos que no corresponden a hojas son de la forma (*apuntador, R*) donde *apuntador* es un apuntador a un nodo hijo, y *R* es el MBR que cubre todos los rectángulos en el nodo hijo. Los nodos hoja contienen entradas del tipo (*identificador objeto, R*) donde el primero de los elementos del nodo contiene un apuntador a la descripción del objeto, y *R* es el MBR del propio objeto. En los árboles-R los nodos padre se pueden solapar, con lo que se garantiza un buen uso del espacio y permanece como árbol balanceado. Puede indexar tanto puntos como rectángulos.

En (Beckmann, Norbert 1990) se introdujo la idea de retrasar la división para la mejora de la utilización, de manera que, cuando se produce un desbordamiento en

un nodo, algunos de sus hijos se seleccionan cuidadosamente para ser borrados y posteriormente reinsertados para obtener, en la mayoría de los casos, un árbol-R mejor estructurado.

Los árboles-Vp y los árboles piramidales fueron estudiados en (Berchtold et al. 1998), (Bozkaya and Ozsoyoglu. 1997), (Bozkaya et al. 1997). Mediante esta aproximación se divide el conjunto de datos en lugar de particionar el espacio. En los árboles-Vp se particionan los puntos, a cada nivel, basándose en la distancia al centro. El escaneo secuencial recorre la base de datos una vez, calculando las distancias. Se puede optimizar suministrando cotas inferiores de la distancia. Es un método competitivo si la dimensionalidad es alta.

Para dimensionalidades menores que 10 las técnicas de particionado del espacio funcionan bien. Para grandes dimensionalidades el escaneo secuencial será probablemente la mejor técnica. Si nos encontramos en una situación intermedia el particionado de la base de datos funciona bien.

En (Salzberg and Tsotras. 1999) disponemos de una amplia comparativa de los distintos métodos de indexado.

En lo que respecta al problema de la búsqueda de subsecuencias coincidentes se han utilizado, entre otros métodos, los siguientes: ST-filter en (Bozkaya et al. 1997) y STB-indexing que usa la representación STB(shape to bit-vector). Permite un acceso rápido para búsqueda de similitudes en series temporales largas. El índice se forma creando cubos que contengan subsecuencias de la serie temporal de aproximadamente la misma forma. Para cada cubo podemos encontrar un límite inferior para la distancia entre la consulta y el elemento más

parecido del cubo. Esto permite la búsqueda en los cubos en orden 'primero el mejor' y realizar una poda de algunos cubos sin necesidad de explorar su contenido. Una optimización posterior impide que el contenido de cada cubo tenga que compararse íntegramente con la consulta.

Para construir el índice se hace uso de una ventana deslizante de tamaño fijo que se mueve a través de la secuencia de datos. La ventana contiene una malla que la divide en contenedores de un mismo tamaño. La sección de la serie temporal que cae dentro del mismo contenedor se discretiza en dos posibles clases 'up' y 'no-up'(1 y 0). Los patrones obtenidos leyendo la secuencia de izquierda a derecha y codificándola en 1's y 0's es lo que decide en que cubo (bin) se dejará el apuntador a dicha serie. Después de que todas las secuencias hayan sido copiadas a sus respectivos cubos, comparamos cada secuencia en un cubo con todas las demás secuencias en el mismo cubo y guardamos los resultados en una matriz de distancias.

Después de que el proceso termine, se dispone de una serie de cubos indexados por una secuencia de bits únicos. Cada cubo contiene una o más subsecuencias de los datos originales, junto con la matriz de distancias entre todo par de subsecuencias dentro del cubo. La matriz de distancias permite la poda en el espacio de búsqueda usando para ello la desigualdad triangular. El algoritmo hace uso de la siguiente estrategia para agilizar las comparaciones: si guardamos la mejor distancia hasta el momento podemos eliminar de la selección aquellos cubos que nunca podrían contener un candidato mejor que el obtenido hasta el momento.

Será posible podar un cubo sin necesidad de acceder a la página en disco donde residen los datos representados en el cubo. La ventaja es que la medida de la distancia puede ser calculada *k/K* veces más rápido de lo que podía calcularse en la serie original. El inconvenientes es que se ha de decidir de antemano el tamaño de la consulta, aunque podrían tratarse consultas de mayor o menor longitud. Para ello serían necesarios procesos adicionales. Se ha de decidir el tamaño de la malla, y el tamaño de las *b* (subdivisiones del mismo tamaño de la malla).

Podríamos también optar por construir índices de distinta longitud. Respecto a consultas de longitud superior a la malla, se puede entender que el contenido del índice será un prefijo de la secuencia buscada. Sería necesario chequear posteriormente si dentro de los recuperados se encuentra aquel que resulte la mejor secuencia coincidente con la de la consulta. Si no fuese el caso, se seguiría ampliando la región del espacio de estados hasta que así fuese.

Muchos de los mecanismos que se presentan en el siguiente capítulo, en el apartado de representación en el espacio de transformadas, serán esquemas que garantizan que no se produzcan omisiones incorrectas. Son los denominados esquemas aceptables.

Distancia entre series temporales

La implementación del algoritmo más sencillo de clasificación supervisada o no supervisada de series temporales, sería aquel que seleccionase una función distancia, por ejemplo la Euclídea, y la aplicase directamente sobre los datos. El

problema es que la distancia Euclídea ha resultado ser sensible a muchas de las perturbaciones que pueden sufrir las series temporales.

Dadas dos series temporales X e Y:

$$X = x_1, x_2, x_3, ..., x_n$$

$$Y = y_1, y_2, y_3, ... y_n$$

Denotaremos la distancia entre dichas series de la siguiente manera $D(X,Y)$. Se calculará la similitud-disimilitud entre las dos series según (4)

$$(4) \quad D(X,Y) = \sum_i \sqrt{(x_i - y_i)^2}$$

La ventaja que plantea es que es fácil de calcular y permite soluciones escalables para otros problemas como el indexado, clustering, etc. La desventaja es que no es capaz de dar una buena medida de la similitud entre series cuando éstas tengan diferencias de escala, por ejemplo cuando el stock X fluctúa entre 20 y 40 euros y el stock Y entre 180 y 190 euros.

En dichas situaciones, podemos normalizar la secuencia, de manera que, si tenemos que μ_X es la media y σ_X es la varianza de X obtendremos una nueva secuencia (5)

$$(5) \quad X'_i = \frac{(X_i - \mu_X)}{\sigma_X}$$

De esta manera, podemos tratar dicha secuencia en lugar de tratar la original. A pesar de ello, la medida de la similitud obtenida sigue siendo demasiado rígida puesto que no es capaz de tratar series con ruido o series con fluctuaciones a corto

plazo. Tampoco permite cambios de fase ni aceleraciones-desaceleraciones sobre el eje del tiempo. Hay una gran variedad de aplicaciones donde es importante poder formular preguntas en términos de similitudes de objetos en lugar de hacerlo en términos de igualdad o desigualdad. Además, la noción de similitud puede cambiar dependiendo del dominio de aplicación. A continuación, vamos a enumerar las propiedades deseables de cualquier medida de la distancia:

- Simetría: $D(A,B) = D(B,A)$
- Auto similitud constante: $D(A,A) = 0$
- Positiva: $D(A,B) = 0 \Leftrightarrow A = B$
- Desigualdad triangular: $D(A,B) \leq D(A,C) + D(B,C)$

Definida dicha medida de la similitud, podríamos responder preguntas del tipo: ¿Son semejantes los dos movimientos de stock observados?. La respuesta a dicha pregunta podría ser afirmativa sin que las series tengan que ser idénticas. A diferencia de las consultas que realizamos en las bases de datos donde se buscan los datos que coincidan exactamente con una consulta, una búsqueda de similitud busca secuencias de datos que difieran sólo ligeramente de una consulta dada. En relación con algunos de los métodos de indexado mencionados anteriormente, se han explorado también distancias basadas en rectángulos de área mínima (Perng et al. 2000) .

Otras aproximaciones, tal como la empleada en (Srikant and Agrawal. 1996), concibe la similitud entre dos series temporales de la siguiente manera: se divide la serie temporal en segmentos cortos, denominados envolturas. Se definen, a

continuación, dos envolturas como similares siempre que las diferencias punto a punto esté dentro de un cierto umbral. Se definía la similitud entre dos series temporales como el máximo número de envolturas consecutivas similares.

Antes de pasar a definir una medida de la similitud, hemos de considerar que las series temporales pueden estar sujetas a ciertas deformaciones que es preciso tener en cuenta. El preproceso de las series o la definición de la propia medida de la similitud dan la posibilidad de que series que reflejen estas deformaciones puedan ser consideradas similares,

Ciertos modelos de similitud utilizados (Chu and Wong. 1999) se han extendido en múltiples direcciones, considerando el escalado longitudinal o la distorsión en el eje temporal, puesto que muchas aplicaciones reales no requieren que las subsecuencias coincidentes estén perfectamente alineadas a lo largo del eje temporal. Es decir, podríamos permitir que pares de subsecuencias sean consideradas coincidentes si son de la misma forma, aunque difieran por la presencia de saltos dentro de una secuencia, o por diferencias en amplitud o en desplazamiento. En un modelo de similitud mejorado los usuarios o los expertos pueden modificar parámetros como el tamaño de la ventana deslizante, la anchura de una envoltura para la similitud, el máximo salto, el porcentaje de coincidencia, etc.

Hemos introducido el concepto de similitud. Describiremos, a continuación, situaciones en las cuales deberíamos llegar a la conclusión de que X y X' siguen siendo razonablemente similares: la translación, el escalado en amplitud , el crecimiento lineal, el escalado longitudinal y el ruido.

- Translación: será el caso en el que, por ejemplo, podamos encontrarnos ante dos fluctuaciones de stocks semejantes pero separados por una cantidad constante. La figura 2.5 es un ejemplo de dicha situación.

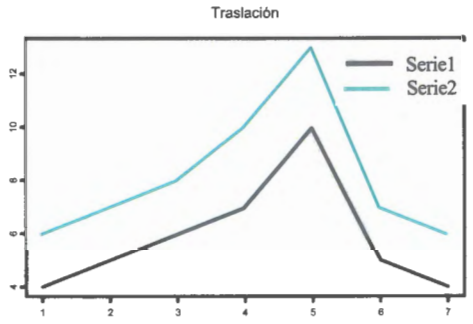


Figura 2. 5. Las series Serie1 y Serie2 presentan traslación

En este caso, si tenemos que μ_X es la media, obtendremos una nueva serie X' según la fórmula dada en(6):

$$(6) \quad X'_i = (X_i - \mu_X)$$

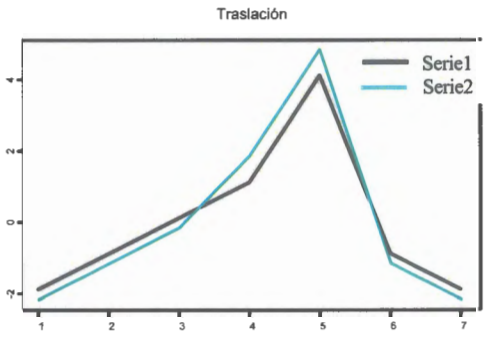


Figura 2. 6. Las series Serie1 y Serie2 de la figura 2.5 transformadas de manera que ambas tengan media 0.

- El escalado en amplitud:

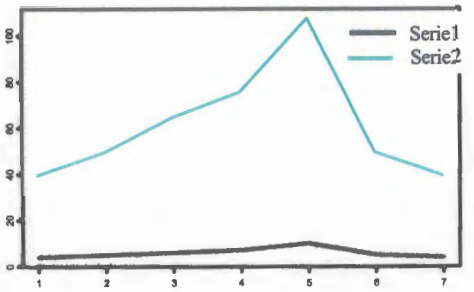


Figura 2. 7. Las series Serie1 y Serie2 presentan escalado en amplitud

Es posible eliminarlo normalizando la serie de manera que, si tenemos que μ_X es la media y σ_X es la varianza de X, obtendremos una nueva secuencia (7)

$$(7)\quad X'_i = \frac{(X_i - \mu_X)}{\sigma_X}$$

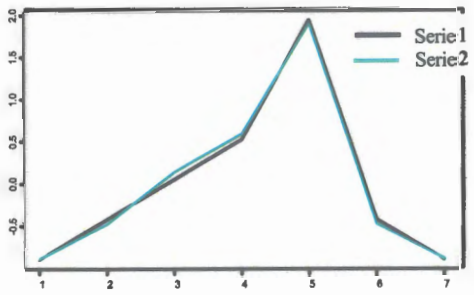


Figura 2. 8. Las series Serie1 y Serie2 de la figura 2.7 normalizadas

- Crecimiento lineal: este caso se presenta cuando, por ejemplo, disponemos de dos series que miden el nivel de venta de helados en dos poblaciones, sólo que una población se mantiene constante y la otra crece linealmente. Es posible borrar este tipo de tendencias eliminando de la serie la tendencia lineal observada .

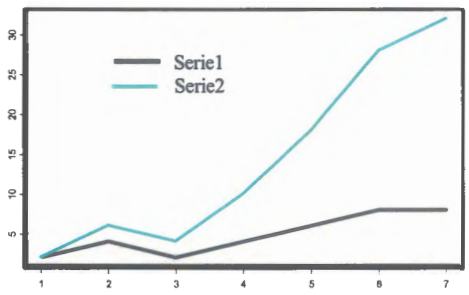


Figura 2. 9. Serie1 y Serie2 presentan escalado longitudinal

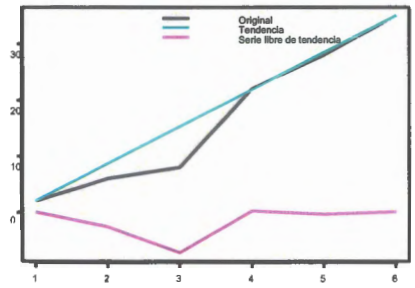


Figura 2. 10. La serie Serie2 de la figura 2.9, su tendencia lineal observada y la serie libre de tendencia.

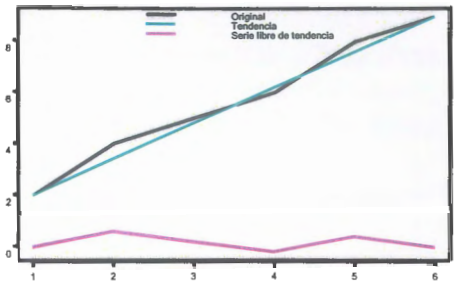


Figura 2. 11. La serie Serie1, su tendencia lineal observada y la serie libre de tendencia.

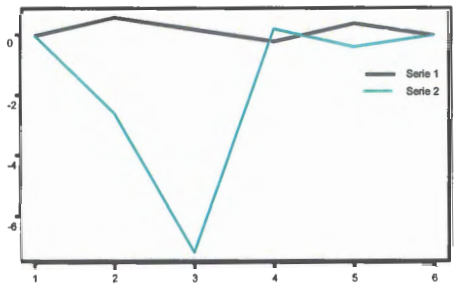


Figura 2. 12. Las series Serie1 y Serie2, libres de tendencia.

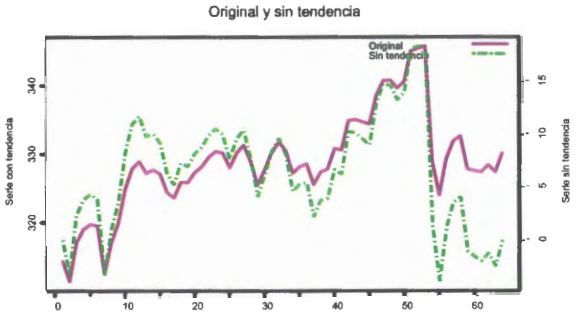


Figura 2. 13. Una serie y la misma serie sin tendencia mostradas en el mismo gráfico.

- Escalado longitudinal o distorsión en el eje temporal: sólo diferirán en el intervalo de tiempo empleado por cada una de ellas. Es posible tratarlo pero resulta más difícil que el escalado en amplitud. No se trata la serie previamente, dejando que sea la medida de la distancia utilizada la que considere la posibilidad de que una serie pueda encogerse o dilatarse con respecto al eje temporal sin que esto repercuta de una manera drástica en la similitud considera entre ambas.

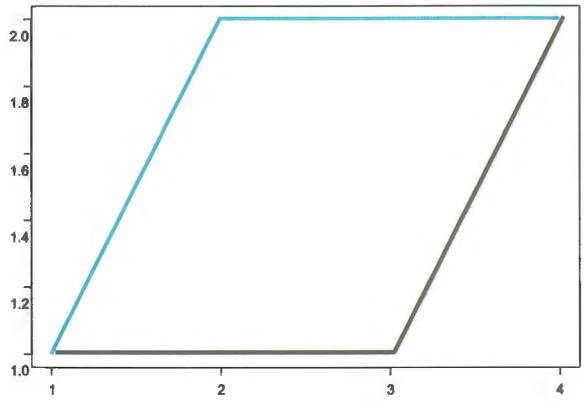


Figura 2. 14. Las dos series mostradas presentan escalado longitudinal

- Ruido: las series temporales pueden sufrir ruido que puede dificultar la comparación de series entre sí, para lo cual es preciso utilizar técnicas de suavizado que lo eliminen. Generalmente, el ruido se trata calculando la media de cada punto con sus vecinos.

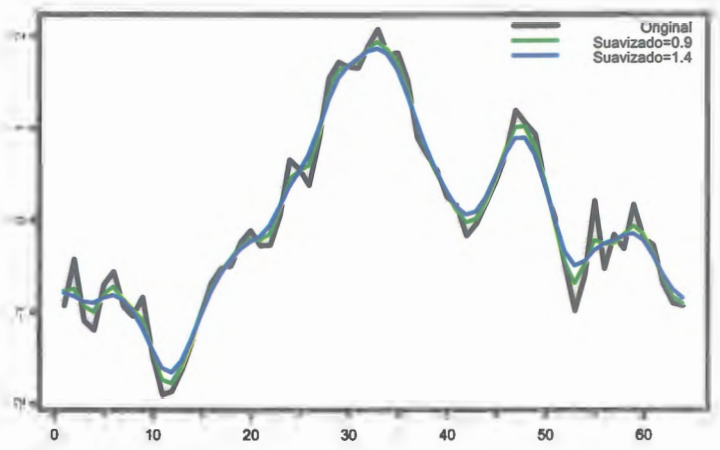


Figura 2. 15. Una serie y dos suavizados posibles de la misma.

CAPÍTULO 3

La descomposición espectral: La Transformada Discreta de Fourier

La primera de las técnicas usadas para la reducción de la dimensionalidad de las series temporales fue la Transformada Discreta de Fourier. La idea básica de la descomposición espectral es que cualquier señal, no importa lo compleja que sea, puede ser representada por la superposición de un número finito de ondas senos y/o cosenos. Cada onda es representada por un número complejo único conocido como el coeficiente de Fourier. Una serie temporal representada de esta manera se dice que está en el dominio de la frecuencia. La ventaja de representar la serie temporal en el dominio de la frecuencia es la compresión. Una señal de longitud n puede descomponerse en n ondas senos y cosenos que pueden recombinarse en la señal original. Pero el hecho cierto es que, muchos de los coeficientes de Fourier

tienen amplitudes pequeñas y contribuyen poco a la hora de reconstruir la señal. Estos coeficientes de baja amplitud pueden ser descartados sin demasiada pérdida de información produciendo, por lo tanto, compresión.

Para una serie $s = (s_0,....,s_{n-1})$ su DFT (Discrete Fourier Transform) será una secuencia $\vec{S}$ de n números complejos S_f, $f = 0,...,n-1$ dada de la siguiente manera (8):

$$(8)\quad S_f = \frac{1}{\sqrt{n}} \sum_{i=0,..,n-1} s_i e^{\frac{-j2\pi fi}{n}}, \quad f = 0,1,...,n-1, j^2 = -1$$

En la figura 3.1 se muestra una gráfica de la siguiente señal (9):

$$(9)\quad x_i = 6sen(\frac{2\pi 4i}{n} + 0.5) + 3sen(\frac{2\pi 8i}{n}) \quad i = 0,....,31$$

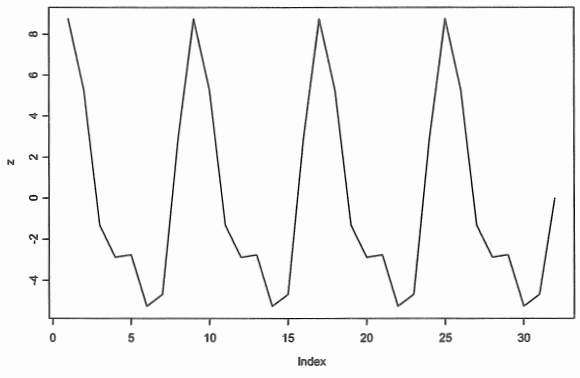


Figura 3. 1. Suma de funciones sinusoidales de frecuencias 4 y 8

El resultado de obtener el gráfico de $|S_f|$ respecto a f es el espectro de amplitudes. En la figura 3.2 se muestra el espectro de las amplitudes correspondientes a la serie mostrada en la figura 3.1, siendo la señal correspondiente la dada en la ecuación (9). Para aproximar la serie temporal nos quedamos con los primeros k coeficientes o bien con los más grandes.

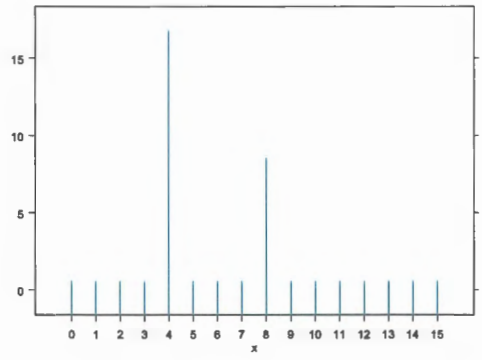


Figura 3. 2. Espectro de amplitudes para la serie mostrada en la figura anterior.

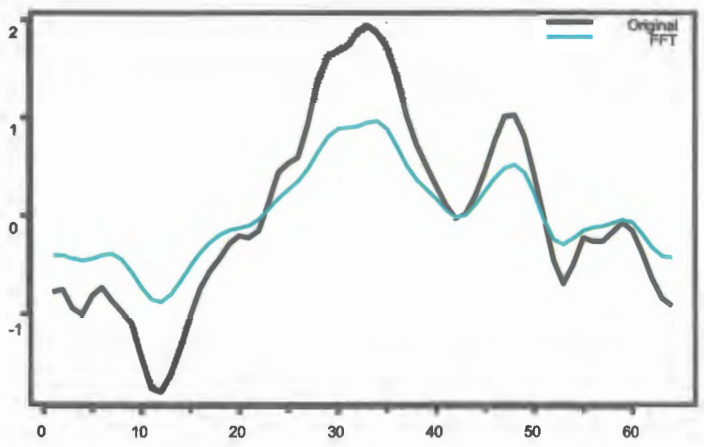


Figura 3. 3. Una serie temporal de longitud 64 y la serie obtenida usando para la reconstrucción los primeros 12 coeficientes de Fourier.

La observación clave es que la distancia Euclídea en el dominio del tiempo es preservada en el dominio de la frecuencia, tal y como se demuestra mediante el teorema de Parseval (10).

$$(10) \sum_{i=0,1,..,n-1} S_i^{\,2} = \sum_{i=0,...,n-1} S_f^{\,2}$$

Y además DFT es una transformada lineal con lo cual:

$$(11) - \sum_{i=0,1,..,n-1} (S_i - T_i)^2 = \sum_{i=0,1,..,n-1} (S_f - T_f)^2$$

La Transformada Discreta de Fourier concentra la energía de la serie en los primeros coeficientes. Si borramos el resto de los coeficientes podemos realizar el cálculo de la cota inferior de la distancia entre series. Dicha estimación de la distancia entre dos señales está garantizada que será una cota inferior de la distancia, puesto que estaremos descartando de dicho cálculo algunos términos positivos (12).

$$(12) \sum_{i=0,...,n-1} (S[i] - T[i])^2 > \sum_{i=0,..,k-1} (S_f - T_f)^2$$

En (Shatkay, Hagit 1995) disponemos de una breve y buena introducción a esta transformada.

La aproximación presentada en (Agrawal, Rakesh 1993) a la hora de generar el índice, el F-index, usa la Transformada Discreta de Fourier para dar solución al problema de la reducción de la dimensionalidad. La DFT será una transformada ortonormal que pasa la secuencia del dominio del tiempo al dominio de la frecuencia. Se considera útil al comprobar que es posible observar que una gran

cantidad de secuencias presenta grandes amplitudes para las primeras frecuencias. Se asume que las amplitudes grandes representan datos y las amplitudes pequeñas representan ruido. Usando las k primeras frecuencias, podemos esquivar el problema de la dimensionalidad y realizar el cálculo de una cota inferior de la distancia actual, aunque sea a costa de inclusiones innecesarias o falsos positivos. Estos falsos positivos se podrán borrar en un proceso posterior. El F-index será más favorable que el escaneo secuencial según se vaya incrementando el volumen de la base de datos. El problema que presenta es que se pierde la representación temporal y el índice se vuelve poco intuitivo. El orden del algoritmo de transformación para el F-index será $O(knlogn)$.

La generación del índice implica seleccionar el número de coeficientes a retener (frecuencia de corte) y la evaluación de cómo crece el tiempo empleado por el algoritmo en función del número de secuencias en la base de datos y la manera en la que la longitud de la secuencia afecta al comportamiento del algoritmo. Se determinó, en este primer trabajo, que un valor de la frecuencia de corte $\alpha = 2$ sería suficiente para poder indexar los resultados en un árbol R* de 4 dimensiones.

La transformada DFT ha sido criticada por ser considerada un método indirecto cuando, desde el punto de vista del descubrimiento del conocimiento, se trataría de encontrar una representación más directa que sea capaz de captar mejor la forma de la secuencia.

SVD: La transformada Loeve-Karhunen

Es una transformada global (Korn et al. 1997) donde se examina toda la base de datos como una matriz y luego se rotan los ejes de manera que el primero de los ejes tenga la máxima varianza posible, el segundo de los ejes tenga la máxima varianza posible siendo ortogonal al primer eje, el tercer eje tenga la máxima varianza posible siendo ortogonal a las dos anteriores, y así sucesivamente. La idea básica es la de representar el tiempo como una combinación lineal de ondas principales pero reteniendo sólo los N primeros coeficientes.

Para una matriz cuadrada S de orden $n \, x \, n$, el vector unidad $\vec{x}$ y el escalar λ que satisfacen la siguiente expresión (13), son los llamados vectores principales y correspondientes valores propios de la matriz S.

$$(13) \; S \, x \, \vec{x} = \lambda x \vec{x}$$

Los vectores principales de una matriz simétrica son mutuamente ortogonales y sus autovalores son reales.

Se puede entender una matriz S como una transformación afín $\vec{y} = Sx\vec{x}$ que conlleva una rotación y/o escalado. Los vectores principales serían los vectores unidad a lo largo de las direcciones que no son rotadas por S y sus correspondientes autovalores muestran el escalado. La formulación matemática de dicha transformada es la siguiente:

Siendo A la matriz $M \times n$ de M series temporales de longitud n, la SVD de A es la dada en (14):

$$(14)\ A = UxLxV^T$$

U es una matriz cuyas columnas son ortogonales entre sí y de orden $M \times r$, siendo r el rango de la matriz A. V será una matriz de orden $n \times r$ cuyas columnas serán ortogonales. L será una matriz diagonal que contiene los valores principales o autovalores de $A^T A$.

El método consiste en usar la SVD de la matriz A expresada según (15), truncando la expresión de manera que se retengan los k primeros términos $(k \leq r \leq M)$ tal como se muestra en (16):

$$(15)\ A = \sum_{i=1}^{r} \lambda_i u_i v_i^t$$

$$(16)\ A = \sum_{i=1}^{k} \lambda_i u_i v_i^t$$

En (Leach.) disponemos de una buena introducción a dicha transformada. La SVD será la transformación que minimice el error de reconstrucción pero el problema que plantea es que se trata de un algoritmo que necesita un tiempo de orden $O(Mn^2)$ y orden $O(Mn)$ en espacio.

Además, una inserción en la base de datos implica la reconstrucción de todo el índice. En (Kanth, K. V. Ravi et al. 1998) se usa SVD como mecanismo de reducción de la dimensionalidad en bases de datos dinámicas. El mecanismo presentado propone el recalculo de la transformada SVD de la matriz de datos sólo cuando la distribución de los datos cambie sensiblemente. Para recalcular la

transformada se propone el uso de agregados existentes en el propio índice en lugar de usar los datos completos.

Los coeficientes retenidos serán conocidos como los *k* componentes principales.

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	1	1	0	0
[2,]	2	2	2	0	0
[3,]	1	1	1	0	0
[4,]	5	5	5	0	0
[5,]	0	0	0	2	2
[6,]	0	0	0	3	3
[7,]	0	0	0	1	1

Figura 3. 4. Matriz de datos A

Autovalores

9.64	5.29	0.00	0.00	0.00

V

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.58	0.00	0.00	-0.82	0.00
[2,]	-0.58	0.00	-0.71	0.41	0.00
[3,]	-0.58	0.00	0.71	0.41	0.00
[4,]	0.00	-0.71	0.00	0.00	-0.71
[5,]	0.00	-0.71	0.00	0.00	0.71

U

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.18	0.00	0.08	0.98	0.02
[2,]	-0.36	0.00	0.92	-0.14	0.01
[3,]	-0.18	0.00	-0.09	-0.07	0.52
[4,]	-0.90	0.00	-0.37	-0.13	-0.11
[5,]	0.00	-0.53	0.00	0.00	0.71
[6,]	0.00	-0.80	0.00	0.00	-0.43
[7,]	0.00	-0.27	0.00	0.00	-0.14

Figura 3. 5. Descomposición de la matriz de datos *A*, dada en la figura 3.4, usando SVD.

Transformada Discreta Coseno

La fórmula correspondiente a la Transformada Discreta Coseno es la mostrada en (17):

$$(17)\ X_f = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} x_i \cos \frac{\pi f (i+0.5)}{n} \quad f = 0,....,n-1$$

Siendo la transformada inversa la mostrada en (18)

$$(18)\ x_i = \frac{1}{\sqrt{n}} X_o + \frac{2}{\sqrt{n}} \sum_{j=1}^{n-1} X_f \cos \frac{\pi f (i+0.5)}{n} \quad i = 0,...,n-1$$

Para muchas señales reales, los valores sucesivos están correlados. En estas condiciones DCT, la Transformada Discreta Coseno, logra mejores concentraciones de energía que la Transformada Discreta de Fourier.

Transformada Wavelet

Las wavelets u ondículas son funciones matemáticas que representan datos u otras funciones en términos de sumas y diferencias de una función prototipo llamada la onda madre. En este sentido, es similar a la Transformada Discreta de Fourier, pero difiere en aspectos importantes. La Transformada de Fourier está basada en la observación de que cualquier señal puede representarse como superposición de ondas seno y coseno. Las wavelet u ondículas se pueden entender como una generalización de dicha idea a un conjunto de funciones más amplio que el seno y el coseno. Otra de las grandes diferencias es que las ondículas están localizadas en el tiempo. Decir lo anterior es lo mismo que decir que algunos de los coeficientes representan subsecciones pequeñas y localizadas de los datos que están siendo estudiados a diferencia de los coeficientes de la transformada de Fourier donde representan contribuciones globales a los datos. Dicha propiedad es muy útil para el análisis multiresolución. El análisis multiresolución representa una función usando para ello un número finito de aproximaciones sucesivas. Se pueden utilizar Wavelets para aproximar una serie temporal reteniendo sólo los primeros k coeficientes y aproximando el resto con 0. Si estos coeficientes son los primeros k equivaldría a aplicar un filtro paso bajo. En finanzas o en economía, las ondículas han sido ampliamente utilizadas

(Gençay et al. 2002), (Goffe. 1994). También como herramientas en la minería de datos (Huang and Yu. 1999), (Han et al. 2003), (Stollnitz et al. 1995), (Struzik and Sibes. 1999), (Vlachos et al. 2003).

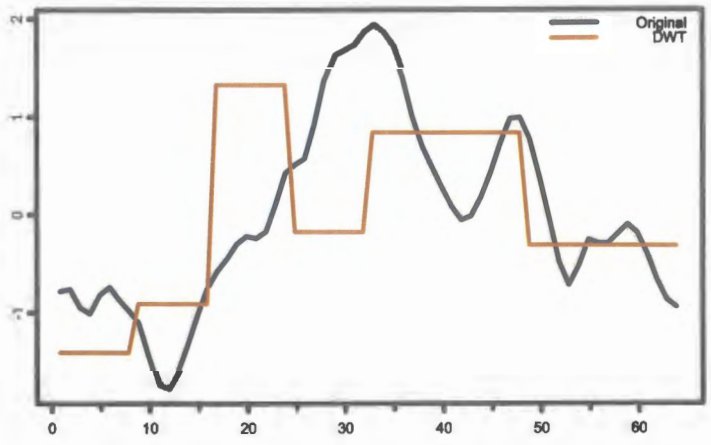


Figura 3. 6. Serie de longitud 64 reconstruida con los primeros 12 coeficientes de la Transformada Haar Wavelet.

Los primeros coeficientes contienen una aproximación general, burda, de los datos. Según vamos añadiendo coeficientes podemos tener la sensación de estar haciendo zoom sobre los datos a áreas de más detalle. El análisis multiresolución es el marco general para la construcción de bases ortonormales.

Por definición, un análisis multiresolución ortonormal de *L2* se genera mediante una función escalado y es una secuencia de subespacios encajados de manera que satisfaga lo siguiente:

$\ldots V-2 \subset V-1 \subset V0 \subset V1 \subset V2 \subset \ldots$

- $Vi \subset L2(R), i \in Z$
- $\cup_{i \in Z} V_i = L2(R)$
- $\cap_{i \in Z} V_i = \{0\}$
- $f(.) \in V_0 \Leftrightarrow f(2i.) \in V_i$ que es la condición multiresolución y que determina que V_i corresponda a una resolución más fina en el momento en el que i se incremente.
- $f(.) \in V_i \Leftrightarrow f(.-j) \in V_i$ lo cual determina que traslaciones enteras de cualquier función en el espacio deben permanecer en el espacio.
- las translaciones $\phi_{i,j}(x) = 2^{i/2}\phi(2ix - j)$ forman una base ortonormal para V_i.
- para todo $j \in Z$, denotamos como W_j el complemento ortogonal del espacio Vj en $Vj+1$

$$.V_{j+1} = V_j \oplus W_j,$$

para $j \neq j'$ y denotando de la siguiente manera $W_i \perp W_j$ que los espacios W_i y W_j son ortogonales, de manera que, cada vector en W_i es ortogonal a cada vector en W_j tendremos, por lo tanto, que $W_j \perp W_{j'}$ y $V_i \perp W_j$

Si extendemos la anterior definición con la propiedad de que cada W_i es generado por translaciones $\psi_{i,j}$ de la función ψ mostrado en (19), estaremos hablando de Wavelets.

$$(19)\ \psi_{i,j} = 2^{i/2}\psi(2ix - j)$$

Esta última propiedad asegura que W_i y las correspondientes $\psi_{i,j}(x)$ satisfagan:

- W_i es multiresolución: $f(.) \in W_0 \Leftrightarrow f(2i.) \in W_i$
- $\forall i \in W_i$ es invariante a translaciones: $f(.) \in W_i \Leftrightarrow f(. - j) \in W_i$
- para $i \neq k$, W_i y W_k son ortonormales :
- todas las funciones L2 se representan como una única suma:

$$L2 = \oplus Wi$$

Si esta propiedad se cumple, obtendremos una base ortonormal. La base estará compuesta por las traslaciones y escalados de la función wavelet ψ, de manera que, los coeficientes Wavelet de una función f se definen como los coeficientes de f con respecto a dicha base:

$$W_{ij}(f) = \langle f, \psi_{j,k} \rangle$$

Esta condición puede relajarse. De hecho, la mayoría de las wavelets usadas en la práctica no son ortonormales. La condición necesaria y suficiente para una reconstrucción estable es que la energía de los coeficientes Wavelet esté entre dos fronteras positivas (20).

$$(20)\ A\sum_k |X[k]|^2 \leq \|x\|^2 \leq B\sum_k |X[k]|^2$$

La condición expresada anteriormente garantiza una reconstrucción estable. Garantiza que cada función tenga una única representación en términos de coeficientes Wavelet. Cuando $A = B$ estaremos ante una base ortonormal.

La función Haar es el ejemplo más simple de transformada wavelet. Forma parte de la familia Daubechies. La función de escalado para la wavelet Haar es la función caja dada en (21) y su gráfica la mostrada en la figura 3.7 .

$$(21)\ \phi = \{1, si\ 0 < t < 1 \quad y\ 0\ en\ caso\ contrario\}$$

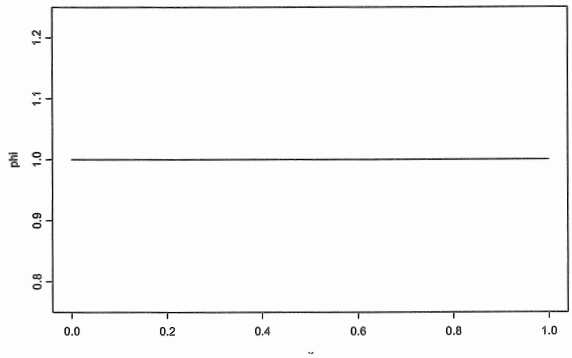


Figura 3. 7. Función de escalado para Haar Wavelet

La función madre ψ , cuyas traslaciones y dilataciones diádicas son la base para los subespacios contenidos W_i del análisis multiresolución, es ,en el caso del wavelet Haar, la mostrada en (22) y su gráfica se puede observar en la figura 3.8.

(22) $\psi = \{1 \text{ si } 0 < t < 0{,}5, -1 \text{ si } 0{,}5 < t < 1 \text{ y } 0 \quad en \quad caso \quad contrario\}$

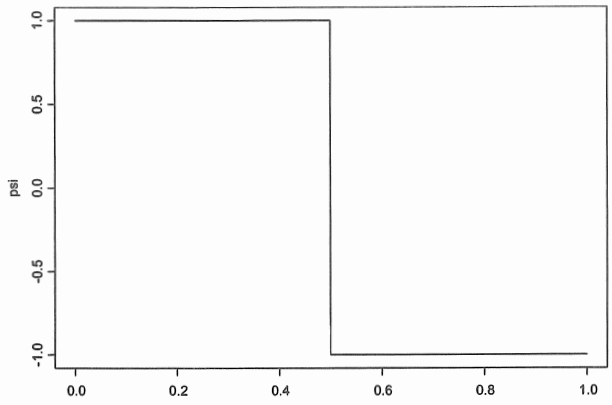


Figura 3. 8. Función madre para Haar Wavelet

Todos los wavelets se agrupan por familias y cada familia lleva el nombre de su creador. Los elementos de una familia se distinguen unos de otros por la longitud de los filtros, por ejemplo Coiflet 4 hace referencia a un wavelet de la familia Coiflet con los dos filtros de longitud 4.

En la figura 3.9 y en la figura 3.10, respectivamente, se muestran la función madre y función escalado para wavelets de la familia Daubechies con longitud de filtro 10.

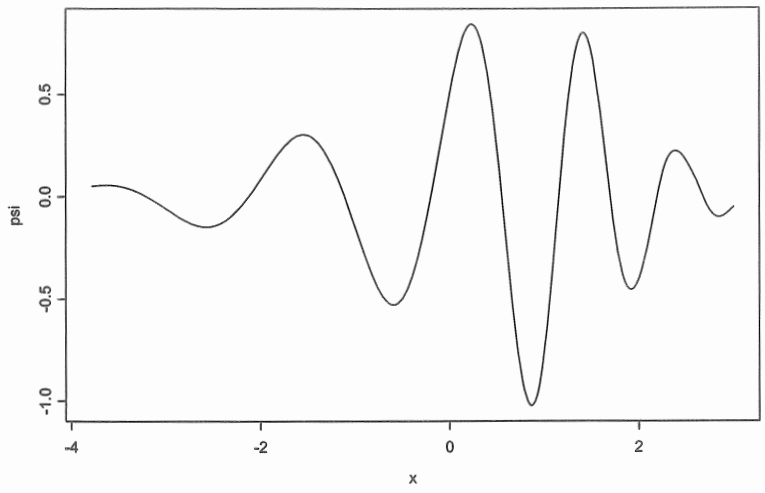


Figura 3. 9. Función madre para Daubechies 10

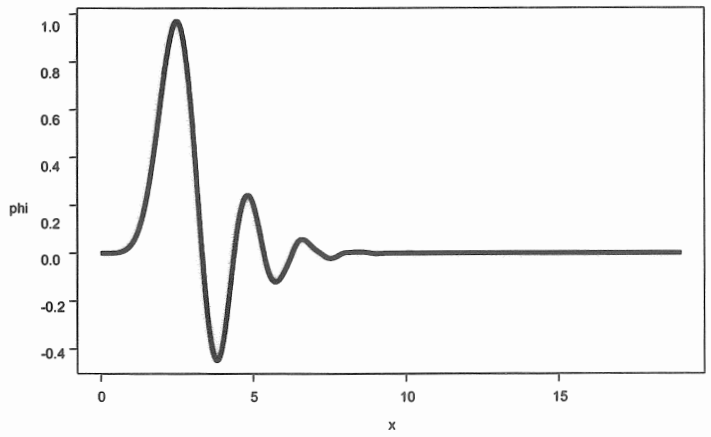


Figura 3. 10. Función escalado para Daubechies 10

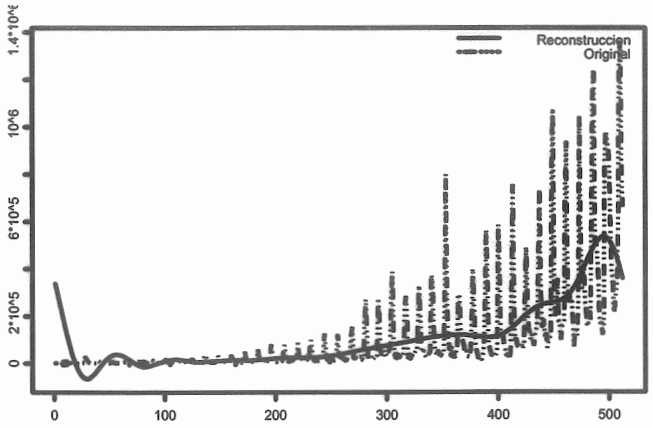


Figura 3. 11. Serie original y reconstruida con daubechies10 usando 16 coeficientes

Si considerásemos el siguiente vector de cuatro dimensiones $y = [2,5,2,7]$,su representación, usando wavelets, vendría dado por $y = Wc$ donde W contiene los vectores de base Haar, $W = [\phi_{00}\psi_{00}\psi_{10}\psi_{11}]$ de manera que el vector y se puede representar según (23)

$$(23)\quad \begin{pmatrix} 2 \\ 5 \\ 2 \\ 7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{pmatrix} \begin{pmatrix} c_0 \\ c_{00} \\ c_{10} \\ c_{11} \end{pmatrix}$$

La inversa de esta matriz de vectores ortogonales es igual a su transpuesta dividida por 4. (En general la inversa de una matriz de vectores ortogonales A con dimensión n es αA^t con $\alpha = |A|^{-\frac{2}{n}}$). La solución para los coeficientes wavelet viene entonces dada por la expresión (24):

$$(24)\begin{pmatrix} c_0 \\ c_{00} \\ c_{10} \\ c_{11} \end{pmatrix} = \frac{1}{2^2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix}\begin{pmatrix} 2 \\ 5 \\ 2 \\ 7 \end{pmatrix} = \begin{pmatrix} 4 \\ -\frac{1}{2} \\ \frac{3}{2\sqrt{2}} \\ \frac{5}{2\sqrt{2}} \end{pmatrix}$$

La transformada wavelet tiene la ventaja de que descompone los datos en diferentes escalas o diferentes niveles de refinado. El vector del ejemplo consiste en tres escalas c_0, c_{00} y $c_1 = c_{10}c_{11}$. Si hiciésemos el último nivel igual a cero $c_1=0$, y multiplicándolo por *W*, obtendremos que el vector de entrada es $[\frac{7}{2}\frac{7}{2}\frac{9}{2}\frac{9}{2}]$, es decir, se hace la media de los primeros dos elementos y la media de los dos últimos elementos. En procesamiento de la señal esto es lo mismo que aplicar un filtro paso bajo.

Si lo que hiciésemos fuese poner los dos últimos niveles a cero, el vector resultante sería $[4\,4\,4\,4]$, la media del vector. Si hiciésemos que todos los coeficientes excepto c_{00} sean 0 e invertimos la transformada, el resultado será el vector $[-\frac{1}{2}-\frac{1}{2}\frac{1}{2}\frac{1}{2}]$, la diferencia entre la media y el segundo nivel de refinado, $[\frac{7}{2}\frac{7}{2}\frac{9}{2}\frac{9}{2}]$ Finalmente, si pusiésemos todos los coeficientes a cero salvo $c_{10}\,c_{11}$, la transformada inversa daría el vector $\left[-\frac{3}{2}\frac{3}{2}-\frac{5}{2}\frac{5}{2}\right]$, la diferencia entre el segundo nivel de refinado y los datos originales.

Podemos a continuación utilizar la descomposición wavelet para representar el vector y como la suma de los componentes suavizados S_2 y los componentes detalles D_2 y D_1 según se muestra en(25):

$$(25)\ y = S_2 + D_2 + D_1 = \begin{pmatrix} 4 \\ 4 \\ 4 \\ 4 \end{pmatrix} + \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} + \begin{pmatrix} -\frac{3}{2} \\ \frac{3}{2} \\ -\frac{5}{2} \\ \frac{5}{2} \end{pmatrix}$$

Calcular las expansiones wavelet a partir de la inversión de matrices es computacionalmente costoso. A mediados de los 80 Mallat introdujo métodos de la teoría de procesamiento de la señal a los wavelets. Utilizando una técnica denominada "Quadrature Mirror Filtering" mostró que cualquier transformación discreta wavelet puede implementarse utilizando un algoritmo tipo cascada. Este mecanismo permitió que el número de operaciones requeridas para la transformada se redujese del $O(nlogn)$ al orden de $O(n)$, haciéndolo más rápido que la transformada rápida de Fourier.

El análisis multiescala usa filtros para dividir funciones $fn \in V_n$ en diferentes componentes que pertenecen a los subespacios V_{n-i} $(i = 1,2,,,n)$ y a sus complementos ortogonales (26):

$$(26)\ f_n = f_{n-1} + g_{n-1} = \sum_{i=1}^{M} g_{N-M} + f_{N-M}$$

Cada uno representa una escala distinta de la función. Es posible entender los subespacios V_i como diferentes niveles de aumento que revelan más y más detalles. Dichas funciones presentan autosemejanza de manera que (27) :

$$(27) \quad f(x) \in V_j \leftrightarrow f(2x) \in V_{j+1}, j \in Z$$

En el lenguaje de procesamiento de la señal, se obtienen las funciones f_{n-1} y g_{n-1} aplicando a la función original f_n, un par de filtros compuestos por dos componentes separados, filtro paso alto y filtro paso bajo, que se corresponden con salidas con frecuencia alta y baja respectivamente.

Para obtener la transformada wavelet en cada paso, la señal se pasa a través del par de filtros. La salida consiste en la componente paso alto y la componente paso bajo donde cada una de ellas es, de longitud, la mitad de la longitud de la entrada. En cada paso se conserva la componente paso bajo y se vuelve a utilizar como entrada.

En un dominio discreto, los filtros mencionados anteriormente son un par de secuencias $\{h(k)\}$ y $\{g(k)\}$ para $k \in Z$, donde $\{h(k)\}$ es un filtro paso bajo, $\{g(k)\}$ es un filtro paso alto y los dos filtros están conectados mediante la siguiente relación (28):

$$(28) \quad g(n) = (-1)^n h(1-n)$$

Cada wavelet tiene una función escalado o función padre que expande el subespacio V_0 y puede ser representado como una combinación lineal de funciones del siguiente subespacio, V_1 . Puesto que los subespacios son

autosemejantes y encajados, existe relación entre las funciones escalado de cualquier par de subespacios vecinos V_j y V_{j+1}. Dicha relación es la denominada ecuación escalado o ecuación dilatación (29) y define los coeficientes del filtro.

$$(29)\ \phi(x) = \sum_{k \in Z} h_k \sqrt{2}\phi(2x-k)$$

Los dos filtros son correspondencias de *l(Z)* a *l(2Z)*, de manera que, transforman un vector de *n* elementos en dos vectores con $\frac{n}{2}$ elementos cada uno, donde uno de ellos conserva los datos suavizados por el filtro paso bajo y el otro contiene los detalles borrados.

Cada wavelet puede caracterizarse por un conjunto finito de coeficientes para dichos filtros, derivados de la ecuación de escalado, que relaciona entre sí las funciones de escalado de diferentes subespacios V_i.

Para el Haar wavelet los coeficientes de los filtros son los siguientes (30):

$$(30)\ h(1) = h(2) = \frac{1}{\sqrt{2}} \text{ y } g(1) = \frac{1}{\sqrt{2}}, g(2) = \frac{-1}{\sqrt{2}}$$

Si utilizásemos el mismo vector que hemos utilizado anteriormente *f(2) = [2 5 2 7]* los vectores filtrados vienen dados por (31)

$$(31)\ f(1) = \left[\frac{7}{\sqrt{2}} \frac{9}{\sqrt{2}}\right] \text{ y } g(1) = \left[\frac{-3}{\sqrt{2}} \frac{-5}{\sqrt{2}}\right]$$

La función *f(1)* es una media sopesada de las primeras dos y siguientes dos entradas de *f(2)* respectivamente, donde los coeficientes del filtro se están utilizando como pesos y donde se utiliza el mismo procedimiento para *g(1),* sólo

que en este caso, uno de los coeficientes del filtro es negativo, con lo que la media sopesada es ahora una diferencia que recorta detalle de *f(1)*.

Es conveniente usar los operadores *H* y *G* para denotar la relación de los filtros aplicados a una secuencia $a = \{a_n\}$ según (32) (33) :

$$(32)\ (Ha)_k = \sum_n h(n-k)a_n$$

$$(33)\ (Ga)_k = \sum_n g(n-k)a_n$$

Si la señal original fuese *c(n),* un vector con *2n* elementos, entonces $c(n-1) = Hc(n)$ y $d(n-1) = Gc(n)$. Si aplicamos el filtro paso bajo dos veces, tendremos que $c(n-2) = H2c(n)$ y $d(n-2) = HGc(n)$.

Usando el análisis multiresolución, la Transformada Discreta Wavelet de una secuencia $y = c(n)$ de longitud *2n,* es otra secuencia de igual longitud dada por (34)

$$(34)\ \begin{matrix} w = \left[d^{(n-1)}, d^{(n-2)}, d^{(n-3)}, ..., d^{(1)}, d^{(0)}, c^{(0)}\right] = \\ \left[Gy, GHy, GH^2 y, ..., GH^{n-1} y, GH^n y, H^n y\right] \end{matrix}$$

Para invertir la transformada wavelet se aplica un procedimiento de filtrado inverso. Los operadores $\hat{H}$ y $\hat{G}$ establecen una correspondencia de *l(2Z)* a *l(Z)* y cada elemento se dobla y se multiplica por los coeficientes wavelet. Si aplicamos los filtros inversos paso alto y paso bajo a los vectores *f(1)* y *g(1)* tendremos (35)

$$(35)\ \hat{H}f(1) = \left[\frac{7}{2}\ \frac{7}{2}\ \frac{9}{2}\ \frac{9}{2}\right], \hat{G}g(1) = \left[\frac{-3}{2}\ \frac{3}{2}\ \frac{-5}{2}\ \frac{5}{2}\right]$$

Sumando las dos expresiones reproducimos el vector inicial *f(2)*. Gráficamente podemos verlo de la siguiente manera, la serie de entrada *S(n)* se descompone en partes detalle $cD_1(n)$ y parte suavizada $cA_1(n)$ utilizando filtros paso alto (HiF_D) y paso bajo (LoF_D) respectivamente. La descomposición obtenida es una descomposición en primera escala. Es posible reconstruir la señal original a través de los coeficientes de la aproximación y los detalles. Es también posible reconstruir las propias aproximaciones y detalles a través de sus vectores de coeficientes. El vector de coeficientes $cA_1(n)$ pasa por el mismo proceso que cuando se usa para reconstruir la señal pero en lugar de combinarlo con $cD_1(n)$ se usa un vector de ceros en su lugar. Con ello obtenemos una aproximación reconstruida A_1 de la serie original *S(n)*.

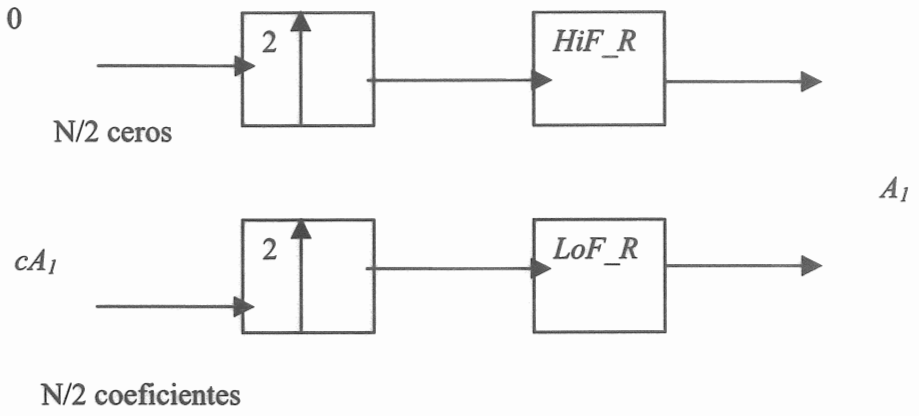


Figura 3. 12. Reconstrucción de la aproximación a primera escala

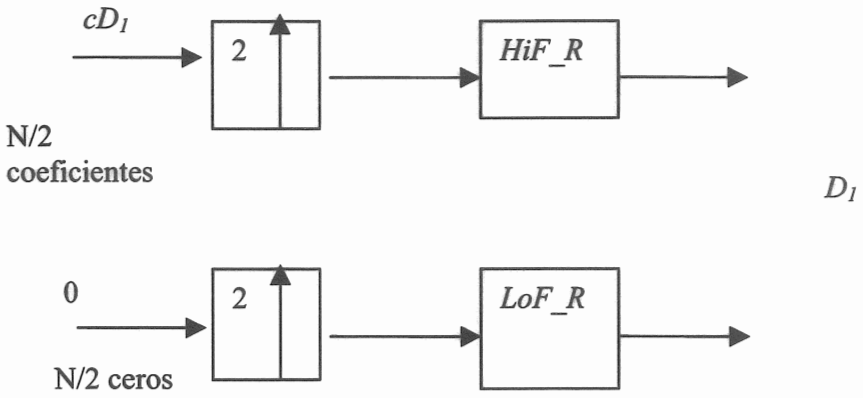


Figura 3. 13. Reconstrucción del detalle a primera escala

En las siguientes figuras es posible apreciar las sucesivas reconstrucciones para una serie dada.

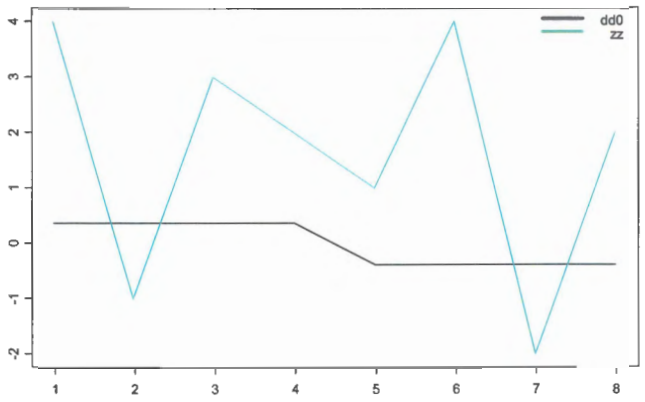


Figura 3. 14. Reconstrucción de la serie *zz* al primer nivel.

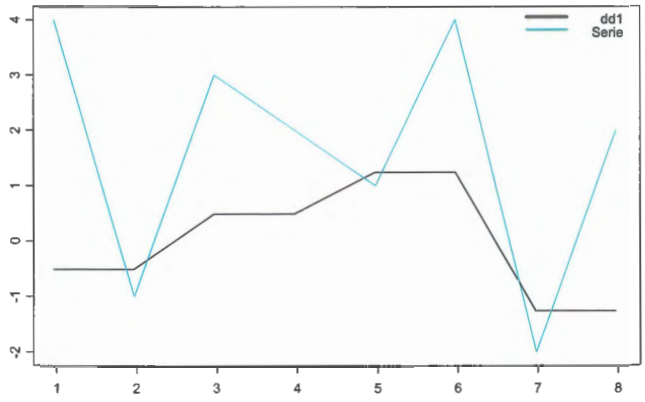


Figura 3. 15. Reconstrucción de la serie *zz* al segundo nivel.

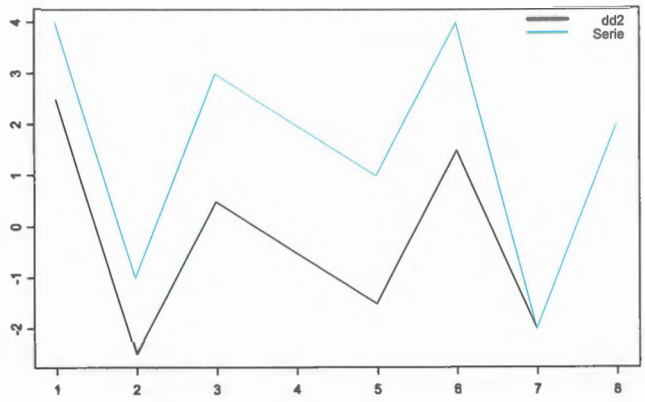


Figura 3. 16. Reconstrucción de la serie *zz* al tercer nivel.

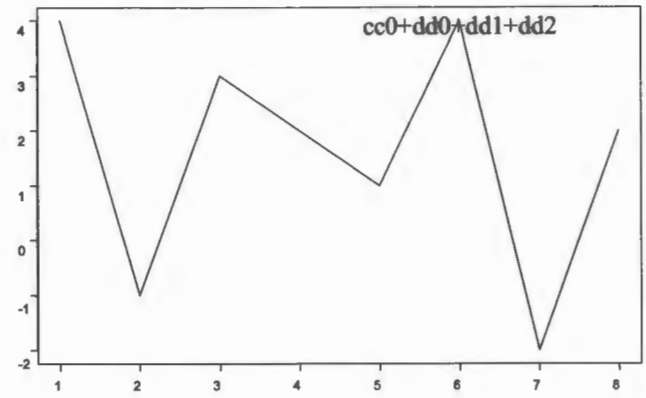


Figura 3. 17. Reconstrucción de *zz* como suma de todos los niveles.

Los detalles y las aproximaciones reconstruidas son los que constituyen la señal original de manera que $A_1 + D_1 = S$.

Las Transformadas Wavelet presentan soporte compacto, esto es, las funciones base son distintas de cero sólo en un intervalo finito, de manera que las señales

pasan de estar en el dominio del tiempo a estar en el dominio de la frecuencia y del tiempo, y no sólo en el de la frecuencia, como sucede con la transformada de Fourier. El tiempo necesario para obtener la transformada es un tiempo lineal con respecto a la longitud de la señal de entrada. El conjunto de funciones base puede ser infinito, en el caso de la transformada de Fourier las funciones utilizadas son el seno y el coseno.

Para la definición de una medida de la distancia en el espacio de transformadas que garantice que no se produzcan descartes se ha de cumplir la propiedad contractiva para dicha distancia de manera que (36):

$$(36)\ D_{transformado}\ (T(A),T(B)) <= D_{real}\ (A,B)$$

Dicha propiedad es la que garantiza que un F-index no genere descartes. Dicha propiedad será cierta para cualquier transformada lineal ortonormal y la Haar wavelet la cumple.

En (King-pong Chan. 1999) basándose en una ondícula simple pero poderosa, la función Haar wavelet, se genera una medida de la distancia que satisface los requerimientos de ser una cota inferior para la distancia. La Transformada Discreta Wavelet presenta ciertos problemas, está sólo definida para secuencias cuya longitud sea una potencia entera de dos.

En (Struzik and Siebes. 1999) se definen otra serie de representaciones derivadas de la transformada Haar. Entre ellas la representación basada en signos (37), la cual utiliza sólo el signo del coeficiente wavelet de manera que si $c_{i,j}$ son los coeficientes wavelet.

$$s_{i,j} = \text{sgn}(c_{i,j})$$

(37) $$\text{sgn}(x) = \begin{cases} 1 & para \quad x \geq 0 \\ -1 & para \quad x < 0 \end{cases}$$

Si tenemos las series $x = (3,\ 1,\ 0,\ 2,\ -3,\ -4,\ 1,\ 2)$ e $y = (2,\ 3,\ 1,\ -3,\ -4,\ 1,\ 0,\ 1)$, las transformadas correspondientes serán $x_t = (0.25,\ 1.25,\ 0.50,\ -2.50,\ 1.00,\ -1.00,\ 0.50,\ -0.50)$ e $y_t = (0.125,\ 0.625,\ 1.750,\ -1.000,\ -0.500,\ 2.000,\ -2.500,\ -0.500)$ y sus representaciones usando el signo de los coeficientes transformados serán los siguientes: $x' = (1,\ 1,\ 1,\ -1,\ 1,\ -1,\ 1,\ -1)$ e $y' = (1,\ 1,\ 1,\ -1,\ -1,\ 1,\ -1,\ -1)$.

Otra medida de la similitud planteada en dicho trabajo es (38), la diferencia de los logaritmos de los coeficientes wavelet en la escala superior y en la escala actual.

(38) $$v_{i,j}^{\ DOL} = \log(|c_{i,j}|) - \log(|c_{1,1}|)$$

Donde i, j son respectivamente la escala actual y la posición y $c_{1,1}$ es el primer coeficiente de la representación wavelet.

Esta representación puede normalizarse (39) para obtener el porcentaje de incremento de v^{DOL} con la escala.

(39) $$h_{i,j} = \frac{v_{i,j}^{\ DOL}}{\log(2^{(i)})} \quad para \quad i > 0$$

Si tenemos las series $x = (3, 1, 0, 2, -3, -4, 1, 2)$ e $y = (2, 3, 1, -3, -4, 1, 0, 1)$, las transformadas correspondientes serán $x_t = (0.25, 1.25, 0.50, -2.50, 1.00, -1.00, 0.50, -0.50)$ e $y_t = (0.125, 0.625, 1.750, -1.000, -0.500, 2.000, -2.500, -0.500)$ y su representación normalizada será la siguiente: $x' = (1.609438, 0.500000, 2.821928, 0.5, 0.5, 0.000000, 0.0)$ e $y' = (1.609438, 3.307355, 2.500000, 0.5, 1.5, 1.660964, 0.5)$.

La Transformada Wavelet es útil porque es más fácil liberar a las señales de ruido en el dominio de las ondículas que en el dominio de la transformada de Fourier, y además, es posible liberar de ruido a señales con saltos. En (Wu et al. 2000) se realiza una comparativa entre DFT y DWT como herramientas para búsqueda de similitudes en bases de datos de series temporales.

Las wavelet bi-ortonormales se pueden usar en un entorno F-index usando la propiedad contractiva para la distancia en el espacio real y el espacio transformado. En la práctica, la mayoría de los wavelets utilizados en el área de compresión de la imagen pertenecen al grupo de wavelets bi-ortonormales. En (Popivanov.2001) se pueden leer los detalles que el autor aporta sobre los cambios que es necesario aplicar sobre una consulta de rango para que puedan emplearse wavelet bi-ortonormales.

La característica multiresolución de los wavelet ha sido explotada en recientes trabajos (Vlachos et al. 2003) donde se propone una versión para el algoritmo de clustering k-medias, de manera que, inicialmente se realiza un clustering basado

en una representación burda de los datos y dichos resultados son utilizados posteriormente para inicializar un clustering a un nivel más fino de representación. Dicho proceso continúa hasta que los resultados se estabilicen o la representación utilizada sean los datos originales. Otros trabajos (Huhtala et al. 1999) han centrado su atención en series alineadas o sincronizadas, de manera que, no se toma en cuenta la posibilidad de que exista un traslación a lo largo del tiempo. Para ello, definen una medida de la similitud invariante respecto a la posición vertical, el escalado o la tendencia, basada en características extraídas a partir de la transformada wavelet.

Las propiedades multiresolución han sido también explotadas en (Shahabi et al. 2000) con el objetivo de agilizar consultas multinivel sobre la tendencia o sobre patrones sorprendentes construyendo una estructura de árbol denominada árbol TSA para agilizar el proceso de recuperación y proceso de consultas. Dicho árbol será de un tamaño superior al conjunto de datos originales, pero posteriores definiciones de subárboles óptimos denominados árboles OTSA, consiguen que el espacio de almacenamiento necesario no sea superior al de las series originales. En dicho trabajo se proponen otras dos alternativas para el escalado multidimensional. Será utilizado para descubrir la estructura espacial subyacente de un conjunto de ítems de datos a partir de las distancias entre ellos. Para ello, representa el conjunto de ítems en el plano Euclideo k-dimensional, de manera que, minimiza las diferencias existentes en las distancias entre dichos ítems.

Las entradas al algoritmo serán M series temporales, las distancias dos a dos para dichas series temporales y la dimensionalidad deseada. El algoritmo

establecerá una correspondencia entre cada objeto y un punto en el espacio k dimensional de manera que se minimice una función de "stress" que será la definida en (40).

$$(40)\quad \sqrt{\frac{\sum_{ij}\left(\hat{D}(S_i,S_j)-D(S_{ki},S_{kj})\right)^2}{\sum_{ij}D(S_i,S_j)^2}}$$

En esta función, $D(S_i,S_j)$ es la distancia entre las series S_i y S_j y $D(S_{ki},S_{kj})$ la distancia Euclídea en la representación k-dimensional. La medida obtenida dará el error relativo que, como media, sufre la distancia en el espacio k dimensional. El algoritmo comienza con una asignación a una serie temporal de un punto en el espacio k-dimensional, donde calcula la distancia de dicho punto a los otros $M\text{-}1$ puntos y va moviendo el punto con el objetivo de minimizar la discrepancia entre las disimilitudes en el espacio real y las disimilitudes en el espacio k dimensional. El algoritmo necesita un tiempo $O(N^2)$, siendo N el número de ítems.

Fast Map

FastMap (Faloutsos and Lin. 1995), es una aproximación rápida al escalado multidimensional. Establece correspondencias entre objetos y puntos k-dimensionales preservando las distancias. La idea clave es pretender que los objetos sean puntos en cierto espacio n-dimensional desconocido y tratar de proyectar dichos puntos en k direcciones mutuamente ortogonales. El problema

reside en calcular dichas proyecciones a partir de la matriz de distancias, puesto que es la única entrada disponible. El algoritmo consta de los siguientes pasos:

- Selecciona dos objetos lejanos.
- Proyecta todos los puntos en la línea que los dos objetos definen para conseguir la primera coordenada.
- Proyecta todos los objetos en un hiperplano perpendicular a la línea que los dos objetos definen.
- Repetir *k-1* veces.

Transformada "linear predictive coding cepstrum"

En (Rabiner and Juang. 1993) se pueden consultar detalles sobre dicha transformada. En (Kalpakis et al. 2001) se presenta una medida de la distancia basada en esta transformada, demostrando que dichos coeficientes tiene las características deseadas para un clustering exacto y un indexado eficiente de series temporales ARIMA.

Para una serie temporal AR(p), los coeficientes de la transformada pueden derivarse de los coeficientes de la autoregresión según (41):

$$(41)\ c_n = \begin{cases} -\alpha_1 & si \quad n = 1 \\ -\alpha_n - \sum_{m=1}^{n-1} (1 - \frac{m}{n}) \alpha_m c_{n-m} & si \quad 1 < n \leq p \\ -\sum_{m=1}^{p} (1 - \frac{m}{n}) \alpha_m c_{n-m} & si \quad p < n \end{cases}$$

Una de las ventajas de esta transformada es que serán necesarios pocos coeficientes para discriminar entre series temporales que están modeladas por distintos modelos ARIMA.

La gran diferencia de esta transformada con respecto a las anteriores estriba en que modela la serie temporal sobre la base de su componente tendencial, componente cíclica, componente estacional y componente aleatoria. Estos cuatro componentes es posible capturarlos con un modelo estacional Box-Jenkins o ARIMA.

El concepto de similitud que se plantea es el siguiente: dos series temporales son similares si los modelos físicos subyacentes que los generan son los mismos o están cercanos. Si los parámetros de los modelos ajustados a las series temporales son similares, las series temporales se comportan de una manera semejante. Se extraerán los coeficientes cepstrales para extraer características relevantes de la serie temporal, y se calculará la distancia entre series temporales como la distancia entre los coeficientes cepstra.

Los coeficientes de esta transformada decaen rápidamente a cero con lo que pocos coeficientes son suficientes para retener la mayoría de la información de la serie temporal.

Aproximación lineal a tramos

Representa la serie temporal aproximándola mediante k segmentos lineales. Los segmentos pueden estar conectadas, en cuyo caso, para cada segmento debemos conservar la longitud y la altura izquierda. La altura derecha se puede

obtener mirando en el nuevo segmento. Si N es el número de coeficientes de la nueva representación, podremos guardar información de $\frac{N}{2}$ líneas. Los segmentos pueden estar desconectados, en cuyo caso, para cada segmento, se habrá de calcular su longitud, la altura izquierda y la altura derecha. Si N es el número de coeficientes de la nueva representación podremos guardar información de $\frac{N}{3}$ líneas.

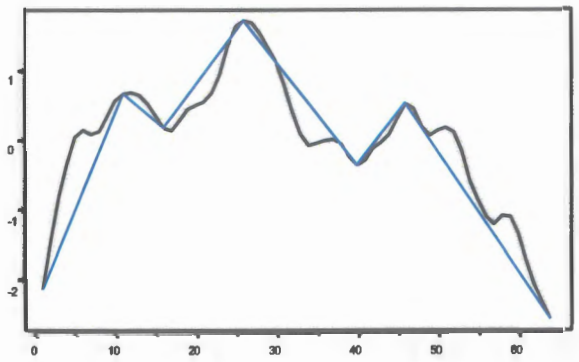


Figura 3. 18. Representación aproximada de una serie usando segmentos lineales continuos.

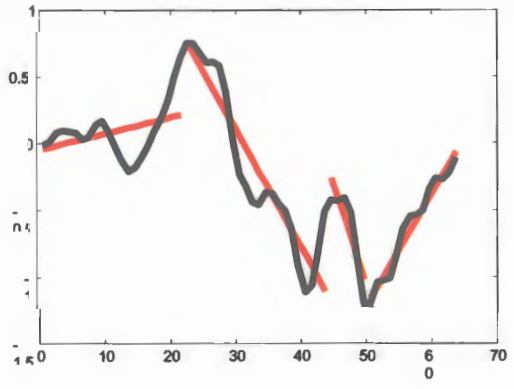


Figura 3. 19. Representación aproximada de una serie usando segmentos lineales discontinuos.

Este tipo de representaciones permiten la compresión de los datos pero el problema reside, básicamente, en la selección del valor de la *k*. Si éste es demasiado pequeño se pierden demasiadas características de la serie temporal y si es demasiado grande se está conservando información redundante. La medida de la similitud entre series representadas de esta manera puede ser la suma de las diferencia entre los segmentos proyectados a lo largo de las dos series. Esta medida de la similitud podría sopesarse, dando más importancia a los segmentos deseados. Una posible aplicación de dicha representación es la detección de los puntos de cambio, donde lo que se desea es localizar los puntos en los que esté teniendo lugar un cambio significativo en el comportamiento de la serie.

Hay trabajos que han adoptado dicha representación como representación subyacente, entre ellos, (Keogh. 1998), (Keogh. 1997c) . Una ventaja adicional de este método es que reduce el impacto del ruido, en cambio, los problemas de escalado, o la existencia de saltos o distorsiones sobre el eje del tiempo no son problemas que se resuelven fácilmente. Es posible hacer uso de un vector de pesos que conservará información sobre la importancia relativa de cada segmento lineal. La información sobre cada segmento será una quíntupla donde tenemos dos valores para indicar el punto inicial, otros dos valores para determinar el segundo de los puntos, y un quinto valor que arrastrará la longitud del intervalo. La selección de los *k* segmentos que aproximan la serie se hace, en este trabajo, mediante el algoritmo presentado por el propio autor en (Polly and Wong. 2001).

Con el fin de particionar la secuencia, existen dos posibilidades, definir de antemano el número de segmentos, o descubrir primero dicho número y pasar

luego a identificar los segmentos correspondientes (Das et al. 1997), (Guralnik and Srivastava. 1999), (Das et al. 1997), (Guralnik and Srivastava. 1999).

Si disponemos del número de segmentos k, tenemos el problema de obtener los segmentos que mejor se ajustan al valor dado. Es posible usar un función de error como mecanismo de búsqueda de la segmentación óptima. Este problema se ha estudiado en (Keogh. 1997b). Formalmente, la problemática de segmentación de una serie temporal es la siguiente, dada una serie temporal Q con n puntos, construir un modelo $\overline{Q}$ a partir de k segmentos (k<<n) de manera que $\overline{Q}$ aproxime Q. La segmentación pude ser utilizada para determinar cuando cambia el modelo que creó la serie temporal (Gavrilov et al. 2000), (Ge and Smyth. 2000) o puede ser utilizada para generar una representación de más alto nivel de la serie temporal que soporte indexado, clustering y clasificación (Keogh. 1997b), (Li et al. 1998), (Park et al. 2001.), (Polly and Wong. 2001.).

En (Keogh. 2001) se parte la serie en segmentos lineales a tramos donde cada segmento tiene un peso representando su importancia, aprendiéndose dichos pesos a partir de un conjunto de entrenamiento, planteando un ajuste de pesos semiautomático.

(Guralnik and Srivastava. 1999) obtienen una partición de la serie temporal en un conjunto de segmentos lineales y no lineales, desconociendo inicialmente el número de segmentos, la posición de los puntos de corte, y la función que describe cada segmento. En (Smyth and Keogh. 1996) se plantea el algoritmo que aprende automáticamente los k mejores segmentos lineales, entendiéndose la búsqueda de dicho valor como un compromiso entre la exactitud y la compresión.

Podemos encontrar más detalles al respecto en (Keogh. 1997a). El criterio para denominar una segmentación como la mejor es considerar que el proceso de segmentación preserva los patrones más significativos para el dominio de aplicación. Se enumeran, en dicho trabajo, las propiedades que con tal fin se habrían de cumplir: consistencia, robustez, etc.

En (Hsu and Ray. 1998) se da una breve descripción de las aproximaciones usadas por los algoritmos más comunes para la segmentación. Uno de los mecanismos más fáciles de aplicar es "iterative end-points fits", tiene el problema de que es sensible a valores desperdigados. Para evitarlo podría aplicarse un proceso previo que, mediante el uso de estimadores suaves, hagan que la curva sea menos dependiente de valores puntuales y a su vez sirvan como preproceso para eliminar el ruido.

Planteado el problema de la segmentación óptima, en (Shatkay and Zdonik. 1996) se han estudiado algoritmos on-line basados en ventanas deslizantes, interpolando un polinomio sobre él y partiendo la secuencia siempre que se desvíe significativamente del polinomio. Se estudian también algoritmos off-line que se usan para completar las secuencias.

En el último trabajo mencionado, se habla del concepto de consultas aproximadas generalizadas como una situación en la que:

- Hay un patrón independiente de los valores que caracteriza el resultado deseado.
- La consulta denotaría un conjunto S de secuencias, en lugar de una secuencia simple.

- *S* sería cerrado ante transformaciones que preserven el comportamiento y no simplemente mediante la identidad de ciertos valores.

El resultado es una coincidencia exacta si la secuencia resulta pertenecer a *S*. Se aborda el problema de la representación de una manera más general, soportándose consultas generalizadas relacionadas con características más comunes, dependientes de la aplicación, en lugar de realizar la consulta por valores muestreados. El acercamiento utilizado para resolver el problema es el de usar funciones con valores reales como alternativa de representación aproximada. En este tipo de problemas no importa tanto las consultas por valores exactos como la forma de algunas subsecuencias. El método se sustenta en la capacidad para poder realizar la división de los datos de entrada en regiones que permitan ser aproximadas por una función. Dependiendo del dominio en el que nos encontremos, por ejemplo, como en este caso, en el dominio de la medicina, los polinomios han sido suficientes.

Índice PCA o aproximación constante a tramos

El índice PCA presentado en (Keogh and Pazzani. 2000) se basa en la observación de que, para muchas series de datos, podemos obtener una aproximación segmentando las series en secciones de igual longitud y guardando las medias de dichas secciones.

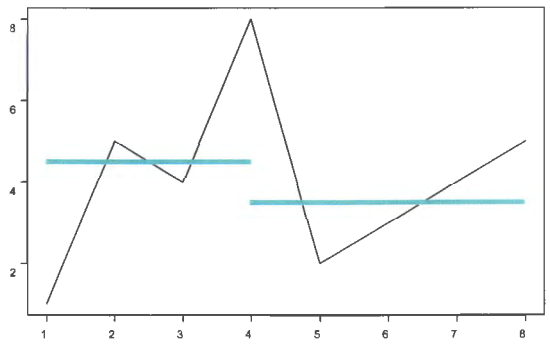


Figura 3. 20. Una serie representada mediante una aproximación constante a tramos.

Será útil como compresor de la imagen y filtrado de ruido (Keogh. 1999a). En el mencionado trabajo se compara dicho método a aquel índice obtenido mediante la representación DFT. El índice se puede construir en tiempo lineal y permite medidas de la distancia más flexibles, por ejemplo, la distancia Euclídea sopesada. El orden del algoritmo para transformar una secuencia será *O(n)*, para construir el índice completo será de *O(kn)*, siendo *k* el número de coeficientes en el espacio transformado. Es capaz de manejar consultas de longitud inferior a *n*, siendo *n* la longitud de la consulta original. Para manejar consultas más largas es un poco más difícil. La medida de la similitud utilizada es la distancia Euclídea. Dicha distancia es la obtenida para normas *Lp* (42) cuando *p*=2.

$$(42)\ L_p(\vec{x}-\vec{y}) = \left(\sum_{i=1}^{L}|x_i - y_i|^p\right)^{\frac{1}{p}}$$

En (Yi et al. 2000) se extiende el estudio sobre la representación PCA a normas *Lp* arbitrarias.

Índice PAA o aproximación agregada a tramos

El índice PAA presentado en (Keogh. 2000a) presenta un índice simple de entender e implementar. Parte cada serie temporal en k subsecuencias, las mismas para todas las series, aproximando cada secuencia por su media o su varianza. Considerando que M es el número de series temporales en nuestra base de datos, n la dimensión original de los datos y N la dimensionalidad reducida, podemos definir el ratio de compresión de la siguiente manera $C_{Ratio} = N/n$. PAA representa cada serie como una secuencia de funciones caja donde cada caja es de la misma longitud (43).

$$(43)\ \overline{X_i} = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} Xj$$

Dada dicha representación, podemos aproximar la distancia Euclídea según (44):

$$(44)\ DR(\overline{X},\overline{Y}) = \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^{N} (\overline{x_i} - \overline{y_i})^2}$$

Esta medida es una cota inferior de la distancia Euclídea, y más flexible, puesto que permite utilizar la distancia Euclídea sopesada. Además el índice puede ser construido en tiempo lineal.

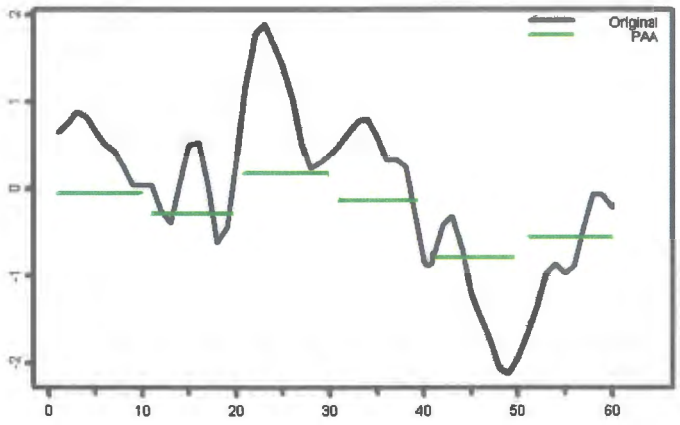


Figura 3. 21. Representación aproximada de una serie mediante PAA.

Índice APCA o aproximación constante adaptativa a tramos

La idea básica es generalizar PAA de manera que los segmentos tengan longitudes arbitrarias. Serán necesarios dos coeficientes para representar cada segmento, su valor y su longitud. La idea subyacente es que muchas señales tienen poco detalle en algunas localizaciones y gran detalle en otras .

Esta representación tiene la posibilidad de ajustarse a los datos. En (Keogh et al. 2001) se presenta un método de indexado apropiado para dicha representación.

El tiempo necesario para obtener esta representación es rápido, pues es de orden $O(n)$, soporta consultas de longitud variable, medidas no Euclídeas y medidas Euclídeas sopesadas. Soporta consultas exactas rápidas y consultas aproximadas más rápidas sobre la misma estructura de datos.

Aproximaciones simbólicas

Supongamos que nuestros datos no son valores reales, o pensemos en lo que se podría hacer con datos discretos en lugar de con datos reales. Se trata de convertir la serie temporal en un alfabeto de símbolos discretos y usar técnicas de indexado de palabras para gestionar los datos. En (Patel et al. 2002) se presenta un mecanismo para obtener una representación simbólica de la serie temporal. La representación simbólica obtenida admite la definición de una cota inferior para la distancia Euclídea.

Trabajos posteriores tales como (Lin et al. 2003) plantean también una cota inferior de la distancia Euclídea. En este caso la representación se obtiene obteniendo inicialmente la representación PAA para posteriormente convertir dicha representación a símbolos. SAX ha sido posteriormente utilizada en otros trabajos, tal es el caso de (Keogh, E. 2005) y una extensión de SAX ha sido también usada para representar de series temporales financieras (LKhagva,B. 2006).

CAPÍTULO 4

MEDIDAS DE LA DISTANCIA

Una vez presentadas algunas de las representaciones de series temporales comúnmente usadas, es necesario estudiar las medidas de la similitud entre series temporales más frecuentemente utilizadas hasta el momento.

Distancia-F

En (Das et al. 1997) se presenta un modelo de similitud que tiene en cuenta outliers y diferentes funciones escala. Dicho concepto de similitud se define sobre la base de un conjunto de transformaciones F que relacionan enteros con enteros. La distancia definida será la distancia-F y se basa en el recuento del número de puntos que se encuentran cerca en las series temporales. Antes de dicho recuento,

las series temporales pueden ser transformadas por una función de transformación perteneciente a una familia predeterminada, F, de transformadas (por ejemplo, para cambio de escala y línea base). Se define un intervalo de confianza de manera que, dos puntos a_i , b_j , se consideran cercanos si el módulo de la diferencia cae dentro del intervalo de confianza. La distancia entre dos series se define como la fracción de puntos que están cerca. Permite capturar similitudes entre series que siguen tendencias con ciertas fluctuaciones alrededor de dichas tendencias.

Una distancia cercana a 1 implica que las dos series están, la mayoría del tiempo, cercanas dentro de unos límites predefinidos.

La noción de similitud plantea que dos secuencias X e Y son similares si existe una función $f \in F$, de tal manera que, una subsecuencia larga X' de X puede ser transformada aproximadamente en una subsecuencia larga Y' de Y. En este caso, X' y Y' no son secuencias consecutivas de X o de Y respectivamente, aunque los puntos de X' y Y' si aparecen en el mismo orden en X e Y. Esto permite que se produzcan outliers. Si X' e Y' son parecidos, el número de outliers será pequeño y X' e Y' serán aproximadamente de la misma longitud que X e Y.

Las funciones F consideradas son las funciones lineales que, aún siendo funciones simples, permiten la búsqueda de similitudes con distintos factores escala y líneas base. Se presenta un algoritmo en tiempo polinómico que soluciona el problema de la búsqueda de la transformación lineal que maximice la longitud

de la subsecuencia común (dentro de un cierto margen ε). El algoritmo usa mecanismos de la geometría computacional.

Posibles modificaciones incluirían restricciones que permitiesen transformar un elemento x_i en otro y_j siempre que $|i - j| < k$,siendo dicha k una constante. Esta restricción impediría que las secuencias tuviesen una gran diferencia temporal.

Reglas de transformación

(Mendelzon and Milo. 1995) plantean un marco de trabajo general para el cálculo de la similitud entre secuencias. Dicho marco de trabajo presenta tres componentes:

- Un lenguaje de patrones *P*.
- Un lenguaje de reglas de transformación *T*. Cada regla tienen asociado un costo. Un ejemplo de reglas de transformación: colapsar segmentos adyacentes en un segmento de manera que la nueva pendiente sea la media sopesada de las pendientes de los dos segmentos y la longitud sea la suma de las longitudes previas.
- Un lenguaje de consulta *L*.

Una expresión en *P* especifica un conjunto de objetos. Un objeto *A* es considerado similar a otro objeto *B* si puede reducirse a él mediante un conjunto de transformaciones definidas en *T*. El lenguaje de consulta propuesto por dicho trabajo es una extensión del cálculo relacional con predicados que chequean si un objeto *A* puede ser transformado en un miembro del conjunto de objetos

descritos por la expresión e usando la transformación t a un costo limitado por c. El marco de trabajo puede afinarse a las necesidades de una aplicación en concreto seleccionando P, T y L.

En (Rafiei and Mendelzon. 1997), (Rafiei and Mendelzon. 1998) y (Rafiei. 1999) se han utilizan combinaciones de medias móviles, escalados y traslaciones. Sea T un conjunto de reglas de transformación y sea $c(t)$ el costo de la regla t. La formulación general de la medida de la distancia planteada es la mostrada en (45):

$$
\begin{aligned}
D(X,Y) = min\{&D(X,Y),\\
&min_{t\ en\ T}(c(t) + D(t(X),Y)),\\
&min_{t\ en\ T}(c(t) + D(X,t(Y))),\\
&min_{t1,t2\ en\ T}(c(t1) + c(t2) + D(t1(X),t2(Y)))\ )\}
\end{aligned}
\tag{45}
$$

Según los experimentos realizados en dicho trabajo, la distancia Euclídea combinada con transformaciones de medias móviles produce una medida de la similitud más intuitiva, pero la desventaja es que el cómputo de subsecuencias coincidentes se vuelve más complicado. Además, la extracción de características se vuelve complicada si las reglas a aplicar llegan a ser dependientes de la X e Y en cuestión. Otra de las desventajas señaladas es que la distancia Euclídea en el espacio de características puede que no sean una buena aproximación de la serie en el espacio original.

Alineamiento dinámico temporal: ADT (DTW)

Esta distancia, a la que denominaremos indistintamente según el acrónimo dado anteriormente (ADT) o bien por el acrónimo correspondiente a su denominación en inglés DTW (dynamic time warping), es un mecanismo utilizado en procesamiento del habla que fue introducido en el área de minería de datos en (Berndt and Clifford. 1994) como una alternativa a la distancia Euclídea. Se introdujo para tratar problemas de clustering en procesos que tienen la misma forma pero diferentes velocidades locales y distintos orígenes temporales. La serie a la que se ha aplicado el alineamiento dinámico temporal evoluciona como la original pero a diferentes velocidades locales y es sensible a outliers. Es aplicable a problemas tales como el reconocimiento de voz, señales de audio y vídeo y señales médicas, siempre que sea necesario permitir una cierta elasticidad, aceleración (encogimiento), o desaceleración (estiramiento), en porciones de las secuencias a ser clasificadas. Posteriormente, ha sido utilizado en múltiples trabajos, por ejemplo en (Chu et al. 2002).

La idea básica es la siguiente: si disponemos de dos series $X = x_1, x_2, x_3, ..., x_n, Y = y_1, y_2, y_3 .... y_m$, podemos extender cada serie repitiendo elementos para, posteriormente, calcular la distancia Euclídea entre las series X' e Y'.

Si disponemos de dos subsecuencias $X = x_1, x_2, x_3, ... x_i$ e $Y = y_1, y_2, y_3, ... y_j$, llamaremos camino de alineamiento a aquel camino obtenido recorriendo la matriz de dimensiones $n \times m$ de manera que, un camino a lo largo de dicha matriz,

al que denominaremos *w*, determina una alineación de las dos secuencias *X* e *Y*. La distancia de alineamiento entre dos series *X* y *Y*, coincidirá con el camino más corto de entre todos los caminos posibles.

$$ADT(X,Y) = min\left\{ \sqrt{\sum_{k=1}^{k} w_k} \Big/ k \right. \tag{46}$$

Para calcular el coste de cada uno de los caminos, debemos completar la matriz $n \times m$ donde, para cada una de las celdas (i, j), tendremos la distancia $D(i, j)$ que obtendremos mediante la siguiente función recursiva (47):

$$D(i, j) = | x_i - y_j | + min\{ D(i - 1, j), D(i - 1, j - 1), D(i, j - 1) \} \tag{47}$$

La matriz almacena en la celda (i, j) la distancia de alineamiento temporal acumulada más corta entre $\{x_0, x_1, ... x_i\}$ y $\{y_0, y_1, ..., y_j\}$. Las técnicas estándar para lograr dicho resultado, usan la programación dinámica con una complejidad cuadrática. La implementación básica es del $O(n^2)$, donde *n* es la longitud de las secuencias y se debe resolver el problema para cada par (i, j). Si se especifica una ventana de alineamiento *r*, sólo se resuelve para los pares (i, j) para los cuales $| i - j | <= r$ resultando el algoritmo del orden $O(nr)$.

Para alinear dos secuencias $Q = q_1, q_2, ..., q_i, ..., q_n$ y $C = c_1, c_2, ..., c_i, ..., c_n$ utilizando ADT, se construye una matiz de $n \times m$ donde el

elemento *(i, j)* de la matriz contiene la distancia $d(q_i, c_j)$ entre los dos puntos q_i y c_j (distancia Euclídea $d(q_i, c_j) = (q_i - c_j)^2$). Cada elemento de la matriz, *(i, j)* corresponde a la alineación de los puntos q_i y c_j. Un camino de alineamiento W es un conjunto de elementos contiguos de la matriz que definen una correspondencia entre Q y C. El elemento número k de W se define como $w_k = (i, j)_k$ de manera que cumpla (48):

$$(48)\ W = w_1, w_2, ..., w_k, ... w_K \qquad \max(m,n) \le k < m + n - 1$$

El algoritmo para el cálculo de la distancia de alineamiento dinámico temporal mediante programación dinámica es el siguiente:

```
Algoritmo distancia_alineamiento_temporal(x,y)
Long_x=longitud_de_x;
Long_y=longitud_de_y;
Matriz[1,1]= D(x1,y1)
Para (2<=i<= Long_x) Matriz[i,1]=D(xi,y1)+ Matriz[i-1,1];
Para (2<=j<= Long_y) Matriz[1,j]=D(x1,yj)+ Matriz[1,j-1];
Para (2<=i<= Long_x)
    Para (2<=j<= Long_y)
     Matriz[i,j]=D(xi,yj)+
       (min(Matriz[i-1,j],Matriz[i,j-1],Matriz[i-1,j-1]);
Devolver Matriz[Long_x, Long_y]
```

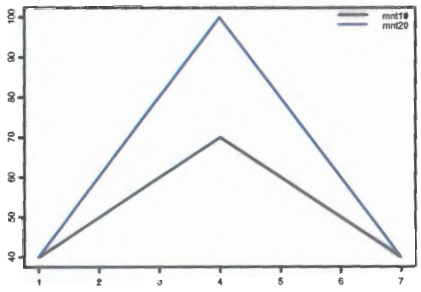


Figura 4. 1. Dos Series mnt10=(40,50,60,70,60,50,40) y mnt20=(40,60,80,100,80,60,40)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	**0**	20	60	120	160	180	180
[2,]	**10**	10	40	90	120	130	140
[3,]	30	**10**	30	70	90	90	110
[4,]	60	20	**20**	**50**	**60**	70	100
[5,]	80	20	40	60	70	**60**	80
[6,]	90	30	50	90	90	**70**	70
[7,]	90	50	70	110	130	90	**70**

Figura 4. 2. Matriz de distancias Manhathan acumuladas para mnt20 y mnt10. En negro, el camino de alineamiento correspondiente.

Existen ciertas restricciones en los caminos de alineamiento:

- El camino debe ser monótono, de manera que el camino no debe ir hacia abajo o a la izquierda. Dado $w_k = (a,b)$, entonces $w_{k-1} = (a',b')$ donde $a - a' \geq 0$ y $b - b' \geq 0$.

- El camino debe ser continuo, no se debe saltar ningún elemento en la secuencia. Dado $w_k = (a,b)$ y $w_{k-1} = (a',b')$ entonces $a - a' \leq 1$ y $b - b' \leq 1$
- El camino de alineamiento debe empezar y terminar en extremos opuestos de la matriz, $w_1 = (1,1)$ y $w_k = (m,n)$.

Además de dichas restricciones globales, se imponen otras restricciones sobre el camino de alineamiento que limitan la distancia a la que el camino de alineamiento puede desviarse de la diagonal. El subconjunto de la matriz que el camino de alineamiento puede visitar es la denominada ventana de alineamiento. Dos de las restricciones más utilizadas son la banda Sakoe-Chiba y el paralelograma de Itakura.

La complejidad dada anteriormente es la que corresponde a la comparación de dos secuencias, pero hemos de tener en cuenta que en las aplicaciones de minería de datos las dos situaciones posibles son las siguientes:

- Coincidencia completa: tenemos una secuencia Q y un conjunto de subsecuencias X de aproximadamente la misma longitud en la base de datos y queremos encontrar entre ellas la secuencia más similar a Q.
- Coincidencia de subsecuencias: disponemos de una secuencia de consulta Q y una secuencia más larga R de longitud X y queremos encontrar la mejor coincidencia para Q en R explorando cada subsección de R.

En cualquiera de los casos la complejidad es $O(n^2 X)$ que es un orden intratable para muchos problemas reales.

Se han probado modificaciones sobre ADT tales como la planteada en (Keogh. 2001) donde, en lugar de construir la matriz de distancias y que el elemento (i, j) de la matriz represente la distancia Euclídea $d(q_i, c_j)$ entre dos puntos q_i y c_j se construye dicha matriz con la distancia obtenida con el cuadrado de la diferencia entre la derivada estimada para los puntos q_i y c_j según (49) :

$$(49)\ D_x[q] = \frac{(q_i - q_{i-1}) + \frac{(q_{i+1} - q_{i-1})}{2}}{2}$$

Otros acercamientos para obtener un cálculo aproximado de ADT se pueden encontrar en (Keogh. 1999b) donde se calcula la distancia de alineamiento pero sobre una representación lineal a tramos de la serie, modificando la medida de la distancia, de manera que, la distancia entre dos segmentos es ahora el cuadrado de la distancia entre sus medias. En otro trabajo del mismo autor (Keogh. 2000b) se utilizaba ADT sobre la representación PAA de las series temporales.

Debido al excesivo tiempo de cálculo, y para agilizar los cálculos de esta distancia, se han utilizado cotas inferiores, menos costosas. Usando cotas inferiores de la distancia de alineamiento sólo se realizan los cálculos más costosos si es estrictamente necesario.

El algoritmo a utilizar para poder utilizar convenientemente una cota inferior de una distancia en el escaneo secuencial es el siguiente:

```
Algoritmo para el escaneo secuencial
  de la cota inferior (Q)
mejor_candidato=∝
para todas las secuencias en la base de datos
      d_c_i= distancia_cota_inferior(Ci,Q)
          si  d_c_i < mejor_candidato
             distancia_real = ADT(Ci,Q)
             si distancia_real < mejor_candidato
                    mejor_candidato= distancia_real
                    indice_ mejor_candidato=i
             fin si
           fin si
   fin para
   fin
```

Se han explorado múltiples cotas inferiores de ADT, entre ellas las planteadas en (Yi et al. 2000) y (Kim et al. 2001). En el primero de ellos se suministra un mecanismo para calcular una cota inferior de la distancia que se procesa en poco tiempo y que permita descartar rápidamente secuencias que no sean interesantes.

Dados $\vec{x} = (x_1,....,x_m)$ y $\vec{y} = (y_1,....,y_n)$ y siendo $\max_{\vec{x}}$ y $\max_{\vec{y}}$ los máximos valores en $\vec{x}$ e $\vec{y}$ respectivamente y $\min_{\vec{x}}$ y $\min_{\vec{y}}$ los mínimos

valores en $\vec{x}$ e $\vec{y}$ respectivamente. Si se definen los siguientes rangos $R_x = \langle \max_x, \min_x \rangle$ y $R_y = \langle \max_y, \min_y \rangle$ hay tres posibles disposiciones de los rangos:

(C1): R_x y R_y son disjuntos ($\min_x > \max_y$)

(C2): R_x y R_y se solapan ($\min_x \leq \max_y, \min_x \geq \min_y$)

(C3): R_x contiene a R_y ($\min_x < \max_y$)

Se define una nueva función distancia D_{lb} según (50):

$$(50)\ D_{lb} = \begin{Bmatrix} \max\left(\sum_i \left| x_i - \max_y \right|, \sum_j \left| y_j - \min_x \right| \right) & ,(C1) \\ \sum_i \phi\left(x_i, \max_y\right) + \sum_j \phi\left(\min_x, y_j\right) & ,(C2) \\ \sum_i \phi\left(x_i, \max_y\right) + \sum_i \phi(\min_y, x_i) & ,(C3) \end{Bmatrix}; \phi(a,b) = \begin{cases} |a-b| & a > b \\ 0 & en\ \ otro\ \ caso \end{cases}$$

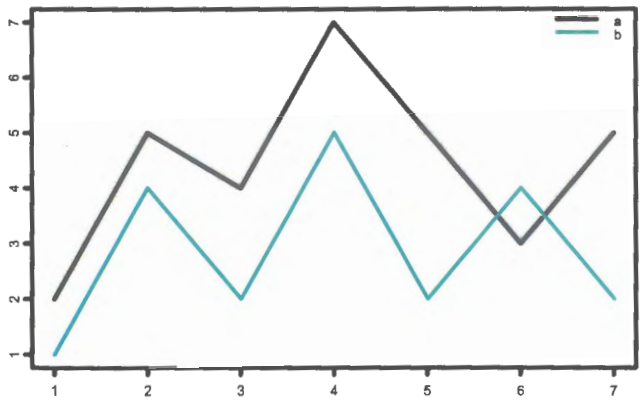


Figura 4. 3. Serie a=(2,5,4,7,3,5) y serie b=(1.4,2,5,2,4,2). D_{lb} es 3 para dichas series siendo ambas del caso C2. La distancia de alineamiento entre las dos series es de 11.

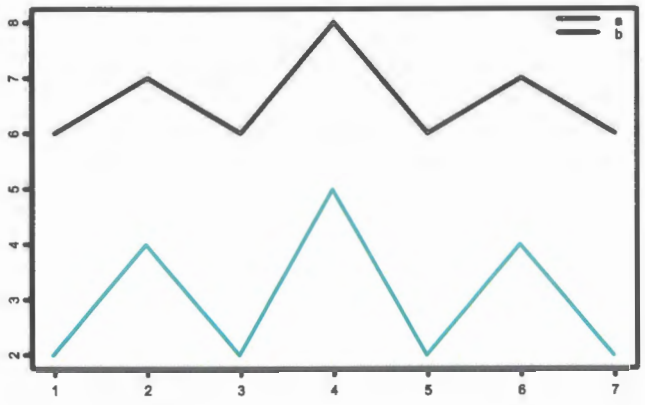


Figura 4. 4. Series a=(6,7,6,8,6,7,6) y b=(2,4,2,5,2,4,2). D_{lb} da un resultado de 21para dichas series siendo ambas series del caso C1. La distancia de alineamiento entre las dos series es de 25.

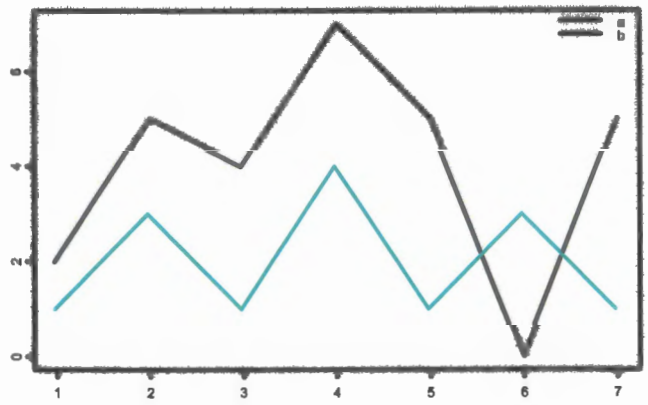


Figura 4. 5. Series *a* y *b*. D_{lb} da un resultado de 7. Ambas series son del caso C3. La distancia de alineamiento es de 15.

Esta distancia se puede calcular en un tiempo lineal respecto a la longitud de la secuencia y además es una cota inferior para la distancia de alineamiento. Dicha cota inferior será la conocida como LB_Yi y por ser cota inferior de ADT cumplirá el siguiente corolario:

Para cualesquiera dos secuencias $\vec{x} = (x_1,....,x_m)$ e $\vec{y} = (y_1,....y_n)$, si la distancia de alineamiento $D_{adt}(\vec{x},\vec{y}) < \varepsilon$ entonces $D_{lb}(\vec{x},\vec{y}) < \varepsilon$.

El algoritmo IDDTW (Iterative deepening DTW) presentado en (Chu et al. 2002) surge a raíz de la limitación que supone el hecho de que el usuario debe especificar el ratio de compresión a aplicar en la representación. Para solventar dicho problema, el nuevo mecanismo obtiene modelos probabilísticos de los errores de aproximación para todos los niveles de aproximación antes del proceso de consulta. El algoritmo examina iterativamente secuencias a niveles más finos de aproximación y compara los resultados a los modelos probabilísticos y a los valores especificados por el usuario para las omisiones incorrectas. Con dicha información, el algoritmo evalúa cuando una secuencia candidata puede ser descartada o puede resultar interesante considerar una aproximación más fina.

El problema del indexado del alineamiento dinámico temporal ha sido abordado por (Yi et al. 1998) y (Keogh. 2002). En el primero de los artículos los autores constatan el hecho de que ADT es una métrica que sufre de dos grandes problemas a la hora del indexado. El primero problema es que es una distancia que no aporta ninguna pista sobre características que puedan resultar indexables, y la siguiente es el tiempo de cómputo. Se propone en dicho trabajo el uso de una

modificación de la técnica de Fastmap, visto en el anterior apartado, como mecanismo de reducción de la dimensionalidad, para obtener un indexado aproximado de ADT.

Dicho mecanismo presenta el problema de que permite omisiones incorrectas y por lo tanto, es utilizado en (Yi et al. 1998) como una estimación del rango de búsqueda de la función distancia Euclídea con el fin de recuperar un número de candidatos más reducido. Posteriormente, en el proceso de filtrado de las inclusiones innecesarias, se utilizará D_{lb} como una cota inferior de la distancia de alineamiento dinámico temporal. En (Kim et al. 2001) se presenta un algoritmo para el indexado exacto de series temporales con ADT. El método extrae cuatro características de las series y organiza dichas características en una estructura de índice. Introduce una función que es una cota inferior de la distancia de alineamiento dinámico temporal basada en las cuatro características extraídas. La cota inferior de ADT entre dos series, será el máximo de los valores absolutos de las diferencias entre las características extraídas, siendo dichas características el vector constituido por el primer, último, máximo y mínimo valor de la serie. Dicha cota inferior se denomina LB_Kim.

Considerando las restricciones sobre el camino de alineamiento $w_k = (i, j)$, de manera que, $j - r \le i \le j + r$ donde r define el alcance o el rango permitido de alineamiento para un punto dado de la secuencia y usando dicha r se definen las siguientes dos secuencias $U_i = \max(q_{i-r} : q_{i+r})$ y $L_i = \min(q_{i-r} : q_{i+r})$. La medida

de cota inferior para ADT propuesta en (Keogh. 2002) a la que se denomina LB_Keogh es la siguiente (51):

$$(51)\ LB\_Keogh(Q,C) = \sqrt{\sum_{i=1}^{n} \begin{cases} (c_i - U_i)^2 & si \quad c_i > U_i \\ (c_i - L_i)^2 & si \quad c_i < L_i \\ 0 & en \quad otro \quad caso \end{cases}}$$

Con el objeto de poder indexar ADT, se ha usado la cota inferior LB_Keogh en un espacio de baja dimensión al que se denomina LB_PAA. Este mecanismo utilizará la aproximación constante a tramos para la representación en el espacio de baja dimensionalidad.

Definidas ciertas aproximaciones constantes a tramos de las anteriormente definidas U y L que se denotan como $\hat{U}$ y $\hat{L}$ y calculadas según (52):

$$(52)\quad \hat{U}_i = \max\left(U_{\frac{n}{N}(i-1)+1}, \ldots, U_{\frac{n}{N}(i)} \right)$$
$$\hat{L}_i = \min\left(L_{\frac{n}{N}(i-1)}, \ldots, L_{\frac{n}{N}(i)} \right)$$

Se define LB_PAA de manera que, dada una secuencia candidata C transformada en $\overline{C}$ mediante la representación PAA y una secuencia de consulta Q con sus funciones $\hat{U}$ y $\hat{L}$ correspondientes entonces definimos LB_PAA según la ecuación (53):

$$(53)\quad LB\_PAA(Q,\overline{C}) = \sqrt{\sum_{i=1}^{N} \frac{n}{M} \begin{cases} (\overline{c}_i - \hat{U}_i)^2 \; si \quad \vec{c}_i > \hat{U}_i \\ (\overline{c}_i - \hat{L}_i)^2 \; si \quad \vec{c}_i < \hat{L}_i \\ 0 \quad en \quad otro \quad caso \end{cases}}$$

El último paso para que el indexado sea posible será definir una función MINDIST(Q,R) que devuelva una cota inferior de la medida de la distancia entre la consulta Q y R donde R es un rectángulo de área mínima (MBR). Suponiendo que la estructura de índice contenga un nodo hoja U y si $R = (L, H)$ es el MBR asociado a U donde $L = \{l_1, l_2, ..., l_N\}$ y $H = \{h_1, h_2, ...h_N\}$ son los menores y los mayores puntos finales de la mayor diagonal de R, MINDIST se define según (54):

$$(54)\quad MINDIST(Q,R) = \sqrt{\sum_{i=1}^{N} \frac{n}{M} \begin{cases} (I_i - \hat{U}_i)^2 \; si \quad I_i > \hat{U}_i \\ (h_i - \hat{L}_i)^2 \; si \quad h_i < \hat{L}_i \\ 0 \quad en \quad otro \quad caso \end{cases}}$$

Trabajos posteriores (Han et al. 2003) siguen abordando la problemática relacionada con el cálculo rápido de aproximaciones de ADT. En este trabajo se obtiene la distancia entre dos series x e y mediante un proceso de reducción de la resolución en el que se siguen los siguientes pasos:

- Descomposición Haar de las series $\vec{x}$ e $\vec{y}$. Las series $x = (3,1,0,2,-3,-4,1,2)$ e $y = (2,3,1,-3,-4,1,0,1)$ se transformarían en las siguientes series

$$x' = (0.25, 1.25, 0.5, -2.5, 1.0, -1.0, 0.5, -0.5)\ e$$
$$y' = (0.125, 0.625, 1.75, -1.0, -0.5, 2.0, -2.5, -0.5)$$

- Reconstrucción Haar de las series $\vec{x}$ e $\vec{y}$ con cierta resolución, por ejemplo, usando los primeros 3 coeficientes, las reconstrucciones serían :

$$X = (2,\ 2,\ 1,\ 1, -1, -1, -1, -1)\ e$$
$$Y = (2.5,\ 2.5, -1.0, -1.0, -0.5, -0.5, -0.5, -0.5)$$

- Obtención de los vectores comprimidos $\vec{X}$ e $\vec{Y}$ correspondientes a $\vec{x}$ e $\vec{y}$ que en este caso se corresponde con los siguientes valores :

$$\vec{X} = (2, 1, -1)\ \ e$$
$$\vec{Y} = (2.5, -1.0, -0.5)$$

- Cálculo de la distancia de alineamiento dinámico temporal en los vectores $\vec{X}$ e $\vec{Y}$ resultantes de manera que se agrupen tan pronto como sea posible los picos y los valles de las dos series para pasar posteriormente a compensar dicha distancia con la distancia perdida en el proceso de compresión. Para las series dadas, y teniendo en cuenta la distancia Euclídea para el cálculo de dicha distancia, ADT será de 1.658312.

	Y_0	Y_1	Y_2
X_0	**0.500000**	3.041381	3.937004
X_1	**1.581139**	2.061553	2.549510
X_2	3.840573	**1.581139**	**1.658312**

Figura 4. 6. Camino de alineamiento entre $\overline{X} = (2, 1, -1)$ e $\overline{Y} = (2.5, -1.0, -0.5)$.

Para obtener la compensación correspondiente al proceso de compresión se consideran dos casos distintos:

- Si en el camino de alineamiento entre $\vec{X}$ e $\vec{Y}$ solo nos movemos a lo largo del eje horizontal (el eje de las X) entonces la compensación necesaria para la distancia será la siguiente (55):

$$(55)\ DC = \left\{\left(X_i - Y_j\right)^2 \times \left|\left\{Y_j,....,Y_j\right\}\right|\right\}$$

Donde se representa por $\left|\left\{Y_j,....,Y_j\right\}\right|$ el número de veces que el valor Y_j se encontraba comprimido en $\overline{Y}$. De la misma manera, podríamos hablar de movimientos a lo largo del eje de la Y. El primero de los movimientos en el camino de alineamiento se corresponde con esta situación de manera que $DC = 0.52$.

- Si en el camino de alineamiento entre $\vec{X}$ e $\vec{Y}$ nos movemos a lo largo de ambos ejes la distancia se compensará según la siguiente expresión (56):

$$(56)\ DC = \left\{ADT\left(\left\{X_i, X_i,...., X_i, X_{i+1}\right\}, \left\{Y_j, Y_j,....., Y_j, Y_{j+1}\right\}\right) - \left(X_i - Y_j\right)^2 - \left(X_{i+1}, Y_{j+1}\right)^2\right\}^{\frac{1}{2}}$$

El segundo de los movimientos en el camino de alineamiento es una situación de este tipo necesitando por lo tanto una compensación de $DC = 1.52$.

El resultado obtenido será $D_{lowresTW}$ que estará definido según (57), teniendo en cuenta que se considerará como estado a todo par correspondiente al camino de alineamiento entre $\vec{X}$ e $\vec{Y}$:

$$(57)\ D_{lowresTW}(\vec{x},\vec{y}) = \left\{ DTW(\vec{X},\vec{Y})^2 + \sum_{s=1}^{n^{\circ}estados} DC_s^{\,2} \right\}^{\frac{1}{2}}$$

Otros trabajos (Rath and Manmatha) han trasladado dicha labor de búsqueda de cotas inferiores para la distancia de alineamiento dinámico temporal al área de series temporales multivariadas.

Distancia editada

La distancia editada entre dos secuencias alfanuméricas *A* y *B* (referidas como texto y patrón) en un alfabeto Σ está definida como el mínimo número de operaciones que son necesarias para convertir un texto *A* de consulta en un patrón *B*. Se permiten tres operaciones: borrados $a \to \varepsilon$, inserciones $\varepsilon \to b$ y cambios $a \to b$ donde *a* y *b* están en Σ y siendo ε la cadena vacía. Asumiendo que el costo de dichas operaciones es 1, la distancia editada es el mínimo número de operaciones para obtener un patrón a partir de un texto.

Si denotamos como *D(i, j)* la distancia editada mínima entre dos textos *Q[1...i]* y *S[1...j]* entonces definiremos la distancia editada recursiva según como (58):

$$(58)\ D(i,j)=\begin{cases} j \quad si \quad i=0 \\ i \quad si \quad j=0 \\ D(i-1,j-1)=\min\{D[i-1,j-1]+1, D[i-1,j]+1, D[i,j-1]+1\} \\ \qquad si \quad i,j>0 \quad y \quad q_j \; es \quad igual \; a \; s_j \end{cases}$$

Se han utilizado versiones modificadas de la distancia editada (Bozkaya et al. 1997) donde para una recuperación eficiente propone un esquema de indexado que está totalmente basado en longitudes y distancias relativas entre secuencias utilizando para ello árboles Vp. Un caso particular del algoritmo de la distancia editada será la subsecuencia común más larga. De hecho, será una versión restringida de la distancia editada donde no se permiten cambios.

Subsecuencia común más larga: SCML

Otro grupo de medidas de similitud es el basado en la medida de la subsecuencia común más larga (SCML). Dicha medida de la similitud permite la existencia de saltos en una secuencia de manera que siendo X e Y dos series temporales con los siguientes valores:

$$X = 3,2,5,7,4,8,10,7$$

$$Y = 2,5,4,7,3,10,8,6$$

Se determina inicialmente la subsecuencia común más larga permitiendo la existencia de saltos, de manera que $SCML = 2,5,7,10$ y definiendo posteriormente la similitud entre las dos subsecuencias como $Sim(X,Y) = |SCML|$. Como defectos o deficiencias de dicha medida de la similitud:

- Diferentes factores escala y líneas base necesitan escalar o transformar una serie a la otra.
- Debería ser transigente a la hora de comparar elementos (incluso después de transformaciones).
- Es una técnica usada en reconocimiento del habla o en la búsqueda de coincidencia de patrones textuales.

Medidas del tipo SCML para series temporales

Comparación de subsecuencias sin escalado (Yazdani. 1996). Si *Sim(i, j)* hace referencia a la similitud entre dos secuencias i, j siendo estas $x_1, x_2, x_3, \ldots x_i$ y $y_1, y_2, y_3, \ldots y_j$ y siendo d una tolerancia permitida, denominada distancia de umbral, se define dicha medida de la similitud según (59):

$$\begin{aligned} &si\,|\,x_i - y_j\,| < d\ entonces \\ (59)\quad &Sim(i,j) = 1 + D(i-1, j-1) \\ &Si\ no\ Sim(i,j) = max\{D(i-1,j), D(i,j-1)\} \end{aligned}$$

Con respecto al escalado, los autores no discuten directamente los problemas de escalado. Es posible usar una transformación de escalado estándar global antes de los cálculos de similitudes. La complejidad de los algoritmos de cálculo de similitudes en este caso se corresponde con las formulaciones en programación dinámica estándar que necesitan un $O(n^2)$ en tiempo, aunque es posible realizar mejoras realizando ciertas suposiciones sobre la entrada.

También podemos encontrar comparaciones de subsecuencias con escalado local y diferentes líneas base en (Agrawal et al. 1995) donde la idea básica es que dos secuencias son similares si tienen el suficiente par de subsecuencias ordenadas no solapadas que son semejantes. Un par de subsecuencias son similares si una subsecuencias escalada y trasladada adecuadamente se parece a la otra. Los escalados y traslaciones son diferentes para cada par de subsecuencias. Para lograrlo, el algoritmo encuentra todos los pares de subsecuencias atómicas (donde atómico significa un cierto tamaño mínimo) en X e Y que sean similares. A continuación se juntan ventanas similares para formar pares de subsecuencias similares más largas. Se trataría, en último lugar, de encontrar una ordenación no solapada de subsecuencias coincidentes que tengan la longitud de coincidencia mayor.

Si analizamos el proceso, podemos ver que la búsqueda de pares de subsecuencias similares puede hacerse utilizando un método de acceso espacial o SAM que contenga todas las ventanas atómicas. A su vez, para juntar ventanas y resolver el problema de la ordenación de subsecuencias, puede reducirse al problema de encontrar el camino mas largo en un grafo dirigido acíclico si los nodos del grafo dirigido acíclico son cada par de ventanas coincidentes.

La idea básica en (Das et al. 1997), que afronta comparación de subsecuencias con escalado global y distintas líneas base, es que X e Y son similares si poseen una subsecuencia común larga X', Y' de manera que Y' es aproximadamente $X'+b$, donde la función lineal de escalado y traslación se deriva de las subsecuencias, y no de las secuencias originales. Si analizamos el algoritmo,

vemos que la subsecuencia común mas larga se puede computar en programación dinámica en un tiempo $O(n^2)$. Se puede modificar fácilmente para que se compute en un tiempo $O(n)$ para un conjunto fijo de transformaciones f o para un conjunto fijo de ventanas coincidentes.

La principal tarea para computar esta similitud, es fijar un conjunto finito de todas las transformaciones lineales fundamentalmente diferentes y obtener la medida de la similitud en cada uno de los f. La primera de las tareas será de un $O(n^2)$, de manera que, el tiempo requerido será de $O(n^3)$. En (Chu and Wong. 1999) se consideran el escalado global y la translación.

Acercamientos probabilísticos al concepto de similitud

En (Keogh. 1997c) se establece un modelo de distancia probabilística entre la serie Q y R de manera que se asume Q como una plantilla ideal que es la que puede ser deformada (de acuerdo a una distribución previa) para poder generar D.

Si D es la deformación observada entre Q y R, es necesario determinar el modelo generativo. Se parte de una representación lineal a tramos de la serie R y de la serie de consulta Q representada como una secuencia de características locales tales como picos, valles, etc. que pueden ser deformadas de acuerdo a distribuciones previas. Se mantiene también información global sobre la localización relativa de dichas características locales en Q. La medida de la similitud probabilística así definida tiene la característica de permitir escalados y

traslaciones y además incorpora conocimiento previo en la propia medida de la similitud.

El presentado a continuación se puede entender como un método basado en el modelo (Ge and Smyth. 2000). Dicho modelo, dada una serie Q, construye un modelo M_Q. Dado un nuevo patrón Q' se mide la similitud calculando $P(Q'|\, M_Q)$.

M_Q es, en este trabajo, un modelo de Markov de estados finitos y tiempo discreto, donde cada segmento corresponde a un estado y donde los datos en cada estado se generan por una curva de regresión. Lo que se suministra es una matriz de transición de estados. Cuando se introduce un estado *i* se dibuja una duración a partir de una distribución *p(t)* de duración de estados, de manera que el proceso continúa en el estado *i* por un tiempo *t* y, después de esto, el proceso cambia a otro estado acorde con la matriz de transición de estados.

Otras aproximaciones, tales como la mostrada en (Focardi. 2001-04), asumen que las series temporales son estacionarias. Permiten medir la distancia entre series de distinta longitud y con formas que, siendo similares en sus distribuciones, no pueden ser directamente comparadas.

Nociones subjetivas de la similitud

Se han realizado intentos de incorporar por parte del usuario nociones subjetivas de la similitud que deberán ser aprendidas por dicho usuario a través de la interacción con el sistema.

"Relevance feedback" es un mecanismo muy utilizado en reconocimiento de voz y que se presentó en (Keogh. 1998) como un mecanismo que era posible

aplicar en el proceso de búsqueda de similitudes. Se usa una representación lineal a tramos de la serie temporal y define una operación de mezcla en la representación de las series temporales, utilizando el mecanismo mencionado para refinar la forma de la consulta.

CAPÍTULO 5

REPRESENTACIÓN BASADA EN PUNTOS IMPORTANTES

Múltiples trabajos (Perng et al. 2000), (Park et al. 1999), (Fink and Pratt. 2003), (Fink et al. 2003), (Pratt. 2001), (Pratt and Fink. 2002) han estudiado la posibilidad de representar las series temporales haciendo uso de los puntos importantes, máximos y mínimos, extraídos con diversos mecanismos. Los extremos son importantes puesto que, si observamos por un instante dos series temporales, podemos considerar que son similares siempre que sus puntos de cambio sean iguales. El resto de la serie estará constituida por las curvas o rectas que conectan dichos puntos de cambio. Una vez realizado este proceso, las tareas de análisis y proceso de series temporales serán más fáciles de realizar, al estar los datos transformados en símbolos más comprensibles. Los trabajos realizados

recientemente tienen en cuenta las distintas maneras de identificar un patrón en una serie temporal. Además de prestar atención al problema de la identificación de los patrones contenidos en una serie, será necesario atender al hecho de que los máximos y mínimos seleccionados se usarán posteriormente para segmentar la serie temporal. El problema de la segmentación se podrá abordar, de esta manera, de una manera más flexible, más efectiva y más acorde a los intereses de los usuarios. Los patrones del análisis técnico utilizados están definidos en base a ciertas restricciones que deben cumplir los extremos extraídos de una serie temporal. En este sentido, la propia definición de los patrones podría ser fácilmente modificada en posteriores pruebas. Es obvio que podríamos estar interesados en introducir más patrones, o, en su caso, redefinir algunos de los utilizados.

Una vez divididas las series temporales en segmentos determinados por estos puntos importantes, es posible realizar comparaciones entre series temporales, usando para ello los segmentos determinados. Las distancias que es posible determinar entre segmentos adquirirán ahora el significado de ser distancias entre segmentos constitutivos de un patrón, ofreciéndonos la posibilidad de buscar coincidencias de patrones de un modo más versátil. Podemos buscar coincidencias en los primeros instantes de formación de un patrón, en los momentos finales de consolidación del patrón o quizá en su momento central. Es posible extraer los instantes posteriores a un patrón, semejante a uno disponible, para predecir su continuidad. Todo ello de una manera efectiva y eficiente.

Los puntos de cambio son esenciales para aquellos que han de leer los gráficos correspondientes a índices bursátiles. Mediante el mecanismo implementado los patrones contenidos en una serie serán rápidamente visualizados. A continuación definiremos los patrones de análisis técnico considerados a lo largo de los capítulos restantes.

Definición de patrones dependientes del dominio

Los conceptos definidos para el análisis técnico parten de la definición de tendencia o dirección que siguen los precios del mercado. Los movimientos en forma de zig-zag seguirán una tendencia alcista desde el momento en el que los sucesivos máximos y mínimos sean cada vez mayores. Una tendencia bajista se detectará con máximos y mínimos cada vez más bajos. Es posible que el mercado presente una tendencia lateral cuando los máximos y mínimos estén siguiendo una línea horizontal.

La "teoría de Dow" parte del supuesto de que en la evolución del mercado se entremezclan tres tipos de movimientos o tendencias: tendencia primaria, tendencia secundaria y tendencia terciaria.

En lo que respecta a las características de las tres tendencias mencionadas anteriormente, las más destacadas serán las que a continuación se mencionan:

- La tendencia primaria, cuya duración puede ser desde varios meses a varios años, corresponde a amplios movimientos alcistas o bajistas que traen como resultado una apreciación o depreciación del valor.

Esta tendencia es seguida por el inversor a largo plazo y determina que el mercado sea alcista o bajista.

- La tendencia secundaria, cuya duración será de tres semanas a varios meses, es una reacción intermedia importante, opuesta a la dirección primaria. Acaba corrigiendo entre un 50 y un 75% del último movimiento primario. Es útil para invertir a medio o corto plazo.
- La tendencia terciaria o menor, es un movimiento de duración menor a tres semanas. Se trata de una breve fluctuación dentro de las tendencias secundarias y su duración oscila entre seis o siete días y puede llegar a ser de hasta tres semanas. Son correcciones a lo largo del día, que suelen estar muy manipuladas, por lo que solamente es útil para los inversores a muy corto plazo.

Dow también argumentaba que, en un gráfico, se puede comparar el movimiento del mercado de valores al movimiento de las aguas marinas, en continuo flujo y reflujo. El movimiento primario serían las mareas, el secundario las olas y, el movimiento diario, las pequeñas ondulaciones que forman las olas. Los movimientos diarios carecen de importancia, lo que realmente interesa descubrir son los movimientos primarios y secundarios (las mareas y las olas), con el objeto de comprar o vender en el momento oportuno. Cuando las crestas o valles de las sucesivas olas son cada vez más altas, el movimiento primario es de alza (la marea está subiendo) y viceversa. Luego, después de un largo periodo de alza (o baja), cuando las crestas de las sucesivas olas son cada vez más bajas (o altas), indicarían un cambio en el movimiento primario. Sin embargo, debido a la

distorsión que suponen los movimientos diarios y también debido a la irregularidad de los movimientos secundarios, se precisa una cierta dosis de prudencia antes de pronosticar un cambio definitivo en la tendencia de fondo del mercado.

Los patrones dependientes del área de aplicación que se mostrarán a continuación, serán extraídos de los fundamentos del análisis técnico. La idea subyacente detrás del análisis mediante gráficos es que determinadas pautas de comportamiento de los precios son repetitivas, es decir, se pueden extrapolar al futuro.

Estos modelos funcionan porque proporcionan imágenes de lo que están haciendo los participantes del mercado y eso permite determinar sus reacciones ante los hechos que se producen. El análisis de gráficos es, en realidad, un estudio de la sicología humana y de las reacciones de los operadores a las cambiantes condiciones del mercado.

Uno de los debates abiertos en el análisis de mercados financieros es la validez de cada uno de los dos mayores métodos de análisis, el fundamental y el técnico. Algunos estudios dicen que el análisis fundamental es más efectivo en la predicción de tendencias a largo plazo (más de un año), mientras que el análisis técnico sería más apropiado para el análisis en horizontes más cercanos, de 90 días, siendo lo ideal combinar ambos.

El análisis técnico examina precios pasados y volúmenes de compra para predecir movimientos futuros en los precios. Centra su atención en la formación de gráficos para capturar tendencias, con el objetivo de identificar oportunidades

de compra o venta, valorando la duración de los cambios de tendencia en el mercado. Es posible aplicar dicha técnica en distintos horizontes temporales: 5 minutos, 15 minutos, cada hora, etc. A lo largo del trabajo mostrado en los siguientes capítulos, se podrá ver que los horizontes temporales cortos son aquellos en los que más se ha incidido. Se han usado para ello cotizaciones minuto a minuto, estudiando series de longitud 40, 90, 180 y 360. Dichas longitudes se corresponden, aproximadamente, con datos correspondientes a media hora, hora y media, tres horas y seis horas respectivamente.

Derivados de la definición de tendencia, aparecen los soportes y las resistencias. Los soportes son niveles de precios para los cuales la cotización rebota y asciende. Las resistencias son niveles de precios donde los precios frenan su ascenso y caen. La línea de tendencia es una línea trazada uniendo sucesivos mínimos o soportes, si es una tendencia alcista, y uniendo sucesivos máximos o resistencias si es bajista.

Las formaciones de hombro-cabeza-hombro (HCH) y hombro-cabeza-hombro invertida (HCHI), son algunas de las señales más seguras y comunes de cambio de tendencia. Vendrán caracterizados por cinco extremos sucesivos $E_1,...,E_5$, que cumplan las condición expuesta en (60) y (61) respectivamente :

$$(60)\ HCH = \begin{cases} E_1 \ \ es \ \ un \ \ máximo \\ E_3 > E_1, E_3 > E_5 \\ E_1 \ \ y \ \ E_5 \ \ están \ \ dentro \ \ del \ \ 1.5\% \ \ de \ \ su \ \ media \\ E_2 \ \ y \ \ E_4 \ \ están \ \ dentro \ \ del \ \ 1.5\% \ \ de \ \ su \ \ media \end{cases}$$

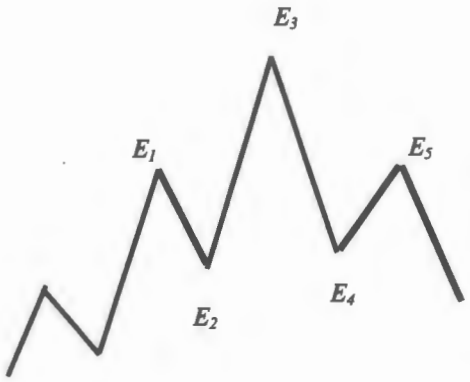


Figura 5. 1. Patrón hombro cabeza hombro

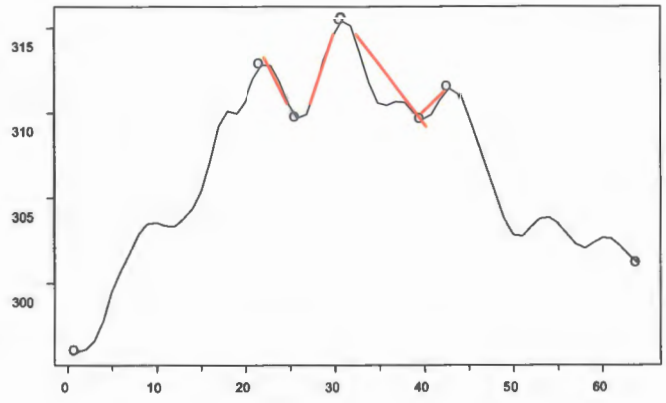


Figura 5. 2. Patrón hombro cabeza hombro encontrado en una serie real

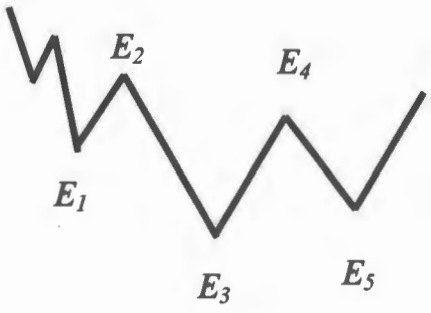


Figura 5. 3. Patrón hombro cabeza hombro invertido.

$$(61)\ HCHI = \begin{cases} E_1 \ \ es \ \ un \ \ mínimo \\ E_3 < E_1, E_3 < E_5 \\ E_1 \ \ y \ \ E_5 \ \ están \ \ dentro \ \ del \ \ 1.5\% \ \ de \ \ su \ \ media \\ E_2 \ \ y \ \ E_4 \ \ están \ \ dentro \ \ del \ \ 1.5\% \ \ de \ \ su \ \ media \end{cases}$$

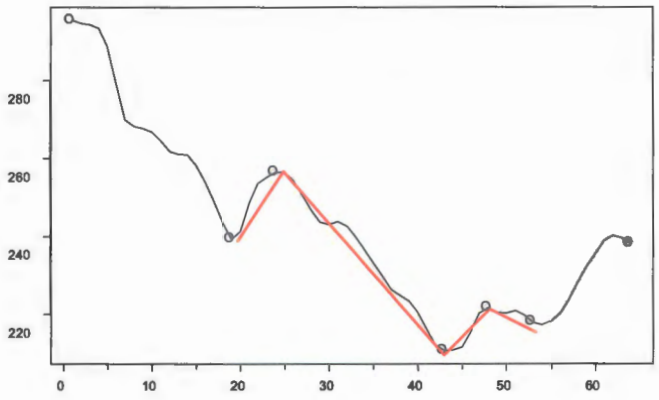


Figura 5. 4. Patrón hombro cabeza hombro invertido encontrado en una serie real.

Estas dos formaciones serán las más frecuentes. A continuación veremos otras formaciones menos comunes. Las formaciones expansivas son aquellas en las que las líneas de tendencia divergen, creando la imagen de un triángulo en expansión. También se llaman megáfonos superiores. Cuando observamos una formación de este tipo sabemos que representa un mercado fuera de control y lo más frecuente es que ocurra cuando se dan los máximos principales del mercado. El patrón expansivo es, normalmente, una formación bajista, pues aparecerá al final de un mercado alcista de importancia.

Vendrá caracterizado por cinco extremos locales consecutivos $E_1,...,E_5$ que deberán de cumplir la condición impuesta en (62).

$$(62)\ MAR = \begin{Bmatrix} E_1 \quad es \quad un \quad máximo \\ E_1 < E_3 < E_5 \\ E_2 > E_4 \end{Bmatrix}$$

Con respecto a la formación expansiva megáfono abajo, las condiciones que se han de cumplir para que cinco extremos consecutivos sean considerados como tal, son las expresadas en (63).

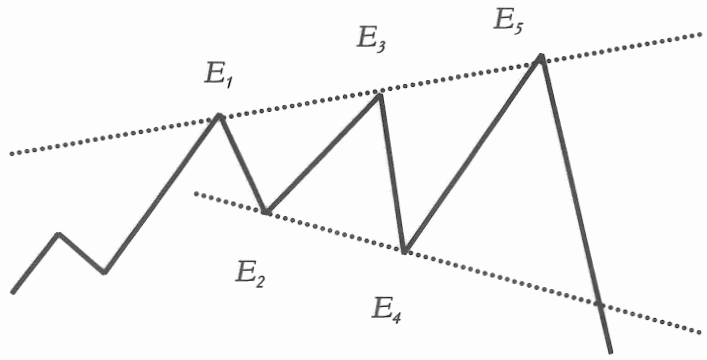


Figura 5. 5. Megáfono arriba

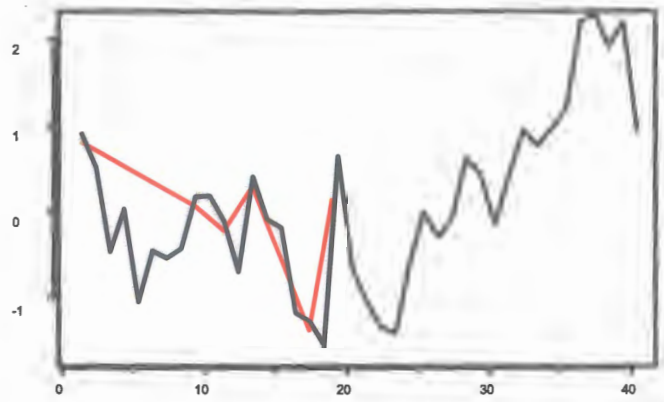


Figura 5. 6. Patrón megáfono arriba detectado en una serie real preprocesada.

$$(63)\ MAB = \begin{Bmatrix} E_1 \ \ es \ \ un \ \ mínimo \\ E_1 > E_3 > E_5 \\ E_2 < E_4 \end{Bmatrix}$$

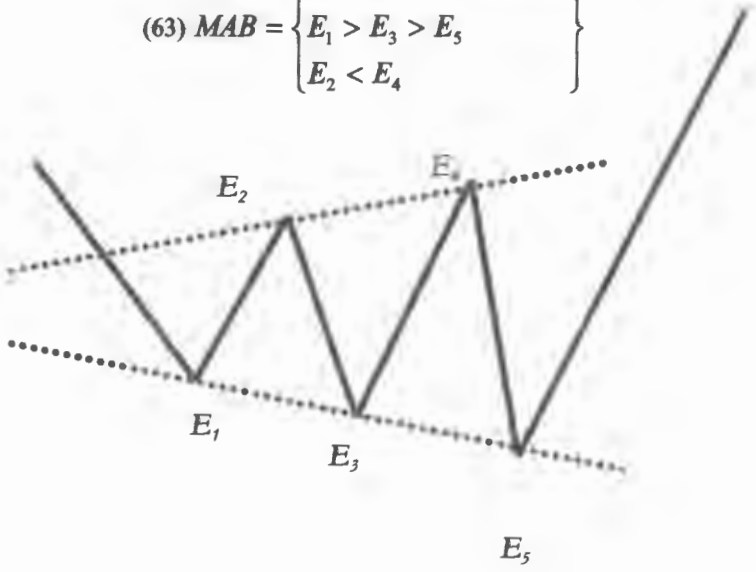


Figura 5. 7. Megáfono abajo

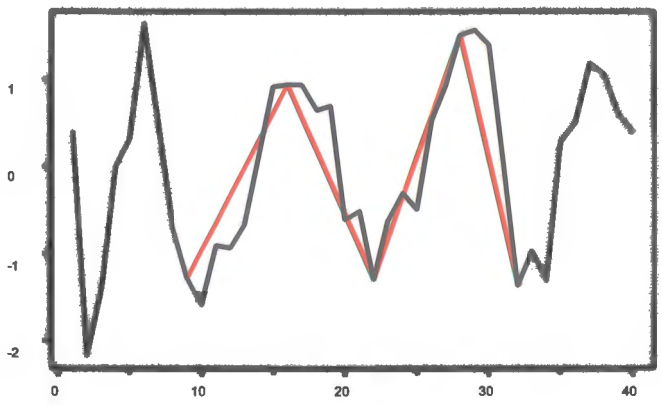


Figura 5. 8. Megáfono abajo detectado en una serie real preprocesada.

Los patrones que se muestran a continuación, indican que el movimiento lateral del precio reflejado en el gráfico no es más que una pausa en la tendencia que prevalece. El siguiente movimiento será en la misma dirección de la tendencia que precedía a la formación. Las formaciones de continuación de tendencia no se prolongan tanto en el tiempo como las formaciones de cambio de tendencia.

El triángulo (TRI) y el triángulo invertido (TRII) son formaciones de continuación de tendencia. El triángulo, también denominado triángulo ascendente es, generalmente, alcista. Ambos están caracterizados por cinco extremos locales consecutivos $E_1,...,E_5$ y deberan cumplir respectivamente las restricciones (64) y (65).

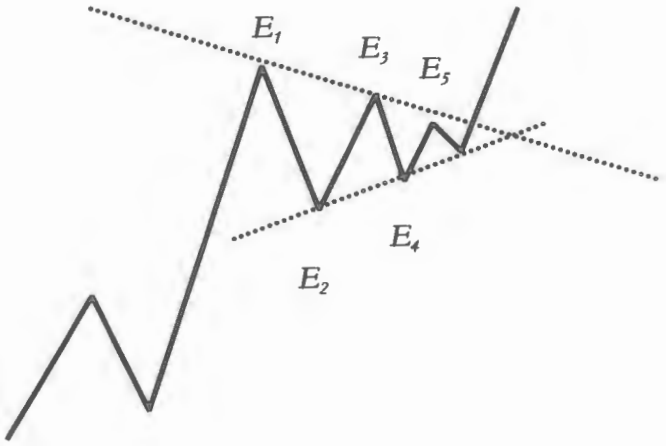


Figura 5. 9. Triángulo

$$(64)\ TRI = \begin{cases} E_1 \quad es \quad un \quad máximo \\ E_1 > E_3 > E_5 \\ E_2 < E_4 \end{cases}$$

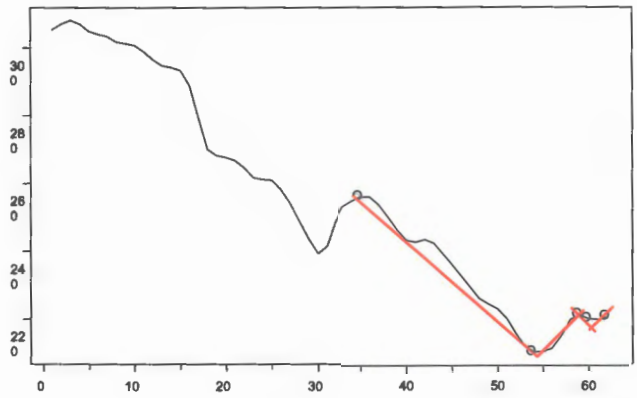


Figura 5. 10. Triángulo encontrado en una serie real.

$$(65)\ TRII = \begin{cases} E_1 \ \ es \ \ un \ \ mínimo \\ E_1 < E_3 < E_5 \\ E_2 > E_4 \end{cases}$$

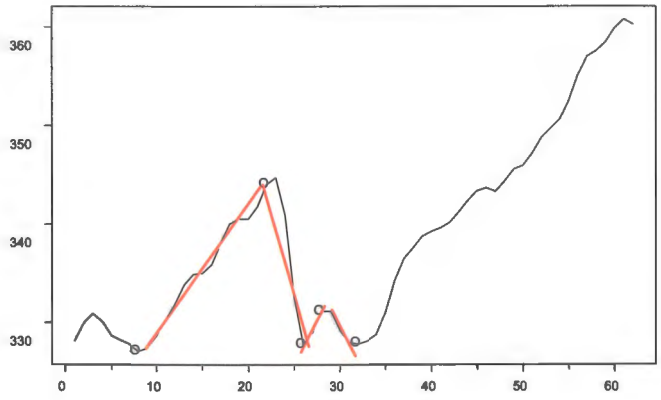


Figura 5. 11. Triángulo invertido encontrado en una real.

Los rectángulos, tanto arriba (RAR) como abajo (RAB), son formaciones de continuación de tendencia. Vendrán caracterizados por cinco extremos sucesivos $E_1,...,E_5$, siempre que cumplan las restricciones impuestas en (66) y (67) respectivamente:

$$(66)\ RAR = \begin{cases} E_1 \ \ es \ \ un \ \ máximo \\ los \ \ soportes \ \ están \ \ dentro \ \ del \ \ 0.75 \ \ de \ \ su \ \ media \\ las \ \ resistencias \ \ están \ \ dentro \ \ del \ \ 0.75 \ \ de \ \ su \ \ media \\ el \ \ menor \ \ de \ \ los \ \ soportes > la \ \ mayor \ \ de \ \ las \ \ resistencias \end{cases}$$

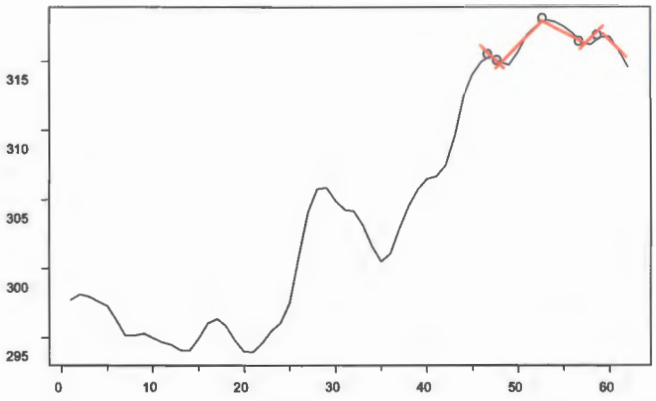


Figura 5. 12. Rectángulo encontrado en una serie real.

$$(67)\ RAB = \begin{cases} E_1 \ \ es \ \ un \ \ mínimo \\ los \ \ soportes \ \ están \ \ dentro \ \ del \ \ 0.75 \ \ de \ \ su \ \ media \\ las \ \ resistencias \ \ están \ \ dentro \ \ del \ \ 0.75 \ \ de \ \ su \ \ media \\ el \ \ menor \ \ de \ \ los \ \ soportes > la \ \ mayor \ \ de \ \ las \ \ resistencias \end{cases}$$

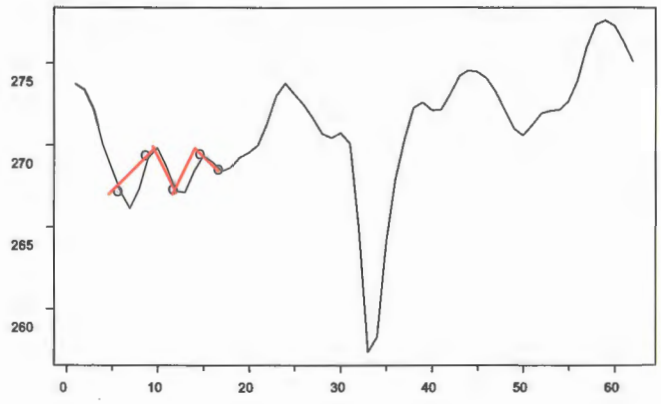


Figura 5. 13. Rectángulo abajo encontrado en la serie real.

Extracción de características: detección de extremos

Definidos dichos patrones, detectarlos en una serie temporal (con un componente aleatorio), es un problema que puede llegar a ser complicado. Al no tener la forma funcional que describa la serie que se trata de estudiar en función del tiempo, resulta muy útil la utilización de estimadores no paramétricos. Entre ellos, uno de los más estudiados es el estimador kernel (Härdel and Müller. 1990), (Wand and Jones. 1995).

En general, olvidándonos, de momento, de que nuestra serie es función únicamente del tiempo, consideremos el proceso generador de datos mostrado en (68):

$$(68)\ Y_i = m(X_{1i},...,X_{di}) + \varepsilon_i \quad para \quad i = 1,...,n$$

Donde desconocemos la función $m(.)$ y los valores $X_{ji}(j = 1,...d)$ son realizaciones independientes de variables aleatorias con función de densidad conjunta $f(X_1,...,X_d)$. Los errores de perturbación son i.i.d. con media 0 y varianza igual a $\sigma^2{}_\varepsilon$.

Para cualquier punto $x = (x_1,...,x_d)$ en el dominio de la variable explicativa d-dimensional, el estimador kernel más general de dimensión d de $m(.)$ puede escribirse según (69) :

$$(69)\ \hat{m}(x) = \frac{\sum_{i=1}^{n} K_H(x - x_i) y_i}{\sum_{i=1}^{n} K_H(x - x_i)}$$

Donde H es una matriz simétrica definida positiva $d \times d$, llamada matriz de suavizado y $K_H(x) = |H|^{-1/2} K(H^{-1/2}x)$. $K(.)$ es una función kernel (ó función de pesos multivariante); con lo que cumple $\int K(x)dx = 1$.

La mayoría de las clases utilizadas de estimadores de regresión kernel multivariante aparece con H diagonal, con lo que la expresión anterior se simplifica a (70):

$$(70)\ \hat{m}(x) = \frac{\frac{1}{nh_1 \ldots h_d} \sum_{i=1}^{n} K(\frac{x_1 - x_{1i}}{h_1}, \ldots, \frac{x_d - x_{di}}{h_d}) y_i}{\frac{1}{nh_1 \ldots h_d} \sum_{i=1}^{n} K(\frac{x_1 - x_{1i}}{h_1}, \ldots, \frac{x_d - x_{di}}{h_d})}$$

Tanto en el contexto multivariante como en el contexto univariante, para estimar la función $m(.)$ debemos escoger la función de pesos o kernel y los parámetros de suavizado. La función kernel seleccionada es la dada en (71):

$$(71)\ K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$

Los únicos parámetros en la metodología no paramétrica que hay que estimar son los parámetros de suavizado. En la literatura hay varios criterios, con muy buenos resultados, para obtener dichos parámetros. Con el fin de comprobar si los resultados obtenidos son sensibles a la elección del parámetro de suavizado, es posible utilizar dos criterios ampliamente conocidos: el criterio de validación cruzada generalizada, y el criterio de Rice. Ambos criterios consisten en

minimizar la función *G(h)*, una medida de error penalizada de mínimos cuadrados definida según (72):

$$(72)\ G(h) = RSS(h)\phi(n^{-1}h^{-1})$$

Donde *RSS(h)* es el error de predicción dado en (73):

$$(73)\ RSS(h) = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{m}(x_i))^2$$

Y $\phi(.)$ es la función penalizadora. En el caso univariante, con el criterio de validación cruzada generalizada (GCV), se trataría de minimizar la función dada en (74):

$$(74)\ GCV(h) = \frac{RSS(h)}{\left[1 - \frac{K(0)}{n}\sum_{j=1}^{n}\frac{1}{\sum_{i=1}^{n}K\left(\frac{x_j - x_i}{h}\right)}\right]^2}$$

En el caso del criterio de Rice, la función G(h) a minimizar será el dado en (75):

$$(75)\ RICE(h) = \frac{RSS(h)}{\left[1 - \frac{2K(0)}{n}\sum_{j=1}^{n}\frac{1}{\sum_{i=1}^{n}K\left(\frac{x_j - x_i}{h}\right)}\right]}$$

En ambos casos *K(0)* es el estimador kernel evaluado en el 0. Una vez que dicha función $\hat{m}(x)$ se ha calculado, los extremos locales se identifican en las series, siempre que $Sgn(\hat{m}'(x)) = -Sgn(\hat{m}'(x+1))$ donde $\hat{m}'$ denota la derivada de $\hat{m}$ respecto de x expresada en (76) y *Sgn* es la función signo.

$$(76)\quad m'(x) = \frac{\frac{1}{nh_1^2 \ldots h_3^2}\sum_{i=1}^{n}K'\left(\frac{x_1 - x_{li}}{h_1},\ldots,\frac{x_d - x_{di}}{h_d}\right)y_i}{\frac{1}{nh_1 \ldots h_3}\sum_{i=1}^{n}K\left(\frac{x_1 - x_{li}}{h_1},\ldots,\frac{x_d - x_{di}}{h_d}\right)} - \frac{\left[\frac{1}{nh_1^2 \ldots, h_d^2}\sum_{i=1}^{n}K'\left(\frac{x_i - x_{li}}{h_1},\ldots,\frac{x_d - x_{di}}{hd}\right)\right]\left[\frac{1}{nh_1 \ldots, h_d}\sum_{i=1}^{n}K\left(\frac{x_i - x_{li}}{h_1},\ldots,\frac{x_d - x_{di}}{hd}\right)y_i\right]}{\left[\frac{1}{nh_1 \ldots, h_d}\sum_{i=1}^{n}K\left(\frac{x_i - x_{li}}{h_1},\ldots,\frac{x_d - x_{di}}{hd}\right)\right]^2}$$

La función de pesos usada en la estimación de la derivada es la dada en (77):

$$(77)\ K'(u) = -\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-u^2}{2}\right)u$$

En el proceso de extracción de extremos, el parámetro a seleccionar es, tal como se ha comentado anteriormente, la *h*. Este parámetro determinará la

vecindad. Una vecindad demasiado grande y el resultado puede ser un suavizado que oculte las nolinealidades interesantes. Si la vecindad es demasiado pequeña, el suavizado obtenido será demasiado variable e incluirá, por lo tanto, mucho ruido. A continuación se tratará de estudiar el valor adecuado para este parámetro, observando su repercusión en el proceso de extracción de patrones.

Los datos utilizados para el estudio son series temporales correspondientes a los valores de cierre del índice de precios de las cotizaciones en bolsa de Telefónica (TELEF) e Iberdrola (IBER). Ambas series contiene los datos de las cotizaciones comprendidos en el período comprendido entre el 02/01/1990 y el 25/5/2004. También se utiliza el Índice general de la bolsa de Madrid (IGBM) en el mismo período y la cotización minuto a minuto de Ebay (EBAY) en la bolsa americana. Se han utilizado, además, series sintéticas para algunas de las pruebas.

En el área de la minería de datos el interés no se centra en las características globales de una serie temporal, normalmente se está más interesado en propiedades locales encontradas en subsecciones de la serie temporal a las que denominaremos subsecuencias. Dada una serie temporal $X = x_1,...,x_n$ de longitud n, una subsecuencia S de X es una muestra de longitud $m <= n$ de posiciones contiguas de X de manera que $S = x_p,...x_{p+m-1}$ para $1 <= p <= n-m+1$. El método más sencillo para extraer subsecuencias a partir de una serie temporal larga, será utilizar una ventana deslizante de una anchura determinada, l. De esta manera, lograremos una base de datos de series temporales obtenidas a partir de subsecuencias de longitud l, obtenidas de una serie temporal más larga. Si

posteriormente deseamos extraer patrones se nos presentan, al menos, los siguientes dos problemas:

- Una vez determinada la longitud de la ventana, *l*, sólo se pueden considerar los patrones cuya longitud no supere el tamaño de dicha ventana y esto es un problema desde el momento en el que es sabido que es posible encontrar un mismo patrón con longitudes distintas.
- El uso de esta ventana hace que ciertos patrones se pierdan si la subsecuencia que los contiene queda dividida a lo largo del tiempo.

Atendiendo a los principios del análisis técnico, en las formaciones de cambio o continuación sería normal especificar la duración del patrón en días. Los patrones de larga duración, por ejemplo un patrón de 90 días, generalmente pronostica movimientos de precios para un periodo más largo, comparado a un patrón de 30 días. Además, el usuario podría estar interesado en pronósticos a largo plazo, para pasar, a continuación, a observar más en detalle los pronósticos a corto plazo.

Para suavizar el impacto que pueda producir el hecho de que los patrones resulten partidos por la ventana deslizante, permitiremos solapamientos. Una vez extraída una subsecuencia de longitud, por ejemplo l=60, en lugar de empezar a extraer los datos de la siguiente serie en la posición 61 (lo cual supone que no hay solapamiento), podemos establecer un cierto desplazamiento, d, que con un valor de , por ejemplo d=45, permitiría empezar a extraer la siguiente subsecuencia en la posición 46. El desplazamiento considerado en las bases de datos TELEF e IBER es d=*1*. Esto supone que se han generado todas las subsecuencias posibles.

En IGBM el desplazamiento es $d=l+1$, es decir no hay solapamiento.. En el caso de EBAY, el desplazamiento empleado vendrá dado por la longitud de la serie extraída en la siguiente proporción $d = \frac{l}{4}$.

Extraídas las subsecuencias, se explorarán para poder buscar los patrones expuestos anteriormente. Determinados los puntos importantes de las subsecuencias, se usarán dichos extremos para ver si la subsecuencia contiene alguno de los patrones contemplados. Detectados los patrones, habrá que seleccionar alguno de ellos como el patrón más representativo y adoptar una representación que, en este caso, no será otra que una aproximación lineal a tramos. Se representará cada subsecuencia temporal mediante seis segmentos lineales. Otros trabajos (Perng et al. 2000) han usado funciones cúbicas para realizar la interpolación entre los extremos.

Los seis segmentos pueden estar conectados, en cuyo caso, para cada segmento debemos conservar la longitud y la altura izquierda. La altura derecha se puede obtener mirando en el nuevo segmento. Tendremos, de esta manera, una representación constituida por los siguientes puntos (78):

$$(78) \quad \begin{aligned} A = \{ &< A_0, l_0 >, < E_1, l_1 >, < E_2, l_2 >, \\ &< E_3, l_3 >, < E_4, l_4 >, < E_5, l_5 >, < A_F > \} \end{aligned}$$

Donde E_1, E_2, E_3, E_4, E_5 corresponden a los cinco extremos que constituyen alguno de los patrones mencionados, $l_0, l_1, l_2, l_3, l_4, l_5$ son las longitudes

obtenidas por la proyección de los segmentos sobre el eje x y $A_0 y A_F$ serán, respectivamente, el primer y último valor de la serie.

En ciertos casos, cada subsecuencia presenta más de un patrón, con lo cual es necesario seleccionar alguno de ellos como el más importante, mientras que en otros casos, no aparece ni un solo patrón, con lo cual será necesario adoptar una representación independiente del dominio para dicha subsecuencia temporal. En lo que respecta a las subsecuencia con más de un patrón, la decisión adoptada es la de asignar una prioridad a los patrones. Se considera que el patrón más importante es, por supuesto, el patrón hombro-cabeza-hombro. Si los extremos seleccionados para una determinada subsecuencia temporal cumplen las restricciones impuestas para dicho patrón la representación se adecua a él. En el caso de que esto no suceda, se exploran el resto de los patrones considerados en el siguiente orden: HCHI, MAR, MAB, TRI, TRII, RAR, RAB.

Este orden concede prioridad a los patrones de cambio de tendencia frente a los patrones de continuidad de tendencia, y da prioridad a los patrones que pronostican un cambio de tendencia de ascendente a descendente, frente a los patrones que pronostican un cambio de tendencia de descendente a ascendente. En caso de no haber encontrado un patrón, se extraerán sus tres extremos más importantes que sean máximos E_1, E_3, E_5 , y además dos extremos E_2, E_4 que, siendo mínimos, estén intercalados entre los anteriores máximos de la siguiente manera E_1, E_2, E_3, E_4, E_5 . En cualquiera de los casos, guardaremos información de 6 segmentos lineales. Denominaremos SP a dicho patrón, el patrón que

representará a las subsecuencia en las cuales no se ha encontrado ningún patrón propio del análisis técnico.

La aplicación estará entonces constituida por los elementos mostrados en la figura 5.14, entre los cuales deberíamos distinguir elementos dependientes del dominio y elementos independientes del dominio.

En la figura 5.15 es posible ver gráficamente la representación adoptada. En la figura 5.16 es posible observar una serie temporal y los extremos extraídos en dicha serie.

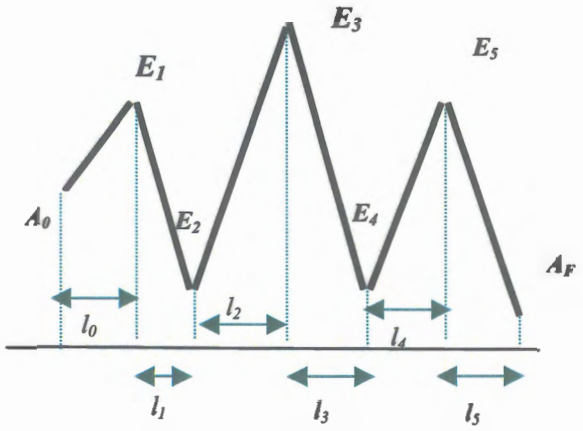


Figura 5. 14. Representación adoptada usando los extremos extraídos de la serie.

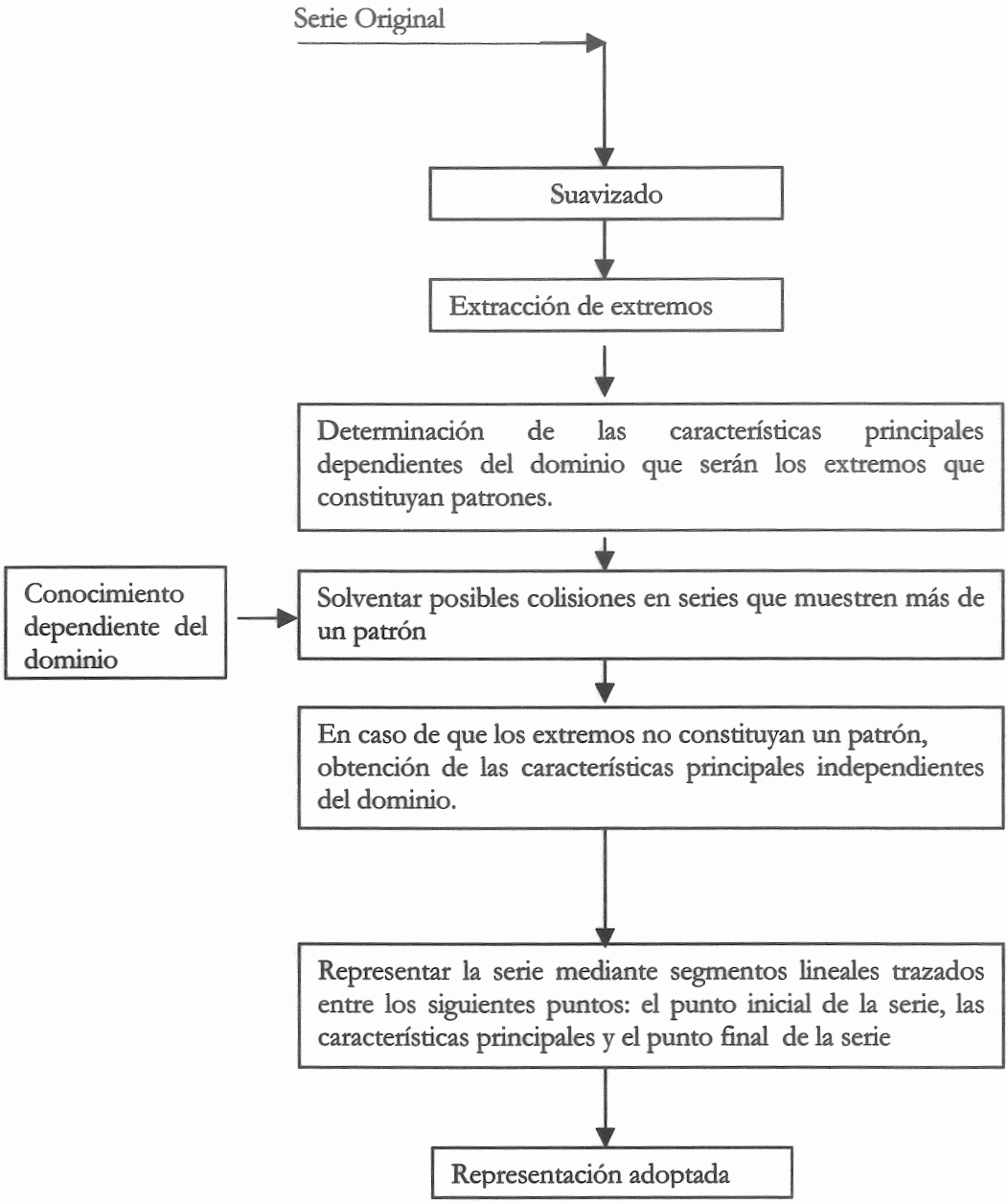


Figura 5. 15. Fases de las que consta el proceso de extracción de patrones y posterior representación.

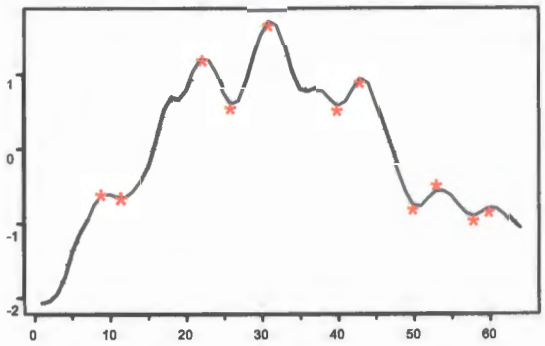


Figura 5. 16. Serie suavizada y los extremos extraídos de la serie, marcados con círculos rojos.

Denominaremos TA a dicha representación. En la figura 5.17 es posible observar tres series temporales y su correspondientes representaciones TA.

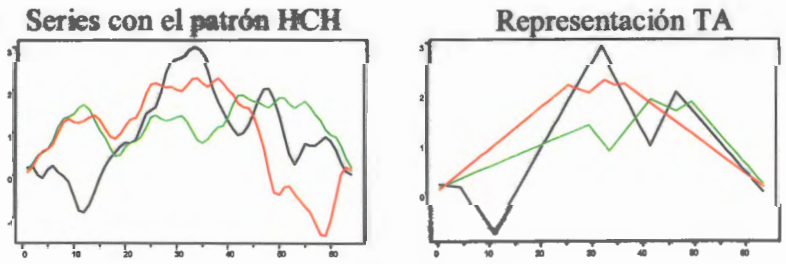


Figura 5. 17. Tres series temporales y su correspondientes representaciones TA.

En una segunda representación, a la que denominaremos EX, se usará el patrón SP para representar a todas las series temporales contenidas en una base de datos. Esta segunda representación es totalmente independiente del dominio de aplicación y se usará para compararla con TA cuando se valoren errores en la

reconstrucción de las series temporales. Es, en realidad, muy semejante a TA pero no está sujeta a las restricciones entre extremos impuestas para ésta.

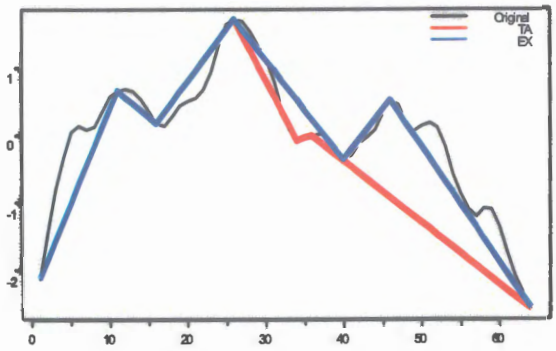


Figura 5. 18. Serie original suavizada con b=0.9 y las representaciones TA y EX.

Si se usasen segmentos lineales discontinuos, con doce coeficientes sería posible conservar información de cuatro segmentos. Es el caso de la representación planteada en (Keogh. 2001a) a la que se denomina SG. Una muestra de dicha representación se puede observar en la figura 5.19.

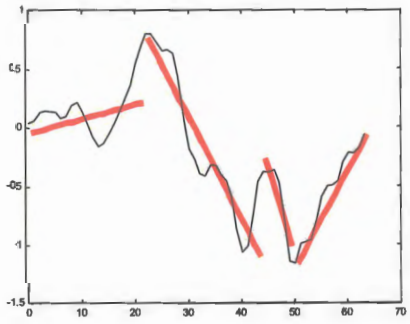


Figura 5. 19. Serie de longitud 64 y su representación SG mediante 4 segmentos.

Una medida del error relativo al reconstruir una serie se define de la siguiente manera: si disponemos de una serie temporal X y su reconstrucción $\hat{X}$, ambas de longitud N, podemos medir la calidad de dicha representación por el error medio de reconstrucción según (79):

$$(79)\ EMR = \sum_{i=1}^{n} \frac{|x_i - \hat{x}_i|}{|x_i|}$$

Otras medidas del error serán las dadas en (80) y (81), siendo en este caso dichas medidas, valores absolutos del error.

$$(80)\ RECM = \sqrt{\frac{1}{N}\sum_{I=1}^{N}(x_i - \hat{x}_i)^2}$$

$$(81)\ EMA = \frac{1}{N}\sum_{I=1}^{N}|x_i - \hat{x}_i|$$

La figura 5.20 muestra los detalles sobre las características de la distribución de los errores en una base de datos de 50 subsecuencias de longitud l=64, extraídas a partir de la serie IGBM. Las representaciones consideradas son TA, EX, la representación agregada a tramos PAA, la Transformada Discreta Wavelet, DWT, y la Transformada rápida de Fourier, FFT.

Se define el ratio de compresión para una serie de longitud original n comprimida a una longitud N según (82).

$$(82)\ R_{compresión} = N/n$$

En la primera base de datos utilizada, el ratio de compresión es de 0.19. Los detalles sobre el preproceso realizado a las series temporales y los valores empleados para el parámetro *h* en el suavizado se pueden consultar en (Basagoiti. 2005).

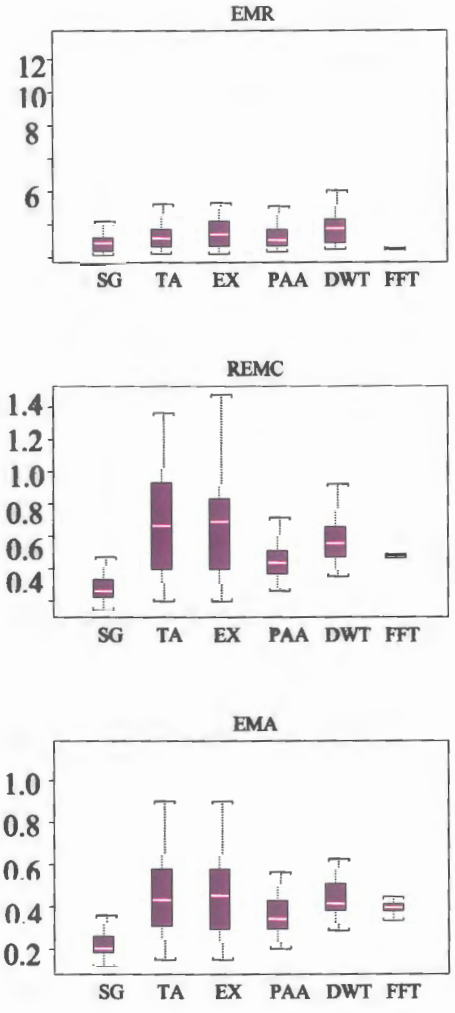


Figura 5. 20. Variabilidad de la mediana en los errores para una base de datos de 50 series de longitud 64 extraídas de IGBM .

Las siguientes dos figuras, 5.21 y 5.22, muestran los errores absolutos acumulados usando la misma base de datos.

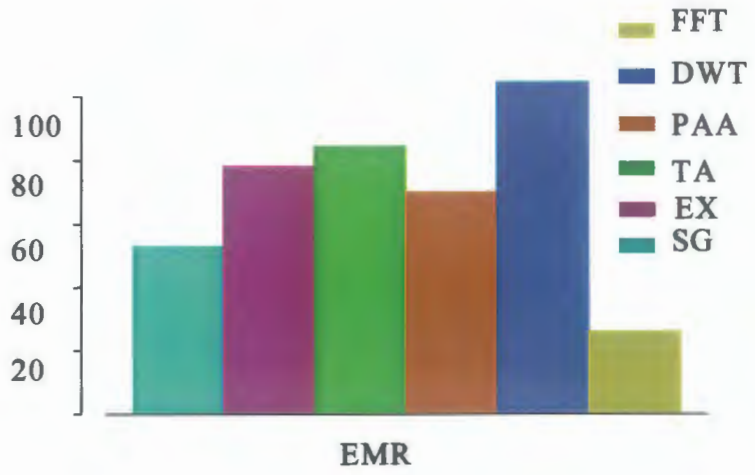


Figura 5. 21. Errores relativos acumulados en una base de datos de 50 series de longitud 64 para las representaciones consideradas.

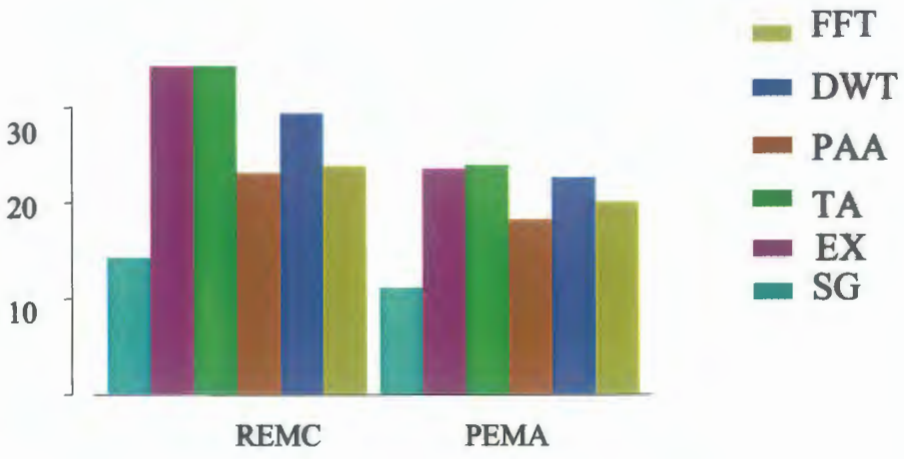


Figura 5. 22. Errores absolutos acumulados en una base de datos de 50 series de longitud 64 extraídas de IGBM para las representaciones consideradas.

Se muestra a continuación, en la figura 5.23, la información sobre la variabilidad de la media en una base de datos de 150 subsecuencias de longitud l=128, extraídas de IGBM y representadas mediante 12 valores. El ratio de compresión es, en esta segunda base de datos, de 0.09.

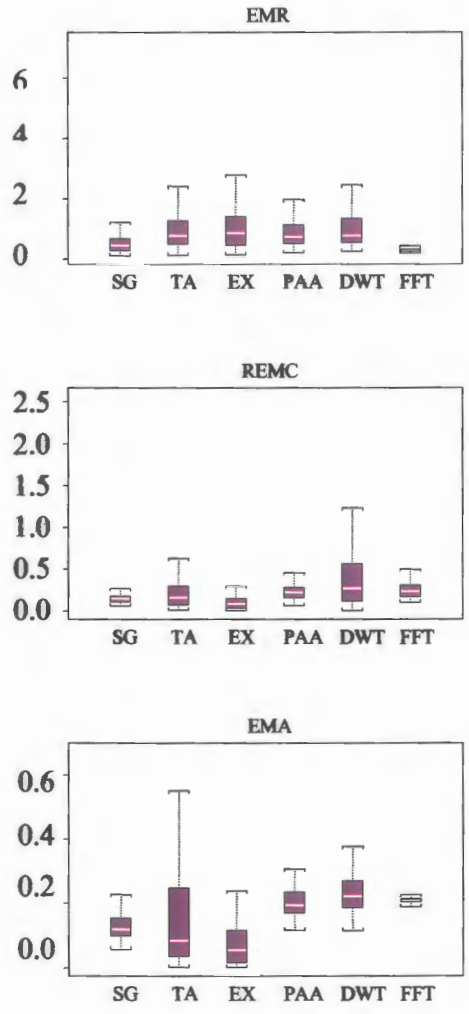


Figura 5. 23. Variabilidad de la mediana en los errores para una base de datos de 150 series de longitud 128 extraídas a partir de IGBM en cada uno de los métodos considerados.

Las figuras, 5.24 y 5.25 muestran los errores absolutos acumulados en la segunda base de datos.

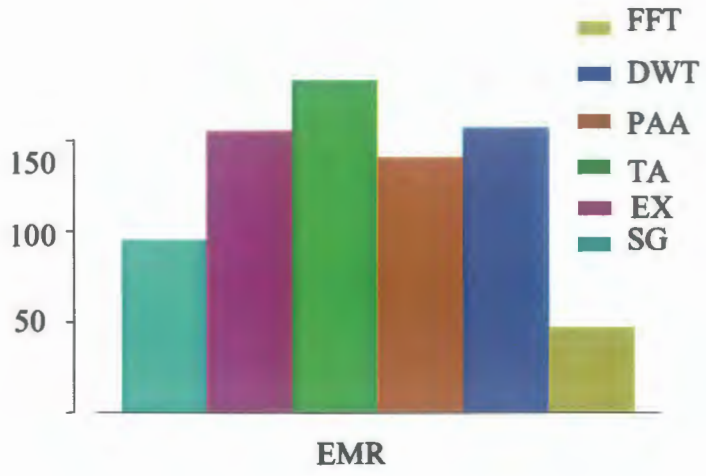


Figura 5. 24. Errores relativos acumulados en una base de datos de 150 series de longitud 128 para las distintas representaciones consideradas.

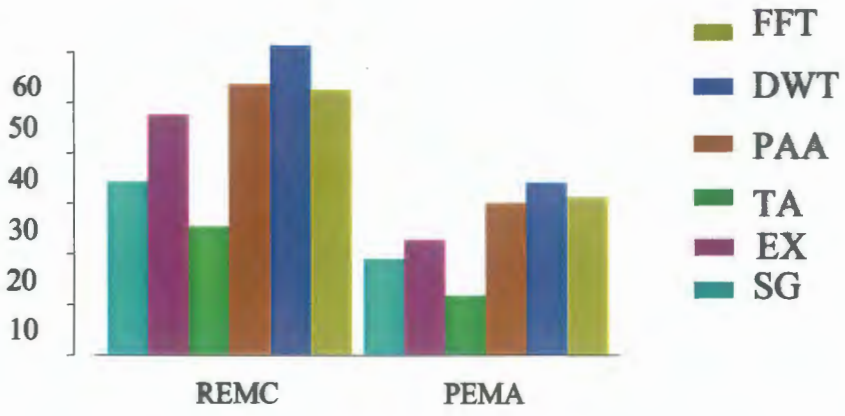


Figura 5. 25. Errores absolutos acumulados en una base de datos de 150 series de longitud 128 para las distintas representaciones consideradas.

Errores de reconstrucción en distintas bases de datos

La representación TA planteada se ha probado inicialmente en dos bases de datos pequeñas, una base de datos de 50 series y otra base de datos de 150 series. Según las recomendaciones dadas en (Keogh and Kasetty. 2002), se han realizado pruebas en bases de datos distintas y de mayor dimensión. Se han generado dos bases de datos con series sintéticas. La primera base de datos está compuesta de series generadas mediante modelos ARIMA. Esta base de datos consta a su vez de 9 bases de datos, disponiendo en cada una de ellas de 1000 series de longitud l=64. La segunda base de datos es una base de datos compuesta de funciones trigonométricas a la que denominaremos TRIG.

Los errores de reconstrucción para estas dos bases de datos se muestran en la figura 5.26.

Además de estas bases de datos, se han utilizado otras dos bases de datos de subsecuencias extraídas de las series IBER y TELEF con subsecuencias de longitud l=64 (el desplazamiento utilizado es d=1). Los resultados se pueden observar en la figura 5.27.

Estos datos no incluyen la comparación con la representación SG que siempre será una representación mejor que TA y EX. Ofrecen, en cambio, la posibilidad de comparar estas dos representaciones a las más comúnmente utilizadas. Los resultados obtenidos en la base de datos sintética y los obtenidos en las bases de datos reales son bastante distintos entre sí. La transformada rápida de Fourier es una buena representación en cualquiera de ellas, pero no es así para la

representación PAA que obtiene resultados más variables en función de la base de datos utilizada.

TA y EX mantienen unos resultados bastantes aceptables en lo que respecta a su comparación con el resto de los métodos. Por otra parte, si comparamos los dos métodos entre sí, la reconstrucción de las series a las que se ha impuesto la representación TA no presenta un error mucho mayor que el de las series con representación EX. Además, los resultados obtenidos en bases de datos más grandes, son mejores para la representación TA que los encontrados en bases de datos más pequeñas.

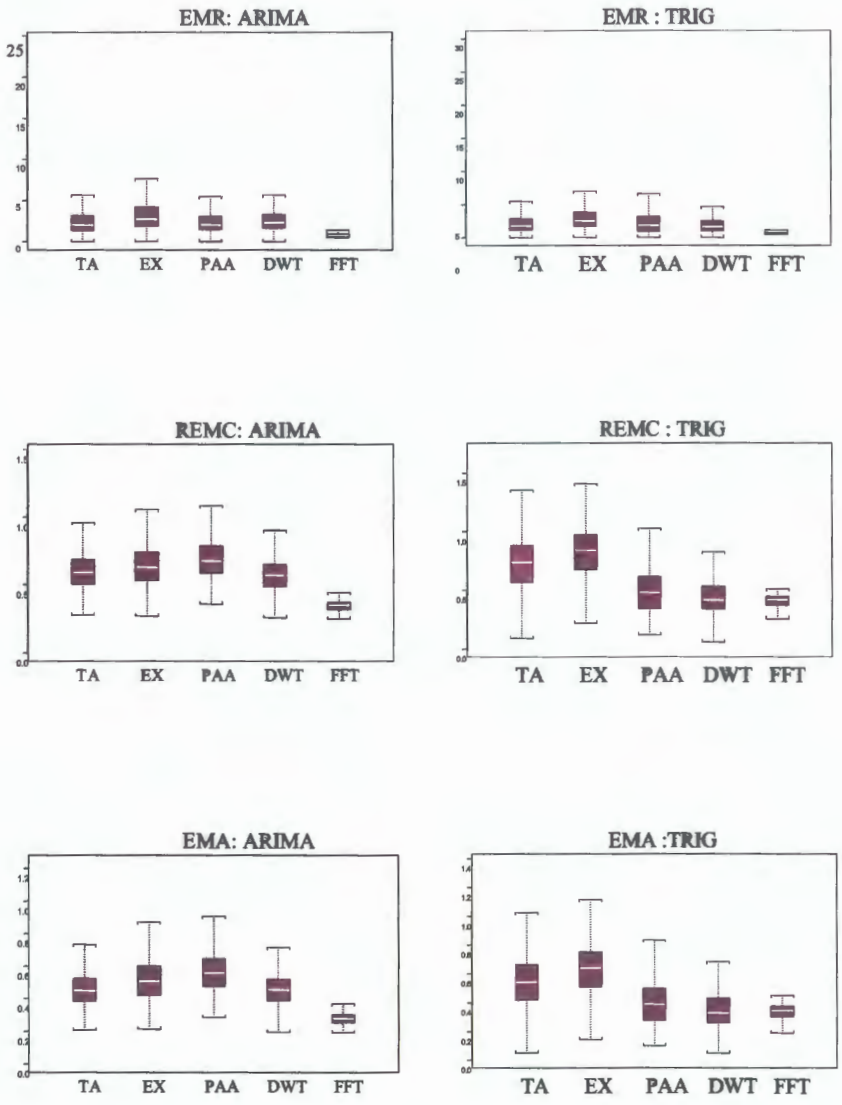


Figura 5. 26. Errores encontrados en la reconstrucción de la base de datos ARIMA y TRIG.

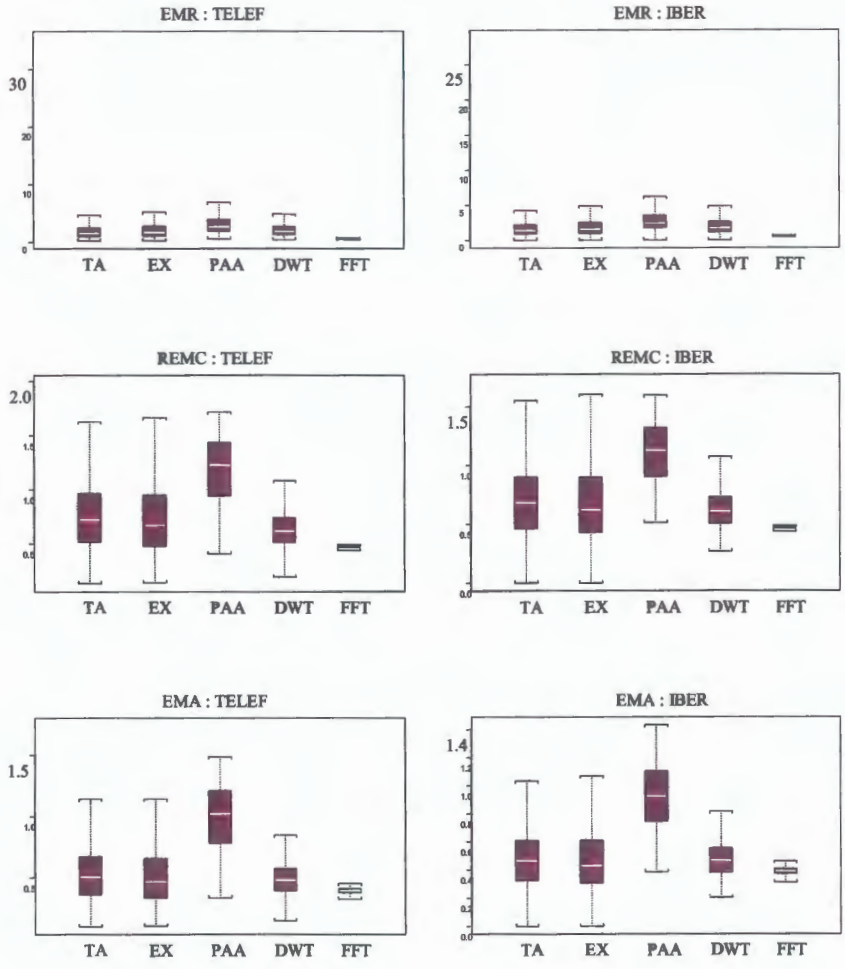


Figura 5. 27. Errores de reconstrucción para distintos métodos en las bases de datos TELEF e IBER.

Extracción de extremos: selección de parámetros

A pesar de que los resultados mostrados no resultaban desalentadores, la representación de muchas de las series no era una representación adecuada desde el momento en el que los patrones encontrados no eran, a veces, de la calidad deseada. La alternativa consiste en el uso de parámetros, aunque no siempre es fácil utilizarlos convenientemente para el fin deseado. Para garantizar la calidad, se usará un parámetro al que se denomina *lgp* que determinará la duración mínima del patrón y que vendrá dado como un porcentaje de la longitud de la serie. En el análisis técnico, los patrones más duraderos son los patrones más fiables. En este sentido, el parámetro servirá para garantizar la calidad de los patrones extraídos.

Con respecto al preprocesado de las series temporales, previo a la extracción de patrones, y con el fin de eliminar el ruido, se realiza un suavizado de las series temporales. Se utilizará el estimador kernel para dicho proceso. El parámetro involucrado en la utilización del estimador kernel es la *h* que determina el número de vecinos implicados en el suavizado. El suavizado es un proceso de suma importancia puesto que la selección del valor adecuado permitirá distinguir entre lo que son los datos y lo que es el ruido. La selección del parámetro *h* será relevante puesto que también determina que punto será considerado un extremo y que punto no será considerado como tal. Todo ello, sin olvidar que los patrones buscados podemos encontrarlos a distintos niveles: podemos estar interesados en buscar patrones a corto plazo, o bien patrones que se hayan desarrollado a lo largo

de periodos de tiempo más prolongados, con lo que su valor predictivo será válido a más largo plazo. Se explorarán, con este fin, distintos tamaños de ventana l . A continuación podemos ver, en la tabla 1, la lista de parámetros utilizados en el proceso de extracción de patrones y la explicación de la función otorgada a cada uno de ellos.

Tabla. 1. Tabla de parámetros considerados

Parámetro	**Explicación**
h	Parámetro de suavizado utilizado en la función kernel.
l	Longitud de ventana utilizada para extraer los patrones.
lgp	% de la longitud de la serie correspondiente al patrón.
n	Número de series.
r	Ancho de la banda de Sakoe-Chiba
i	Intervalos de distorsión para banda de Sakoe-Chiba
k	Número de clusters.
d	Desplazamiento empleado a la hora de extraer la siguiente subsecuencia.

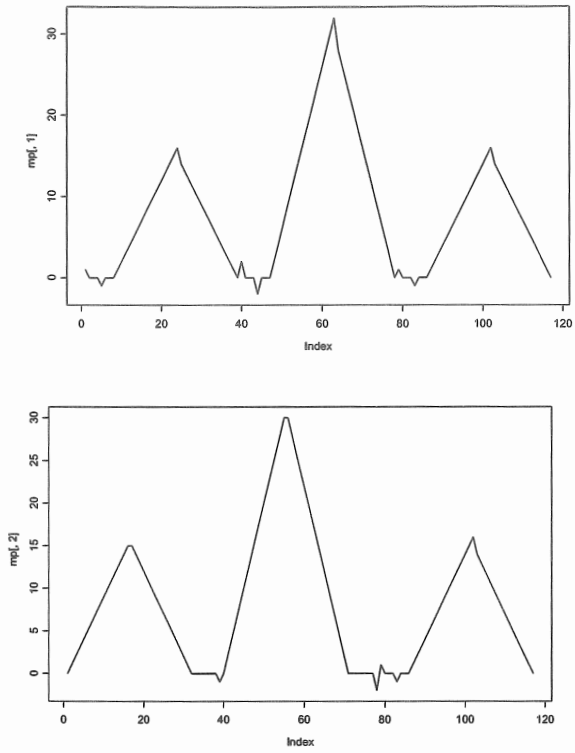


Figura 5. 28. Dos series sintéticas de longitud 120 que a primera vista presentan el patrón HCH.

Tomando como referencia las dos series mostradas en la figura 5.28, más las dos series mostradas en la figura 5.29, podemos observar, a continuación, de que manera puede repercutir la selección de los parámetros *h* y *lgp* a la hora de extraer patrones a partir de una serie temporal. Los resultados son los mostrados en la tabla 2. Para valores de *h* pequeños, tales como 0.3, según se aumenta el porcentaje que determina cuanto de la longitud de la serie debe ser ocupada por el patrón, *lgp*, las series van mostrando patrones diferentes. Es posible pensar que quizá los extremos considerados se encuentren demasiado cerca del ruido. Observando los patrones encontrados para *h=0.6*, éstos se mantienen incluso

cuando aumentamos las restricciones sobre la duración del patrón. Para valores de *h* mayores, tales como 0.9, el número de series que se quede sin patrón puede considerarse demasiado grande.

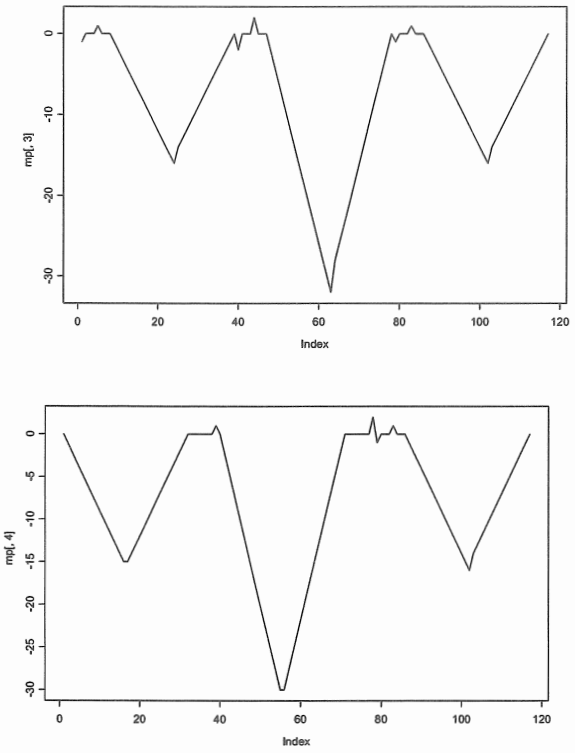


Figura 5. 29. Cuatro series sintéticas de longitud 120 que a primera vista presentan el patrón HCHI,HCHI.

Tabla. 2. Patrones encontrados en las 4 series mostradas para distintos valores de *h* y distintos valores de *lgp*.

	lgp=10%	***lgp=30%***	***lgp=40%***	***lgp=50%***
h=0.3	TTOP BTOP HCHI HCHI	RTOP BTOP HCHI HCHI	SP BTOP SP HCHI	SP SP SP SP
h=0.6	SP SP HCHI HCHI	SP SP HCHI HCHI	SP SP HCHI HCHI	SP SP HCHI HCHI
h=0.9	SP SP SP HCHI	SP SP SP HCHI	SP SP SP HCHI	SP SP SP HCHI

En la siguiente página se muestra la tabla 3, donde podemos ver el impacto del uso del parámetro *h,* de suavizado, y *l,* de longitud de ventana, a la hora de extraer patrones sobre distintas bases de datos extraídas de IGBM. Para *l*=40 hay demasiadas series que se quedan sin patrón, y para longitudes *l*=90 aparecen demasiados patrones por ser la ventana amplia. Sin restricciones sobre el tamaño del patrón *l*=60 podría ser el mas adecuado porque para *l*=40 incluso el suavizado más leve deja demasiadas series sin patrón.

Por lo que podemos observar, y puesto que el orden en el que se realiza la búsqueda de los patrones es el orden en el que se presentan los resultados, según aumentamos *h* y suavizamos más la serie, hay patrones que desparecen. Lo que antes era un patrón HCH ahora ya no lo es y en su lugar surgen otros patrones explorados posteriormente. El recuento de subsecuencias a las que corresponde el patrón SP, ha aumentado en 6, siendo en el caso de *h*=0.6 casi un 26% del total.

Tabla. 3. Patrones encontrados en distintas bases de datos extraídas de IGBM para distintos valores de *h* y *lgp*.

	***h*=0.3**	***h*=0.6**	***h*=0.9**
***l*=40** ***n*=89**	HCH=39 HCHI=10 BTOP=6 BBOT=3 TTOP=6 TBOT=4 RTOP=4 RBOT=0 **SP=17**	HCH =23 HCHI =9 BTOP =6 BBOT =4 TTOP =9 TBOT =9 RTOP =6 RBOT =0 **SP =23**	HCH=24 HCHI=4 BTOP=0 BBOT=1 TTOP=1 TBOT=3 RTOP=2 RBOT=0 **SP=4**
***l*=60** ***n*=59**	HCH=38 HCHI=5 BTOP=4 BBOT=2 TTOP=4 TBOT=0 RTOP=4 RBOT=0 **SP=2**	HCH=28 HCHI=3 BTOP=6 BBOT=3 TTOP=9 TBOT=1 RTOP=4 RBOT=8 **SP=3**	
***l*=90** ***n*=39**	HCH=33 HCHI=1 BTOP=0 BBOT=0 TTOP=1 TBOT=1 RTOP=0 RBOT=0 **SP=0**	HCH=32 HCHI=1 BTOP=0 BBOT=0 TTOP=1 TBOT=1 RTOP=0 RBOT=0 **SP=0**	

La calidad de los patrones extraídos es también cuestionable. Con el fin de mejorar este aspecto, se ha observado el porcentaje de las series ocupadas por el patrón. En la tabla 4 podemos ver los resultados que al respecto se han obtenido en sucesivas bases de datos que para distintas longitudes se han extraído a partir de IGBM.

Tabla. 4. Porcentaje global de las series ocupadas por el patrón.

l=40 *h*=0.3	*l*=40 *h*=0.6	*l*=60 *h*=0.3	*l*=60 *h*=0.6	*l*=90 *h*=0.9
30.8	37.6	20.5	25.3	26.8

Se han observado las longitudes medias de los patrones de manera que la cantidad que a continuación se muestra, en la tabla 5, es el porcentaje de la serie que ocupa el patrón. En la misma tabla se muestra el número de series que quedan sin una representación propia de los patrones de análisis técnico considerados.

El uso de un parámetro *lgp*, será necesario para controlar la duración del patrón. Mediante los distintos valores adjudicados a este parámetro, 15%, 20% y 25%, se realizará un filtrado de los patrones encontrados en cada serie. Si un patrón no tiene una duración que supere el 15%, 20% y 25% ,respectivamente, de la longitud de la serie, quedará descartado. El mecanismo servirá para descartar patrones poco representativos. Según las pruebas exploratorias realizadas, valores para *lgp* superiores al 25% no eran adecuados, debido al gran número de series que quedaban sin patrón.

Tabla. 5. Porcentaje de las series ocupadas por los distintos patrones para distintas longitudes *l*, número de series *n* y parámetros de suavizado *h*.

% de la longitud de las series contenida en el patrón	***l*=40 *n*=89 *h*=0.3**	***l*=40 *n*=89 *h*=0.6**	***l*=60 *n*=59 *h*=0.3**	***l*=60 *n*=59 *h*=0.6**
***lgp*=25%**	SP=46 38.8%	SP=41 40.6%	Sp=36 39.5%	SP=30 37%
***lgp*=20%**	SP=36 35.6%	SP=30 39.5%	SP=22 34.7%	SP=14 27.5%
***lgp*=15%**	SP=35 35.6%	SP=28 38%	SP=6 24.2%	SP=7 26.8%

Según este planteamiento, el proceso de extracción de extremos para la posterior obtención de la representación TA, vendrá determinado por los valores de *l, lgp* y *h* .

Para poder evaluar el impacto que produce la variación de éstos parámetros se han programado unas pruebas con el objetivo de obtener algo de información sobre la interrelación existente entre ellos.

La serie utilizada para generar estas bases de datos es la cotización minuto a minuto de Ebay correspondiente al período que va desde el 11/11/2002 al 12/9/2003.

Si se determinan para la variable *l* los valores (40.90,180,360), en realidad estaremos utilizando una ventana temporal que equivale a aproximadamente media hora, hora y media, tres horas y un día de cotización.

Una vez determinados los valores para *l,* era necesario seleccionar unos valores para la variable *h* que permitiesen extraer ciertas conclusiones. Los valores de *h* estudiados serán *h*=(0.6,0.9,1.4).

Quedan por determinar los valores que se desean asignar a la variable *lgp*. Los valores seleccionados para este parámetro no son muy restrictivos, *lgp* = (15%, 20% , 25%).

Considerando todas las combinaciones posibles entre los tres parámetros mencionados, el número de bases de datos a tratar asciende a 36.

En las pruebas realizadas, el número de subsecuencias consideradas será en todos los casos fijo, *n*=250 siendo el desplazamiento empleado en el proceso de extracción variable en función de la longitud de las series, *d*=*l*/4.

Una vez generadas las 36 bases de datos y extraídos los extremos, se ha llevado a cabo el posterior proceso de representación. Obtenida la representación TA para cada serie, es posible observar ciertos aspectos no considerados hasta el momento, tales como el número medio de extremos encontrado en cada base de datos o el coeficiente de variación del número de extremos.

Los resultados se muestran en la figura 5.30. El número medio de extremos va decreciendo, tal como era previsible, según aumenta el valor del parámetro de suavizado.

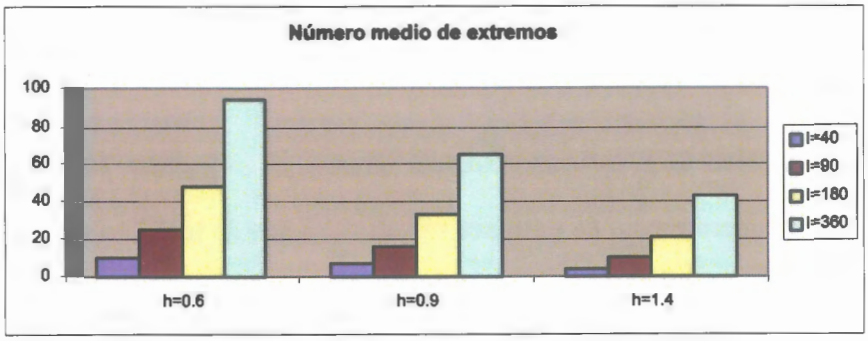


Figura 5. 30. Número medio de extremos teniendo en cuenta distintos valores de *h* y *l*.

En lo que respecta al coeficiente de variación del número de extremos, es más estable en las bases de datos con series más largas, presentando valores mayores en las series más cortas, tal como es posible ver en la figura 5.31.

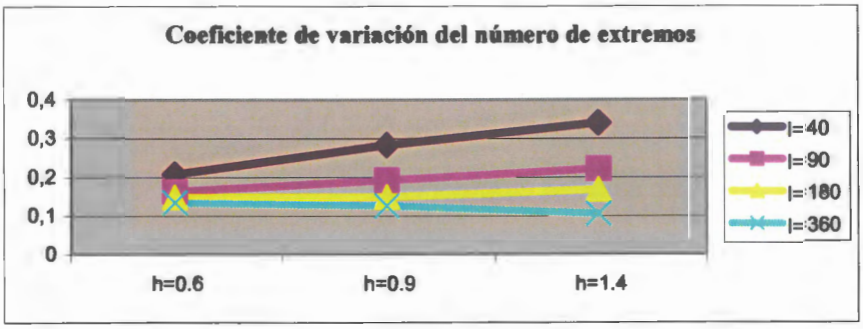


Figura 5. 31. Coeficiente de variación del número de extremos.

Una vez extraídos los extremos y detectados los patrones, es posible evaluar el número medio de patrones encontrado para cada base de datos.

Sabemos, por lo dicho anteriormente, que, hay series que presentan más de un patrón, pero sabemos que también es posible que haya series que ni siquiera tengan cinco extremos.

El número medio de patrones se calculará, por lo tanto, sumando el numero de patrones encontrado en cada base de datos, teniendo en cuenta que hay series que presentan más de un patrón y teniendo en cuenta todos ellos y dividiendo dicha cantidad por el número de series que tienen más de cinco extremos. No se consideran, para el cálculo de dicha media, las series que tengan menos de cinco extremos, puesto que en dichos casos es imposible detectar la presencia de ningún patrón. Los algoritmos de clustering que se utilizarán posteriormente se beneficiarán del hecho de que se hayan borrado dichos casos. Estos se podrían considerar separadamente como pertenecientes a otra clase. Eliminar dichos casos nos ofrece la posibilidad de seguir estudiando al resto.

En las figuras 5.32 y 5.33 es posible observar los resultados encontrados a lo largo de las 36 bases de datos consideradas.

Es destacable la diferencia que presentan los resultados obtenidos para valores de $l=40$ con respecto a los resultados obtenidos para el resto de las longitudes. En estas series, el número medio de patrones para h=0.6 es superior al número medio de patrones para otros valores (h=0.9 y h=1.4) para los valores de lgp= 15% y lgp=20%, manteniéndose los valores muy parecidos para lgp=25%.

Según vamos aumentando la longitud de las series, l, esto no volverá a suceder, el número medio de patrones se mantendrá por encima del resto cuando $h=1.4$.

Como conclusión podemos decir que, a pesar de que con valores de h más pequeños obtenemos más extremos, ello no contribuye a que en dicha serie se encuentren más patrones. Se podría concluir que muchos de los extremos extraídos impiden ver el patrón subyacente.

Los resultados obtenidos para series de longitud l=360 son también bastante especiales. Para suavizados con h=0.6 y para restricciones de lgp=25% el número medio de patrones es en realidad muy cercano a 0. No ha sido posible encontrar ningún patrón.

El uso de un valor de h más grande permite salvar dicha dificultad permitiendo un mayor suavizado de la base de datos. El mayor suavizado permite encontrar patrones que cumplan las restricciones impuestas para la longitud.

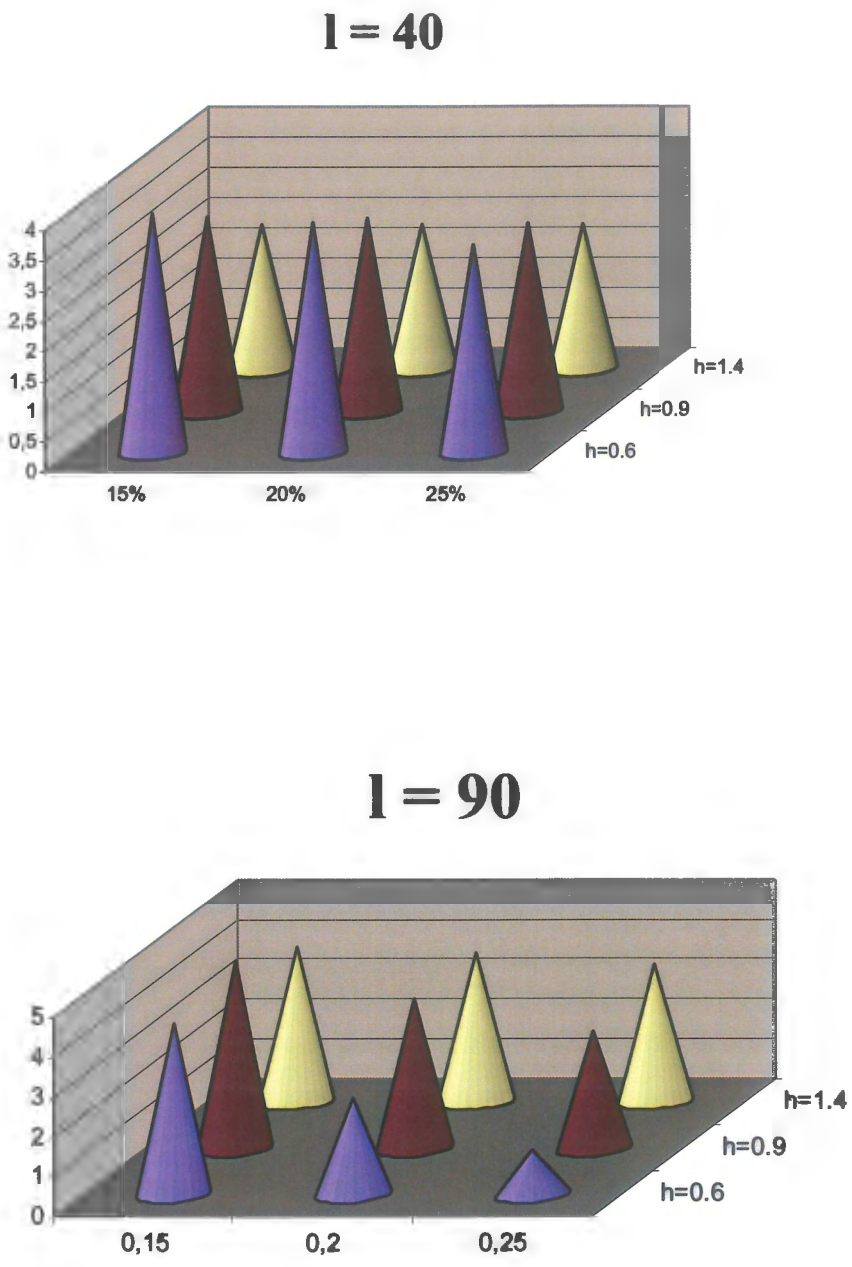


Figura 5. 32. Número medio de patrones encontrados para *l=40,90*, parámetro de suavizado *h* y mínima longitud de patrón *lgp*.

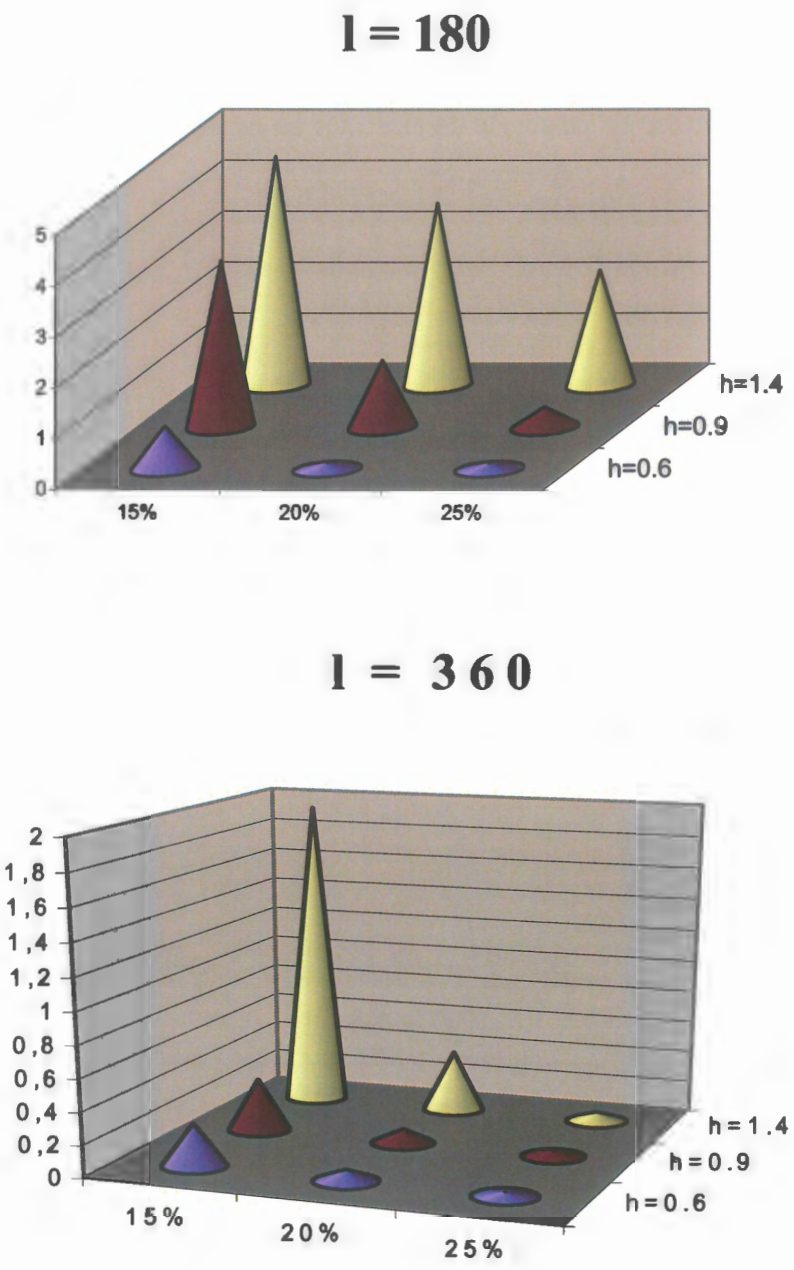


Figura 5. 33. Número de patrones encontrados para *l=180,360*, parámetro de suavizado *h* y mínima longitud de patrón *lgp*.

Además de las consideraciones realizadas sobre el número de extremos y el número de patrones, es importante considerar la calidad de la segmentación obtenida. El proceso de detección de extremos ha generado una segmentación de las series temporales una vez seleccionado el patrón adecuado para su representación TA. A la hora de evaluar la calidad de la segmentación obtenida, se ha considerado la desviación estándar de las longitudes de los segmentos. Los valores mostrados a continuación en las figuras 5.34 y 5.35 tienen en consideración las segmentación de una serie temporal sólo si contiene alguno de los patrones considerados.

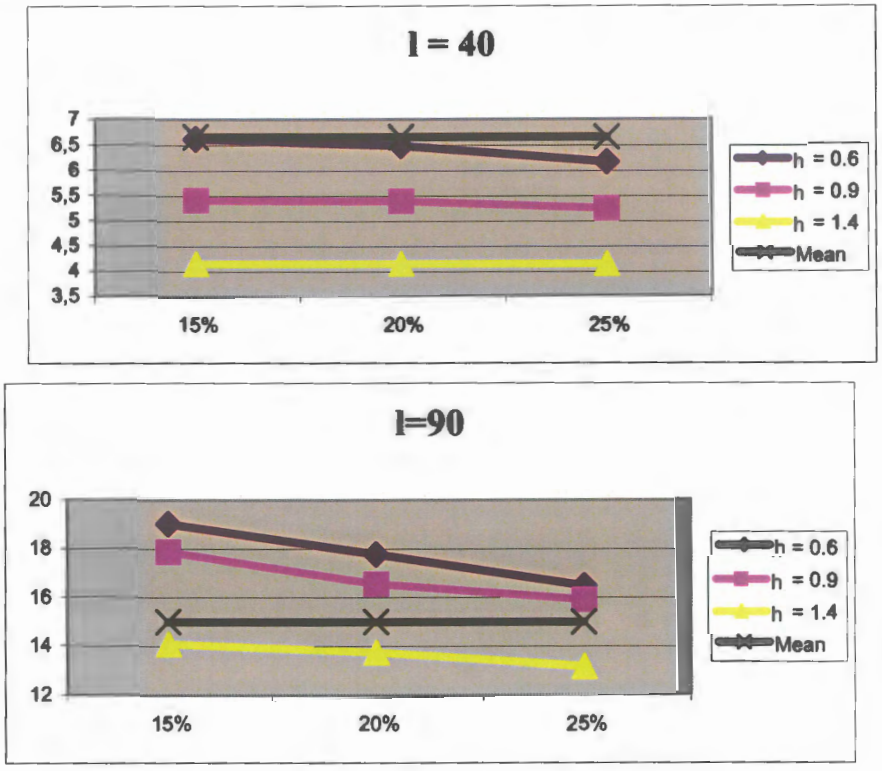


Figura 5. 34. Desviación estándar de las longitudes de los segmentos para distintos valores de *l=40,90, lgp* y *h*

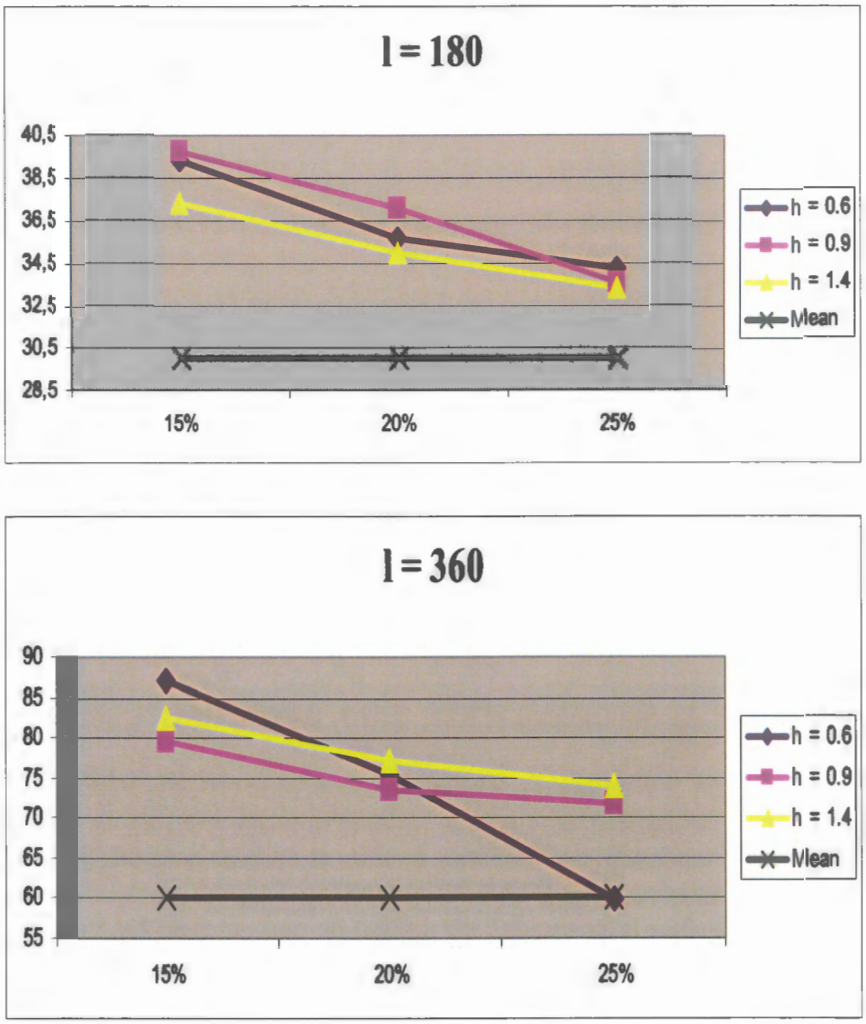


Figura 5. 35. Desviación estándar de las longitudes de los segmentos para distintos valores de *l=180,360, lgp* y *h*

Según se puede observar en las anteriores dos figuras para *l*=40 Y *l*=90 las bases de datos con el parámetro de suavizado más grande son las que mantienen los tamaños de los segmentos más estables con respecto a la media.

En lo que respecta a l=360, es necesario recordar que no se encontraron patrones para h=0.6 y lgp=25% con la repercusión que ello tiene en la segunda de las gráficas mostrada en la figura 5.35 donde para h=0.6 la desviación estándar de la longitud de los segmentos cae a la media para lgp=25% . En realidad, en este caso no se han encontrado patrones y es imposible evaluar la segmentación.

Según la representación adoptada, TA, disponemos de los segmentos previos y posteriores a los patrones encontrados. Puesto que en el análisis técnico éstos patrones adquieren un significado como predictores de un cierto comportamiento del mercado, la siguiente consideración realizada es una evaluación de la capacidad predictiva de cada tipo de patrón.

Los patrones utilizados se clasifican como pertenecientes a tres categorías: predictores de continuidad (TRIT, TRIB, RTOP, RBOT), predictores de cambio de tendencia descendente a tendencia ascendente (HCHI, BBOT), y predictores de cambio de tendencia ascendente a tendencia descendente (HCH, BTOP).

Para evaluar la capacidad predictiva de cada uno de los patrones extraídos, se estudia la pendiente de los movimientos observados antes y después de los patrones.

En la figura 5.36 observaremos el porcentaje de éxito obtenido para cada categoría a lo largo de las 36 bases de datos.

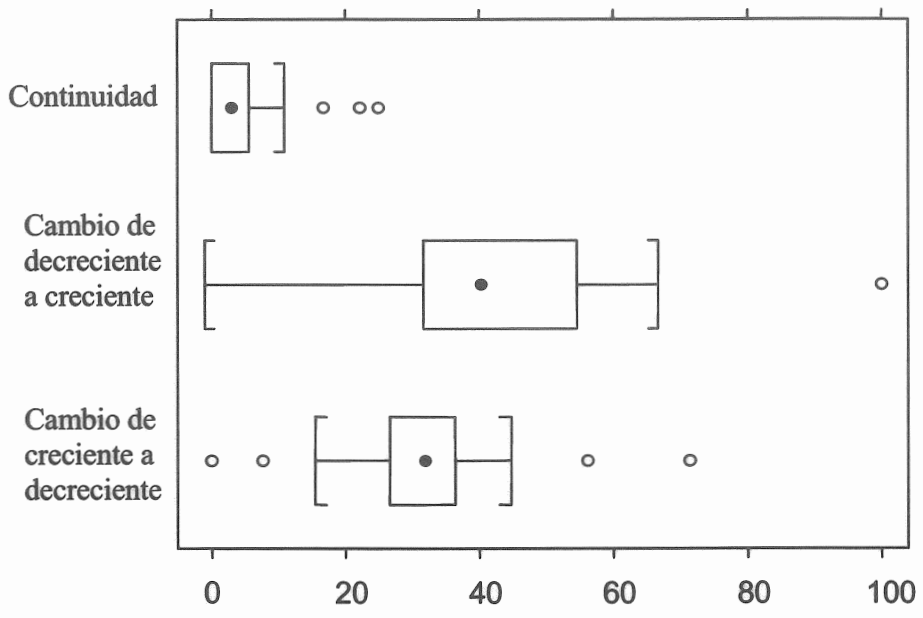


Figura 5. 36. Boxplot de la capacidad predictiva de los patrones encontrados a lo largo de las 36 bases de datos consideradas.

Una visión más detallada de la capacidad predictiva, según las restricciones impuestas para la duración del patrón, es la mostrada en la figura 5.37.

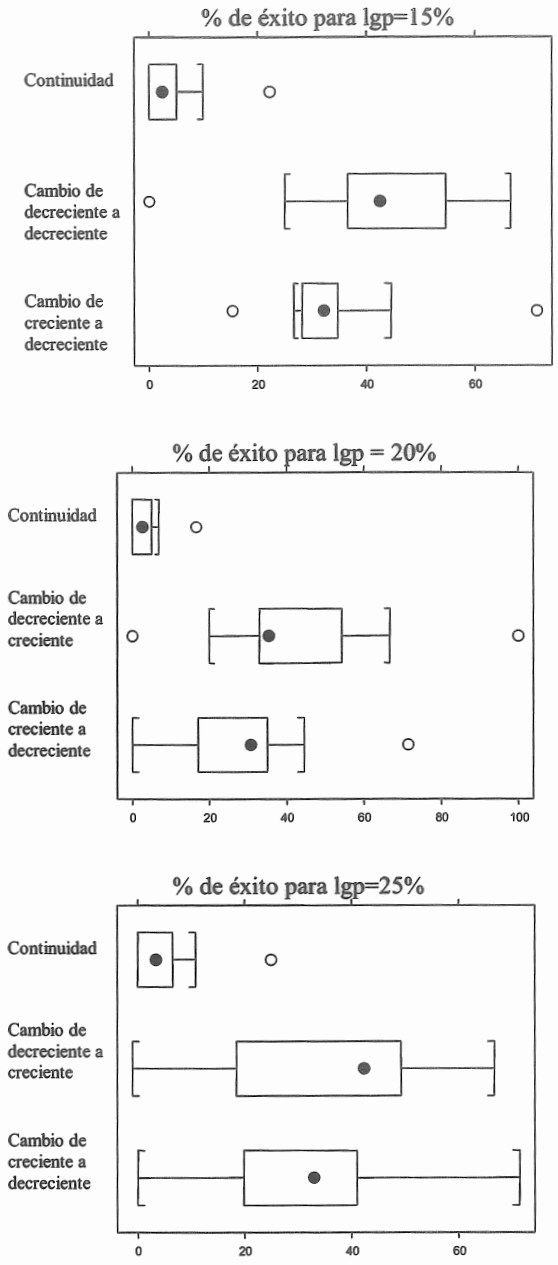


Figura 5. 37. Boxplot de la capacidad predictiva de los patrones encontrados para los distintos valores de *lgp*=(15%,20%,25%).

CAPÍTULO 6

CLUSTERING DE SERIES TEMPORALES

En este capítulo exploraremos distintas medidas de la distancia prestando especial atención a los clusters generados por éstas. Los clusters obtenidos permitirán recuperar las series más semejantes entre sí. Para la representación TA, estas semejanzas se obtendrán en base a la segmentación realizada, usando para ello la distancia DS, que medirá el grado de paralelismo de los segmentos que constituyen una serie temporal. Las series de datos económicos presentan características propias que las diferencian de otras series temporales desde el momento en el que pueden venir caracterizadas por unos cuantos puntos importantes, y presentan, además, características multiresolución desde el momento en el que puede interesarnos realizar un análisis a largo plazo o bien un

análisis a corto plazo puesto que los patrones de análisis técnico se repiten constantemente con distinta amplitud y/o duración.

Distancias entre segmentos y su uso en series que guardan patrones de análisis técnico

En el anterior capítulo hemos visto cómo aproximar una secuencia mediante segmentos lineales. Además, será necesario utilizar una medida de la distancia que nos permita resolver los problemas de minería de datos anteriormente planteados, la búsqueda de subsecuencias similares, o el clustering, entre otros. La distancia Euclídea es una distancia sencilla y muy potente, pero que no es capaz de considerar que dos series temporales iguales desplazadas en el tiempo sean semejantes, algo que sería deseable en el dominio que nos ocupa, donde dos series con la ocurrencia de un mismo patrón desplazado en el tiempo, pueden ser consideradas iguales.

Extraeremos patrones que estarán constituidos por segmentos de distinta longitud, cosa que debemos tener en cuenta a la hora de calcular las distancias. La distancia planteada en (Keogh. 1998), a la que llamamos DS, es una distancia entre segmentos de la misma longitud. Las cotas inferiores de ADT son también distancias que se calculan entre segmentos de la misma longitud, de manera que, cuando dos segmentos a ser comparados no sean de la misma longitud será necesario realizar interpolación lineal para adecuar las longitudes de los segmentos.

En la figura 6.1 se dan tres series con distorsión en el eje temporal que nos servirán para posteriores pruebas. Los valores son los mostrados a continuación:

s1=(0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,15,14,13,12,11,10.9,8,7,6,5,4,3,2,1,0)
s2=(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,15,14,13,12,11,10.9,8,7,6,5,4,3,2,1,0,0,0,0,0,0,0,0)
s3=(0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,15,15,15,14,13,12,11,10.9,8,7,6,5,4,3,2,1)

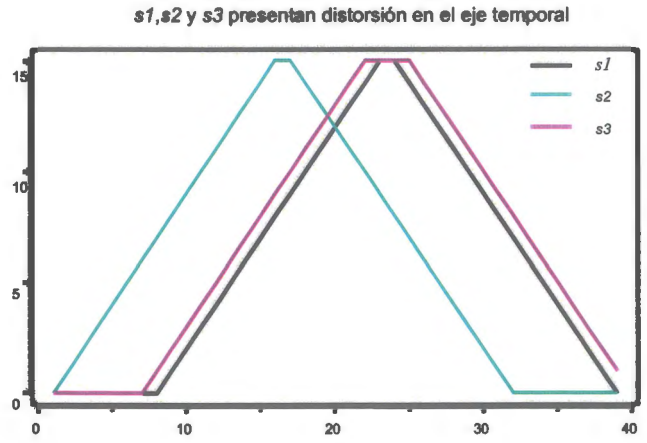


Figura 6. 1. Tres series, *s1, s2, s3* que presentan distorsión en el eje temporal.

Usando la base de datos IGBM, en las siguientes dos gráficas se muestran las subsecuencias más parecidas a una serie dada usando la distancia ADT y la distancia DS sobre la representación TA respectivamente.

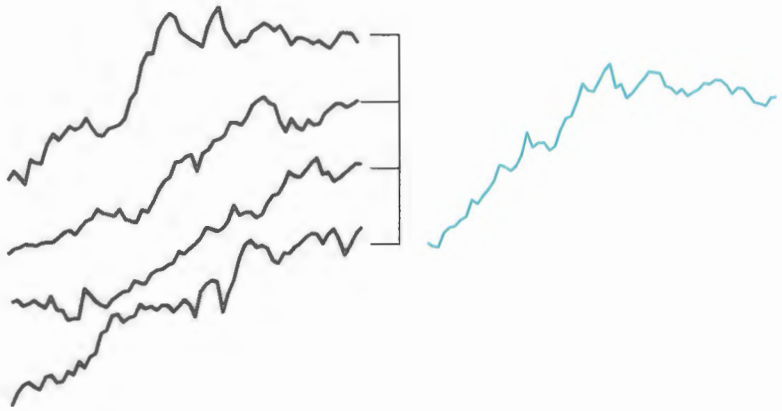


Figura 6. 2. Una serie de la base de datos y las 4 series más parecidas a la serie dada encontradas en la base de datos usando como distancia ADT.

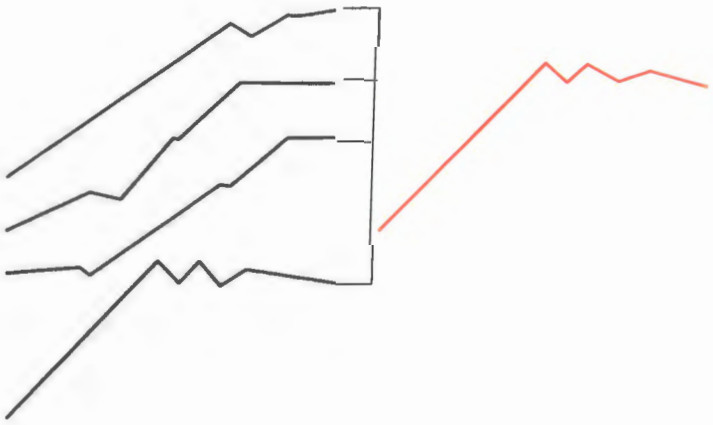


Figura 6. 3. Una serie de la base de datos en la representación TA y las 4 series más parecidas a la serie dada(también en la representación TA) encontradas en la base de datos usando para ello DS.

Distancia entre segmentos

Definimos la similitud entre dos segmentos, de la misma longitud y alineados, según la medida de la similitud propuesta en (Keogh. 1998).

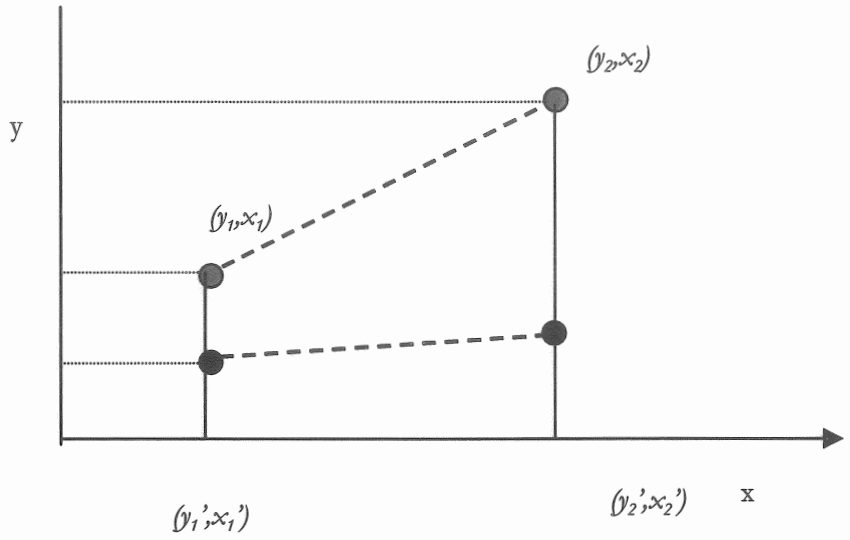


Figura 6. 4. Representación gráfica de la medida de la similitud entre segmentos propuesta en (Keogh, Eamonn 1998).

Dados dos segmentos representados por las coordenadas del punto inicial y las coordenadas del punto final:

$$A = < (y_1, x_1), (y_2, x_2) >$$
$$B = < (y_1', x_1'), (y_2', x_2') >$$

Una medida sopesada de la distancia entre los puntos correspondientes a dichos segmentos será la definida según (83). AW y BW serán los pesos asignados a los segmentos A y B respectivamente.

$$(83)\ DS(A,B) = AW * BW * |(y_1 - y_1') - (y_2 - y_2')|$$

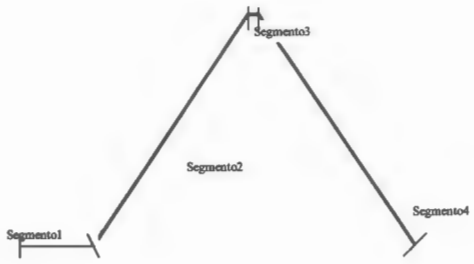
La distancia dada cumple las siguientes propiedades:

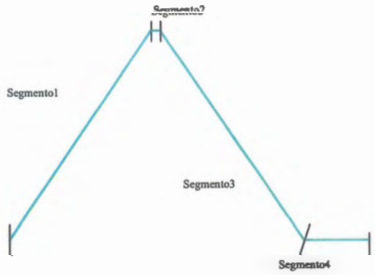
- $DS(A,A)=0$.
- $DS(A,B)=DS(B,A)$.
- $DS(A,B)=0$ si y sólo si $A=B$ o A y B están desplazados entre sí en el eje x o A y B están desplazados entre sí en el eje y.

Usando las series *s1,s2,s3*, dadas en la figura 6.1, se obtienen otras tres series con una representación lineal a tramos a los que denominaremos *S1*, *S2* y *S3*. Se guarda información de 4 segmentos contenidos en cada serie. Para cada segmento, será necesario almacenar cuádruplas del tipo (valor inicial, posición inicial, valor final, posición final) pero es posible que, el segundo segmento y subsiguientes, busquen su valor y posición inicial en el valor y posición final del anterior segmento. De esta manera, tendremos las representaciones mostradas en la tabla 6. La representación gráfica correspondiente será la mostrada en la figura 6.5.

Tabla. 6. Segmentación de las series *s1*, *s2*, *s3* dadas en la figura 6.1. Los valores mostrados son pares: en negrita el valor y a continuación la posición..

s1=(<**0**,0>,<**0**,8>,<**15**,25>,<**15**,26>,<**0**,39>)
s2=(<**0**,0>,<**15**,16>,<**15**,17>,<**0**,32>,<**0**,39>)
s3=(<**0**,0>,<**0**,7>,<**15**,23>,<**15**,24>,<**0**,39>)





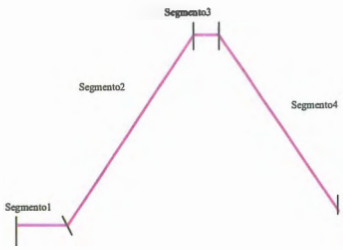


Figura 6. 5. Tres series, *s1 ,s2 y s3* y su división en segmentos.

Para el cálculo de la distancia, los segmentos deben tener la misma longitud y estar alineados. Para conseguirlo, se utilizará la interpolación lineal. Si disponemos de dos segmentos:

$$Sg1 = (< y_1, x_1 >, < y_2, x_2 >)$$

$$Sg2 = (< y'_1, x'_1 >, < y'_2, x'_2 >)$$

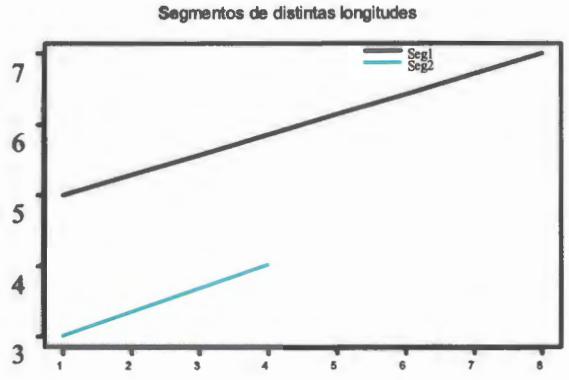


Figura 6. 6. Dos segmentos de distinta longitud.

Queremos obtener:

$$Sg1_f = (< y_{1f}, x_{1f} >, < y_{2f}, x_{2f} >)$$

$$Sg2_f = (< y'_{1f}, x'_{1f} >, < y'_{2f}, x'_{2f} >)$$

Una alternativa será estirar los segmentos a izquierda y a derecha hasta que ambos tengan la misma longitud. Una vez realizada la interpolación, tendremos que (84) es cierto.

$$(84)\ x_{1f} = x'_f = \min(x_1, x'_1) =_1 y\ x_{2f} = x'_{2f} = \max(x_2, x'_2).$$

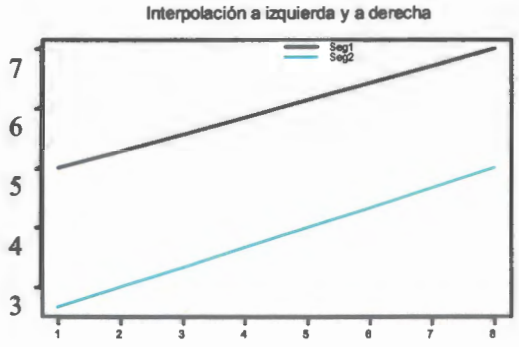


Figura 6. 7. Los dos segmentos de la figura anterior después de interpolar a la izquierda y a la derecha..

La segunda posibilidad es la de ajustar los segmentos a la derecha, sin interpolar, según (85), y en lo que respecta a la izquierda, se interpolará hasta que los segmentos tengan la misma longitud y (86) sea cierto.

$$(85)\quad \begin{array}{l} si \quad x_2 >= x_2' \quad entonces \quad x_{2f}' \leftarrow x_2 \quad y \quad x_{2f} \leftarrow x_2. \\ si \quad x_2' > x_2 \quad entonces \quad x_{2f} \leftarrow x_2' \quad y \quad x_{2f}' \leftarrow x_2'. \end{array}$$

$$(86)\quad \begin{array}{l} x_{1f} \leftarrow \min(x_1, x_1') \\ x_{1f}' \leftarrow \min(x_1, x_1') \end{array}$$

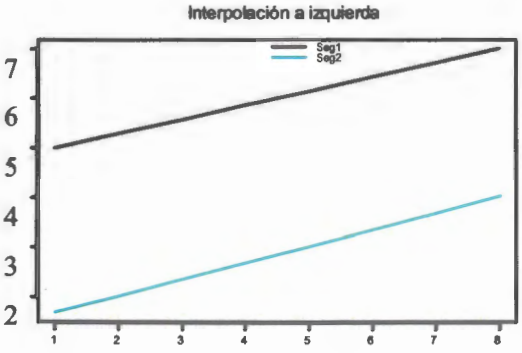


Figura 6. 8. Los dos segmentos de la figura 6.6 después de ajustar a la derecha e interpolar a la izquierda.

Siempre que tenemos una representación mediante segmentos lineales podemos pensar en la posibilidad de determinar los pesos de los segmentos. El sopesado de los segmentos es un problema difícil de resolver automáticamente. En este caso, el sopesado se realiza de la siguiente manera, se mira el quartil en el que se sitúa cada segmento en función de la longitud de su proyección en el eje x, teniendo presentes todos los segmentos mediante los cuales se vaya a representar la serie temporal. A continuación, los pesos para cada segmento se calcularán en función de lo mostrado en la tabla 7.

Tabla. 7. Peso adjudicado a cada segmento según el quartil al que pertenezca su longitud.

Longitud del segmento = *ls*	**Peso**
ls< q1	0.5
q1 < *ls*< q2	0.9
q2 < *ls*< q3	1.1
q4 < *ls*	1.3

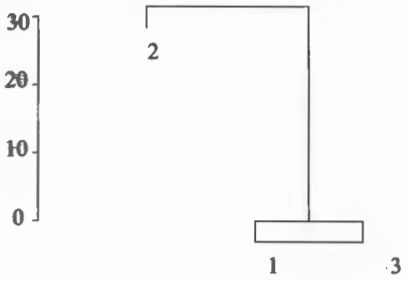
En la tabla 8 es posible observar el resultado del sopesado para las tres series *s1*, *s2* y *s3*. A continuación, en la figura 6.9, el dendrograma correspondiente al clustering jerárquico de dichas series, usando como distancia DS con sopesado.

Utilizaremos dendrogramas para apreciar y evaluar mejor el efecto producido por las similitudes dadas.

La similitud entre dos objetos en un dendrograma es representado como la altura del nodo más bajo que ambos nodos comparten y es una buena herramienta puesto que un dendrograma poco intuitivo nos hará desconfiar de la medida de la similitud utilizada.

Tabla. 8. Pesos obtenidos para los segmentos extraídos de *s1*, *s2* y *s3* dados en la tabla 6 .

	Longitudes segmentos	Quartiles 25%	50%	75%	100%	Pesos w1	w2	w3	w4
s1	8, 17, 1, 13	6	10	14	17	0.9	1.5	0.5	1.1
s2	16, 1, 15, 7	6	11	15	16	1.5	0.5	1.3	0.9
s3	7, 16, 1, 15	6	11	15	16	0.9	1.5	0.5	1.3



DS(*s1*,*s2*)	31.5
DS(*s1*,*s3*)	0
DS(*s2*,*s3*)	31.5

Figura 6. 9. Clustering jerárquico de las series de *s1*, *s2* y *s3* usando DS con sopesado.

Utilizando la distancia definida en (83) y sopesando los segmentos según el criterio dado en la tabla 8, se han obtenido dos ejemplos de las series más similares entre sí de la base de datos IGBM. Se muestran, a continuación, en la figura 6.10.

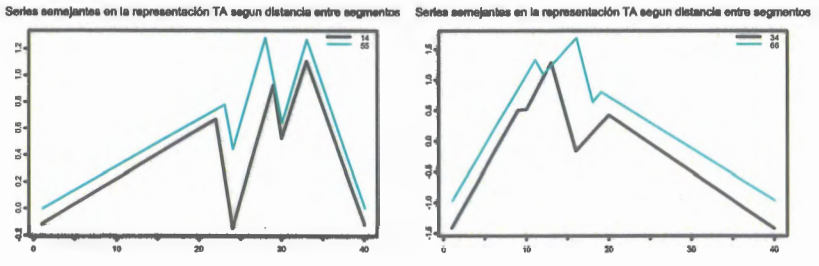


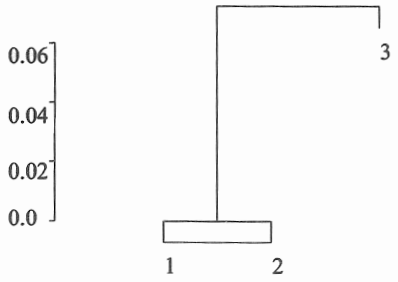
Figura 6. 10. Series semejantes según la distancia dada en la figura 6.4, sopesando los segmentos en base al criterio dado en la tabla 7.

Distancia con alineación temporal: ADT

Las distancias que permiten una distorsión en el eje temporal son aplicables también a datos financieros desde el momento en el que secuencias de stock con las mismas subidas o picos, y las mismas bajadas o valles pueden ser consideradas iguales con independencia de las diferentes escalas temporales. Este tipo de distancias alinearán los picos y las bajadas tanto como sea posible, expandiendo y contrayendo el eje temporal en la medida en la que sea necesario. Será, por lo tanto, una distancia buena a la hora de comparar secuencias de stocks. La distancia que contempla la distorsión en el eje temporal es la mencionada en el capítulo 4 y se vuelve a reproducir en (84) :

$$(87)\ ADT(A,B) = \min\left\{ \sqrt{\sum_{k=1}^{k} w_k} \Big/ k \right.$$

Según esta distancia, utilizando las tres series *s1,s2 y s3* dadas en la figura 6.1, obtendremos el siguiente dendrograma:



ADT(*s1,s2*)	0/39=0
ADT(*s1,s3*)	1/46=0.022
ADT(*s2,s3*)	2.83/39=0.073

Figura 6. 11. Clustering jerárquico de las series s*1*, s*2* y s*3* usando la distancia de alineamiento temporal.

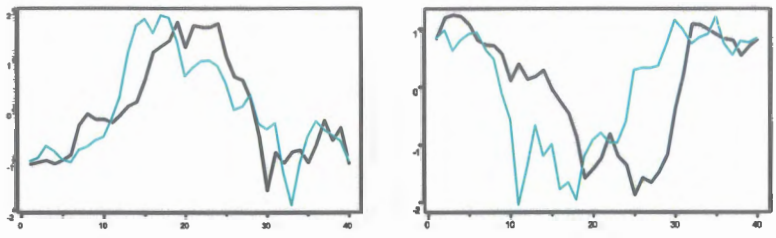


Figura 6. 12. Series de longitud 40, semejantes según ADT , extraídas de la matriz da datos obtenida del Índice general de la bolsa de Madrid, suavizadas con h=0.3.

Distancia con alineación temporal entre segmentos

Es posible calcular la distancia de alineamiento entre representaciones aproximadas de las series temporales. Puede calcularse la distancia de alineamiento entre series aproximadas mediante segmentos o bien aproximando la serie mediante los segmentos obtenidos usando los primeros coeficientes de la transformada wavelet. En (Keogh. 1999b) se optó por la primera de las alternativas.

Dados dos segmentos A y B determinados por dos puntos cada uno:

$$A =< (y_1, x_1), (y_2, x_2) >$$
$$B =< (y_1', x_1'), (y_2', x_2') >$$

Se utilizó la distancia entre dos segmentos dada en (87) para poder encontrar la distancia de alineamiento entre series representadas mediante segmentos. La distancia usada en el mencionado trabajo durante el camino de alineamiento, es una distancia entre segmentos y viene dada en (88).

$$(88)\ ADTS(A,B) = \left[(y_1' + y_2')/2 - (y_1 + y_2)/2\right]^2$$

Si se desean comparar los resultados obtenidos mediante la distancia ADT y la distancia ADTS, se ha de normalizar la distancia, dividiéndola por la longitud del camino de alineamiento. De esta manera, las distancias de los dos caminos, el camino de alineación original y el camino de alineación entre segmentos, serán comparables. Los resultados obtenidos para las tres series *s1*, *s2* y *s3* son los mostrados en la siguiente tabla:

ADTS(*s1*,*s2*)	0.119
ADTS(*s1*,*s3*)	0.008
ADTS(*s2*,*s3*)	0

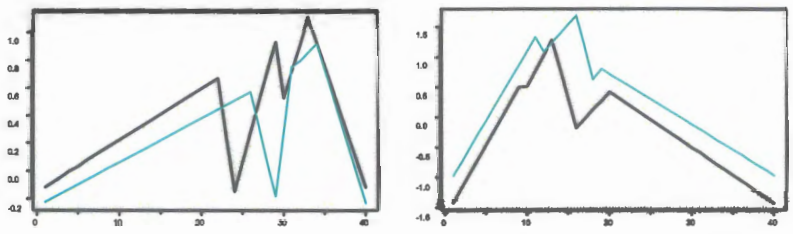


Figura 6. 13. Series más semejantes, encontradas en IGBM, según la distancia con alineación temporal entre segmentos SADT usada en la representación TA.

ADT : Banda de Sakoe-Chiba

La banda de Sakoe-Chiba es un mecanismo que se utiliza para permitir sólo una cierta distorsión en el eje temporal, la determinada por el parámetro *r*. Se define una banda alrededor de la diagonal de la matriz de alineamiento con ancho que vendrá determinado por *r* y no se permiten distorsiones entre dos series que

queden fuera de la banda .En la tabla 9 es posible observar el aspecto de la matriz de alineamiento para una banda de ancho r=4.

Tabla. 9. Banda de Sakoe-Chiba.

	1	2	3	4	5	6	7	8	9
1									
2									
3									
4									
5									
6									
7									
8									
9									

A continuación, podemos ver el efecto que el uso del parámetro r puede tener en el cálculo de las distancia. La tabla 10 muestra información de los 5 pares de subsecuencias de longitud 64 más parecidas encontradas en IGBM, según ADT calculada sin restricciones, con r=20, con r=10 y con r=4. Los números mostrados corresponden a los números de fila ocupadas por las series. El cálculo sin restricciones difiere mucho en los resultados del cálculo con r=20. La diferencia no es tan notoria para valores más restrictivos de r.

Tabla. 10. Número de fila de los cinco pares de subsecuencias más semejantes encontradas en IGBM, sin restricciones, con *r=20, r=10* y *r=4*.

Sin restricciones	*r=20*	*r=10*	*r=4*
{52, 57}	{4, 26}	{4, 26}	{4, 26}
{4, 79}	{4, 47}	{4, 47}	{4, 47}
{1, 44}	{41, 52}	{41, 52}	{41, 52}
{66, 79}	{23, 24}	{23, 24}	{23, 24}
{26, 66}	{41, 89}	{41, 89}	{15, 80}

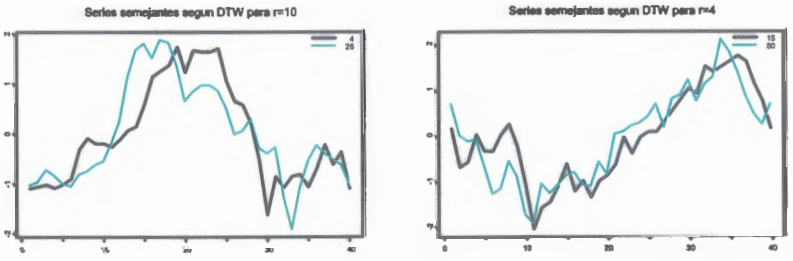


Figura 6. 14. Series semejantes según ADT con distintos anchos de banda(r=10 y r=4). Las series están extraídas de IGBM.

Donde sí es posible apreciar las diferencias existentes en los resultados ofrecidos por la distancia ADT en función del valor de r empleado, tomando como referencia la misma base de datos, es al consultar las series más distintas entre sí. Los resultados se muestran en la tabla 11.

Tabla. 11. Tabla con el número de fila ocupada por los pares de series mas distintas según ADT para r=20, r=10, r=4.

Sin restricciones	r=20	r=10	r=4
{7 , 34}	{5, 72}	{5, 51}	{69 , 81}
{18 , 34}	{47, 72}	{31, 57}	{8 , 41}
{11 , 34}	{34, 59}	{69, 81}	{8 , 51}
{51 , 79}	{47, 51}	{38, 51}	{41 , 44}
{37 , 60}	{72, 79}	{5, 48}	{5, 87}

ADT: Banda de Sakoe-Chiba distorsionada

Junto con el paralelogramo de Itakura, la banda de Sakoe-Chiba ha sido uno de los mecanismos utilizados para rebajar el tiempo de cálculo requerido de ADT. En (Ratanamahatana and Keogh. 2004) se ha utilizado esta banda distorsionada. Si

disponemos de las dos series dadas en la tabla 12, y si calculamos la matriz de distancia con alineación en el eje temporal (sin normalizar), tendremos la matriz de alineamiento dada en la tabla 13.

Tabla. 12. Dos series ejemplo con distorsión en el eje temporal.

(2,2,2,2,4,4,4,4,8)
(2,4,4,4,4,4,4,4,4)

Tabla. 13. Matriz de distancias de alineamiento para las dos series dadas en la tabla 12.

	1	2	3	4	5	6	7	8	9
1	0	4	8	12	16	20	24	28	32
2	0	4	8	12	16	20	24	28	32
3	0	4	8	12	16	20	24	28	32
4	0	4	8	12	16	20	24	28	32
5	4	0	0	0	0	0	0	0	0
6	8	0	0	0	0	0	0	0	0
7	12	0	0	0	0	0	0	0	0
8	16	0	0	0	0	0	0	0	0
9	52	16	16	16	16	16	16	16	16

Utilizando una banda de Sakoe-Chiba distorsionada donde los anchos de banda permitidos sean r=(3,0,3) para los intervalos i=(<1.4>,<5,6>,<7,9>) podemos obtener la matriz de alineamiento dada en la tabla 14.

Tabla. 14. Matriz de distancias de alineamiento para las dos series dadas en el tabla 12 con anchos de banda permitidos r=(0,3,0) para los intervalos i=(<1.4>,<5,6>,<7,9>)

	1	2	3	4	5	6	7	8	9
1	0	4	8	12					
2	0	4	8	12					
3	0	4	8	12					
4	0	4	8	12					
5					12				
6						12			
7							12	12	12
8							12	12	12
9							28	28	28

Dadas las tres series mostradas en la figura 6.13, en la tabla 15 se puede ver la repercusión que puede tener la selección de valor para el ancho de banda en el cálculo de la distancia entre series.

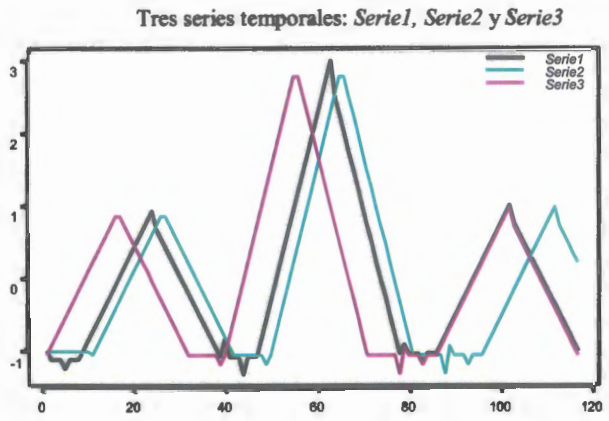


Figura 6. 15. Tres series temporales, *Serie1, Serie2* y *Serie3*.

Tabla. 15. Distancias obtenidas para las series Serie1, Serie2 y Serie3 usando la banda de Sakoe-Chiba para ADT, sin restricciones, con r=(20,10,20) en los intervalos i=(60,80,117) y con r=(20,5,20) en los mismos intervalos .

	Sin restricciones	r=(20,10,20) i=(60,80,117)	r=(20,5,20) i=(60,80,117)
ADT(Serie1,Serie2)	0.017	0.019	0.019
ADT(Serie1,Serie3)	0.004	0.019	0.023
ADT(Serie2,Serie3)	0.020	0.031	0.042

Distancia con alineación dinámica temporal: cotas inferiores

Puesto que ADT es una distancia costosa de calcular se han definido cotas inferiores de esta distancia. La cota inferior puede ser usada para mejorar los

rastreos en las búsquedas por contenidos. Se calcula la cota inferior de la distancia y, si el valor obtenido es mayor que la mejor distancia calculada hasta el momento, podemos abandonar definitivamente dicha serie y pasar a la siguiente. Si en el rastreo, la cota inferior para una cierta distancia resulta ser mejor que la mejor distancia calculada hasta el momento, pasaremos a calcular la distancia de alineamiento real entre las dos series.Una de las cotas inferiores de la distancia ADT mas utilizadas es LB_keogh, tal como mencionaba en el capítulo 4. Dicha cota inferior se basa en la banda de Sakoe-Chiba. Es necesario definir el valor del parámetro *r*, encargado de determinar cuanto se desviará el camino de la diagonal. Se evitarán todos aquellos caminos donde la distancia a la diagonal supere el valor de *r*. Para cada serie obtendremos dos series que la envuelven, una por arriba y otra por abajo:

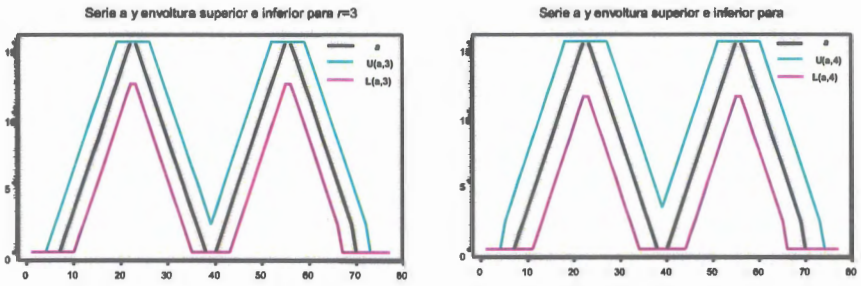


Figura 6. 16. Una serie *a* y sus envolturas, superior e inferior, para dos valores de *r* distintos.

Se muestran, a continuación, dos series similares según LB_keogh, las envolturas superior e inferior para la primera de ellas y la segunda serie graficada entre las envolturas de la primera.

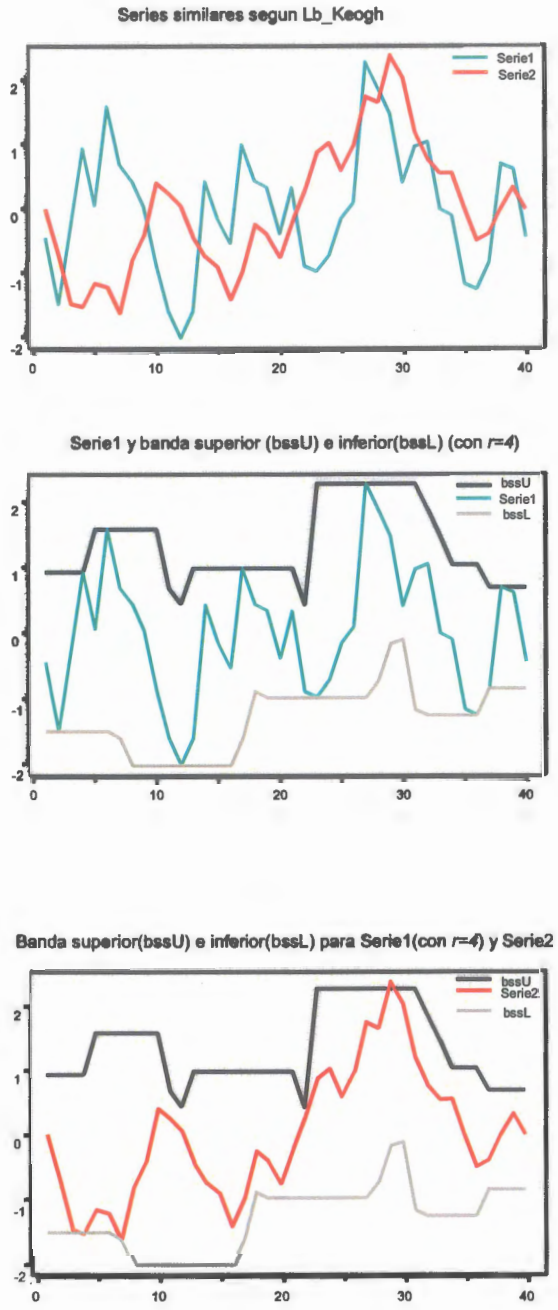


Figura 6. 17. Series semejantes según LB_Keogh con r=30 y las envolturas superiores e inferiores para las dos series. .

Comparando clusters

Un clustering C es una partición de un conjunto de datos D en conjuntos $C_1, C_2, ..., C_k$, llamados clusters, de manera que (89) sea cierto.

$$(89)\ C_k \bigcap C_i = \phi \quad y \quad \bigcup_{k=1}^{K} C_k = D$$

Si el número de puntos en D es n y n_k en C_k se cumplirá (90).

$$(90)\ n = \sum_{k=1}^{K} n_k$$

Asumiendo que $n_k > 0$ podemos contemplar un segundo clustering sobre los mismos datos D, al que denominaremos $C' = \{C'_1, C'_2, ..., C'_k\}$ con tamaños de clusters n'_k y con número de clusters que pueden ser distintos en C y C'.

Para poder comparar los dos clusters podemos utilizar la matriz de confusión o tabla de contingencias de los pares C, C'.

La tabla de contingencias M, es una matriz de $k \times k'$ donde el elemento (k,k) de la matriz es el número de puntos en la intersección de los clusters C_k de C y C'_k de C'. Podemos expresarlo según (91).

$$(91)\ m_{kk'} = \left| C_k \bigcap C'_{k'} \right|$$

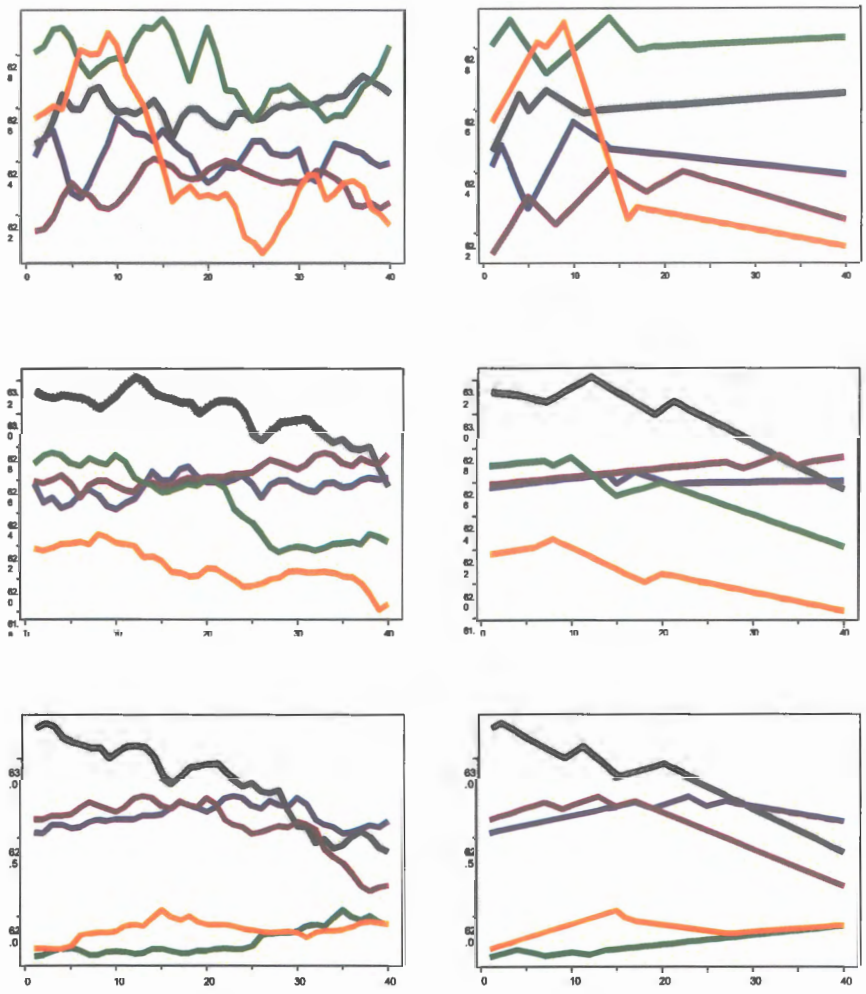


Figura 6. 18. Cinco elementos de 3 clusters distintos obtenidos usando DS sobre la representación TA. Se muestran series originales y sus correspondientes representaciones TA.

Una de las posibilidades al comparar clusters es la de contar los pares de puntos para los cuales dos clusters coinciden o no coinciden. Cada par de puntos de *D,* puede estar en cuatro situaciones tal como se refleja en la siguiente tabla.

Tabla. 16. Dada una matriz de series temporales *D* y dos clusterings *C* y *C'* sobre *D. C*ada par de series o puntos pueden encontrarse en una de las cuatro situaciones mostradas.

$\mathbf{N_{11}}$= número de pares de puntos que están en el mismo cluster en *C* y en *C'*.
$\mathbf{N_{00}}$= número de pares de puntos que están en distintos clusters en *C* y en *C'*.
$\mathbf{N_{10}}$= número de pares de puntos que están en el mismo cluster en *C*, pero no en *C'*.
$\mathbf{N_{10}}$= número de pares de puntos que están en el mismo cluster en *C'*, pero no en *C*.

Los anteriores cuatro valores siempre cumplen la igualdad mostrada en (92):

$$(92)\ N_{11} + N_{00} + N_{10} + N_{01} = n(n-1)/2$$

$$(93)\ W_I(C,C') = \frac{N_{11}}{\sum_K n_k(n_k-1)/2}$$

$$(94)\ W_{II}(C,C') = \frac{N_{11}}{\sum_{K'} n'_{k'}(n'_{k'}-1)/2}$$

Utilizando los dos criterios asimétricos formulados en (93) y (94), Fowkles y Mallows (Fowkles, E.B. 1983;) introdujeron un criterio simétrico mostrado en (95) que será la media geométrica de W_I y W_{II} .

$$(95)\ F(C,C') = \sqrt{W_I(C,C')W_{II}(C,C')}$$

Podemos también usar el índice de Jacard (Ben-hur, Asa 2002;) mostrado en (96):

$$(96)\ J(C,C') = \frac{N_{11}}{N_{11} + N_{01} + N_{10}}$$

A lo largo de la tesis se han considerado distancias con distorsión en el eje temporal y distancias sin distorsión en el eje temporal. La distancia DS podrían compararse tanto a una distancia que no contempla distorsión en el eje temporal, tal es el caso de la Euclídea, como a una distancia tal como la cota inferior LB_Keogh. A continuación podemos ver expresadas gráficamente las propiedades de la distancia utilizada para medir la similitud entre segmentos: DS.

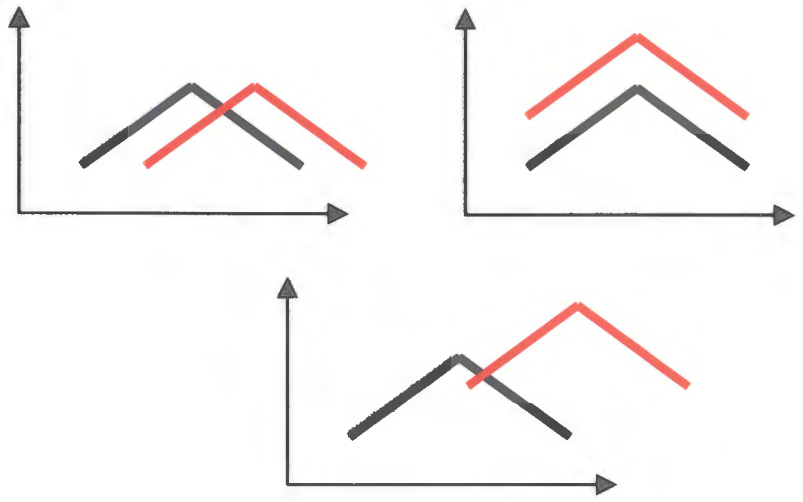


Figura 6. 19. Propiedades de la distancia DS representadas gráficamente. Las figuras mostradas en negro y en rojo siguen siendo similares según DS.

Estas propiedades hacen que dicha distancia sea más versátil que la distancia Euclídea a la hora de permitir ciertos movimientos sobre el eje de las x y el eje de las y, preservando las similitudes.

A pesar de que dichas propiedades resulten interesantes a la hora de buscar semejanzas entre series temporales que guardan patrones de análisis técnico, no llegan a ser tan completas como las propiedades de la distancia ADT. Las series mostradas en la gráfica 6.20 son las mismas series dadas en la figura 4.1 y son similares según ADT pero no según DS.

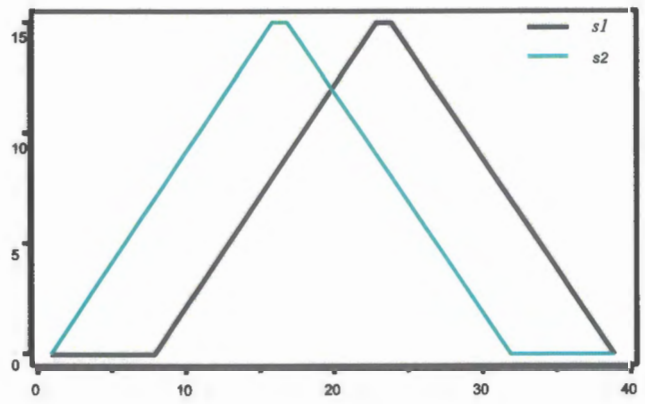


Figura 6. 20. Dos series similares según ADT pero no según DS.

Con el objetivo de observar las semejanzas entre los clusters obtenidos utilizando estas distancias, se ha utilizado el comando hclust de S-plus que genera clustering jerárquico. Se comparan los clusters obtenidos utilizando la distancia Euclídea con los clusters obtenidos utilizando la distancia DS sobre la

representación TA. Al utilizar la distancia DS, sólo se usan los segmentos que constituyen el patrón, no se han tenido en cuenta el segmento inicial y el final.

Los dendrogramas obtenidos se han cortado a distintas alturas, coincidiendo cada altura con distinto número de clusters. Se han considerado los siguientes valores para el número de clusters, k=3,5,7,9,11. Las pruebas se han realizado sobre las 36 bases de datos mencionadas en el anterior capítulo.

Se muestran, en una misma gráfica, similitudes entre clusters obtenidos mediante la distancia DS para representaciones TA con distintos ratios de compresión. En el caso de las series de longitud 40, el ratio de compresión es $Ratio_{compresión} = \frac{12}{40} = 0{,}3$. Esta será la longitud con menor ratio de compresión, en el caso de las series de longitud 360, el $Ratio_{compresión} = \frac{12}{360} = 0{,}0\hat{3}$.

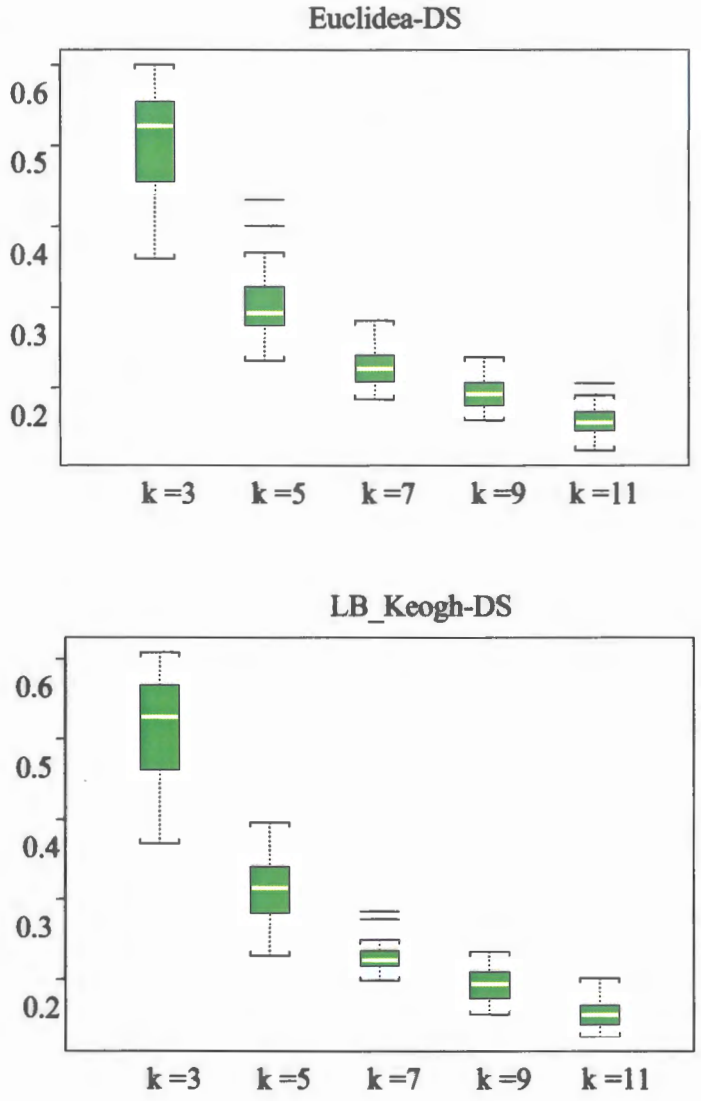


Figura 6. 21. Similitudes entre los clusters obtenidos con la distancia Euclídea y los clusters obtenidos con DS teniendo en cuenta el criterio de Fowkles y Mallows. Los resultados se muestran para distinto número de clusters *k*=(3,5,7,9,11).

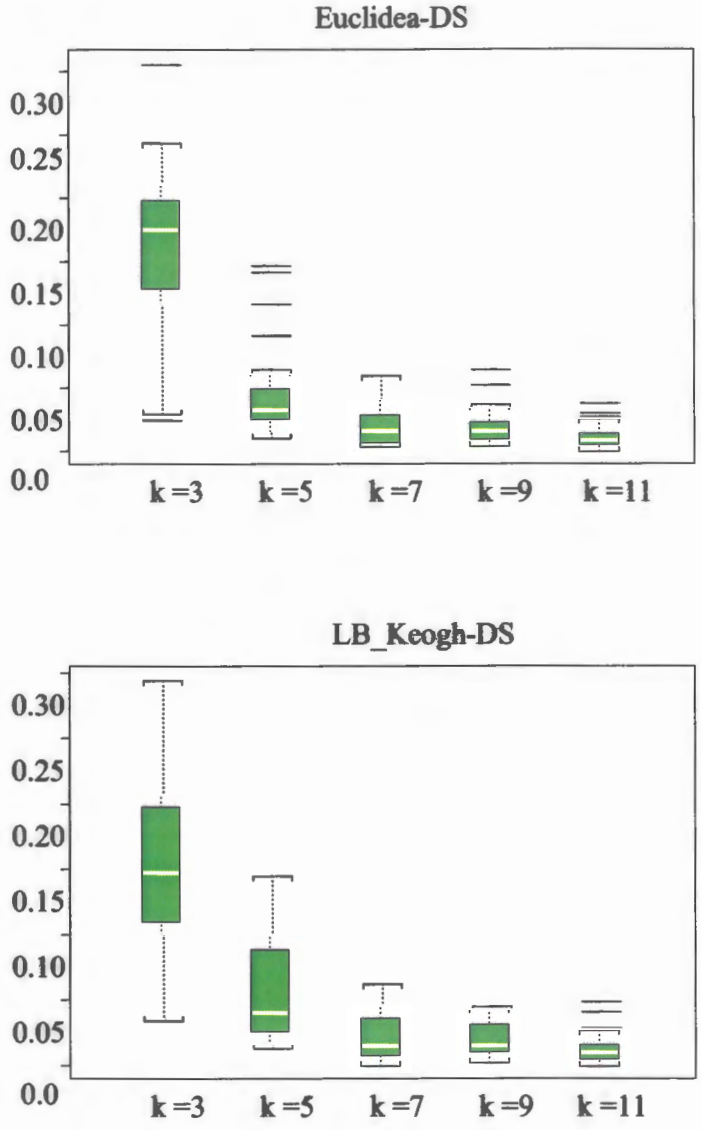


Figura 6. 22. Similitudes entre los clusters obtenidos con la distancia Euclídea y los clusters obtenidos con DS teniendo en cuenta el criterio de Jacard, mostrando los resultados para distinto número de clusters *k=(3,5,7,9,11)*.

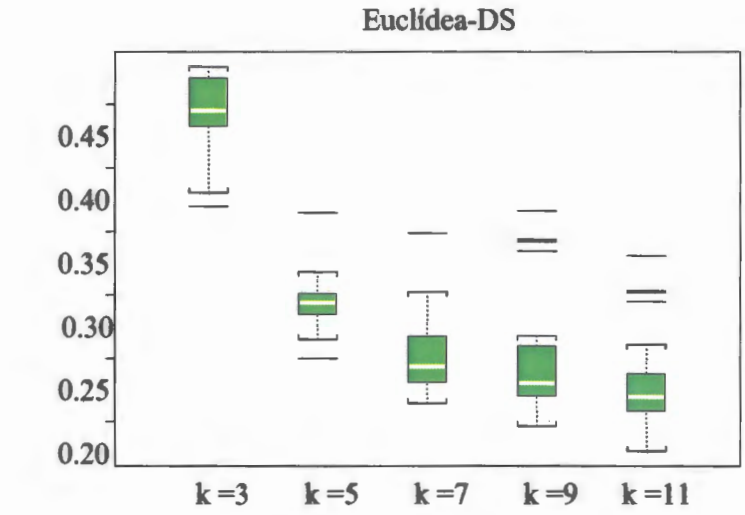
Además de comparar los clusters por recuento de pares, podemos comparar clusters atendiendo al criterio de coincidencia de conjuntos. Estos criterios miran sólo a la cardinalidad de los conjuntos, sin hacer ninguna asunción acerca de cómo han sido generados los clusters.

Dados dos clusters $C = C_1,...,C_k \quad y \quad C' = C'_1,..,C'_k$, la similitud entre ambos se obtiene calculando inicialmente una similitud entre cada par de clusters C_i, C_j según (94) para pasar, a continuación, al cálculo de la similitud entre $C \, y \, C'$ según (95).

$$(97)\ Sim(C_i, C_j') = 2\frac{\left|C_i \cap C_j'\right|}{\left|C_i\right| + \left|C_j'\right|}$$

$$(98)\ Sim(C, C') = \frac{\left(\sum_i \max_j Sim(C_i, C_j')\right)}{k}$$

La medida de la similitud entre clusters planteada en (95) es un criterio asimétrico. Los datos mostrados son los obtenidos para las mismas bases de datos empleadas para los anteriores criterios y las similitudes mostradas son ,respectivamente, la similitud entre los clusters obtenidos para la distancia Euclídea con respecto a los clusters obtenidos usando la distancia DS en los segmentos que constituyen el patrón, y los clusters obtenidos usando la distancia de la cota inferior de la distancia con alineamiento temporal denominada LB_Keogh con respecto a los clusters obtenidos con la distancia DS.



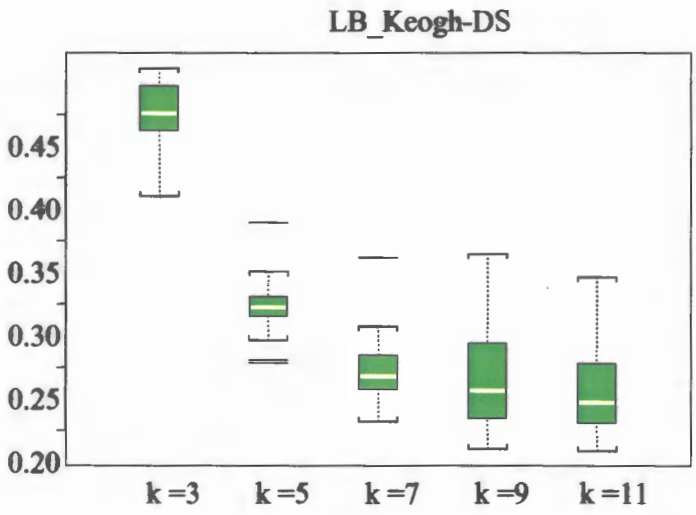


Figura 6. 23. Similitudes entre los clusters obtenidos con la distancia Euclídea y los clusters obtenidos con DS teniendo en cuenta el criterio de la coincidencia entre conjuntos y observando los resultados para distinto número de clusters *k=(3,5,7,9,11)*.

La aportación realizada por la representación y posterior uso de la distancia DS es la de introducir un nuevo criterio a la hora de buscar semejanzas, el criterio de la semejanza entre patrones de análisis técnico calculado según la distancia entre los segmentos constituyentes de estos patrones. Los resultados correspondientes a las comparativas entre clusters han sido bastante parecidos, tanto en el caso de la distancia que permite el alineamiento temporal, como en el caso de la distancia Euclídea. Sólo en algunos casos, la comparativa de los clusters generados mediante DS con los clusters obtenidos mediante ADT, ha devuelto resultados mejores que los obtenidos usando distancia Euclídea. A ello contribuye una propiedad de la distancia que hace que dos segmentos paralelos pero desplazados en el eje de las x resulten semejantes. La representación TA permite la definición de otras distancias, tal es el caso del trabajo mostrado en (Basagoiti and Juaristi. 2006). La misma representación TA se usa, en este trabajo, con una distancia definida mediante una banda que rodea cada segmento. Esta banda está definida utilizando el mejor harmónico para cada segmento.

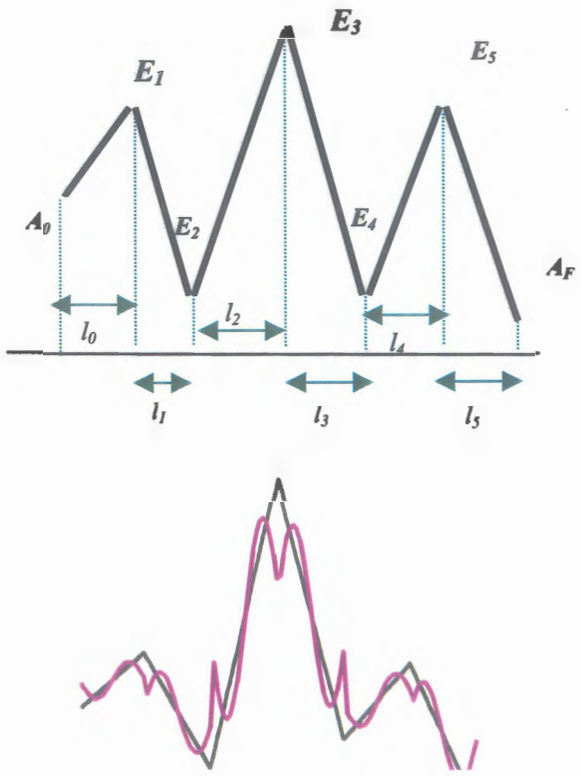


Figura 6. 24. La representación TA y el mejor harmónico trazado sobre cada segmento de la representación.

La distancia entre los segmentos A y C calculada en función de la banda trazada alrededor de cada segmento, según se muestra en la figura 6.25 se calcula según (99). En dicha expresión *AW* y *CW* corresponden a los pesos asignados a los segmentos *A* y *C* respectivamente. Amp1 y Amp2 corresponden a la amplitud de la banda trazada alrededor de A y C respectivamente.

$$DB(A,C) = AW * CW * |(Y_1' - Y_1) - (Y_2' - Y_2)| + AW * CW * |Amp1 - Amp2| \quad (99)$$

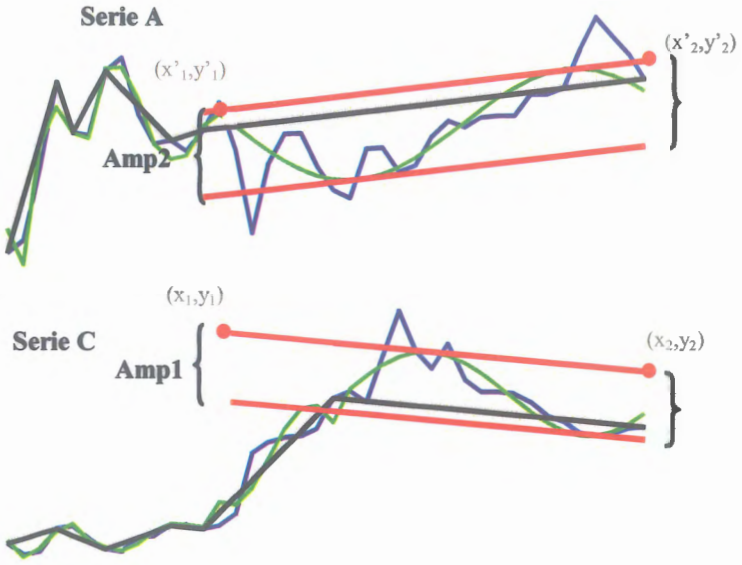


Figura 6. 25. El mejor armónico trazado sobre cada segmento de la representación TA.

Las propiedades de esta distancia son básicamente las mismas que las de la distancia DS pero toma en consideración el hecho de que las series sobre las cuales se hayan trazado los segmentos puedan fluctuar de distinta manera alrededor de dichos segmentos. En el caso de que las amplitudes fuesen distintas DB considera a las series distintas entre sí. Es por ello, una distancia más restrictiva que DS pues ésta no considera las fluctuaciones de la serie original alrededor del segmento trazado.

Por ser más restrictiva que DS, tampoco considera similares a series que presenten entre sí escalado longitudinal. Los resultados de la comparación de los clusters obtenidos según esta distancia con respecto a los clusters obtenidos según distancia Euclídea y LB_Keogh se muestran a continuación en la figura 6.27 .

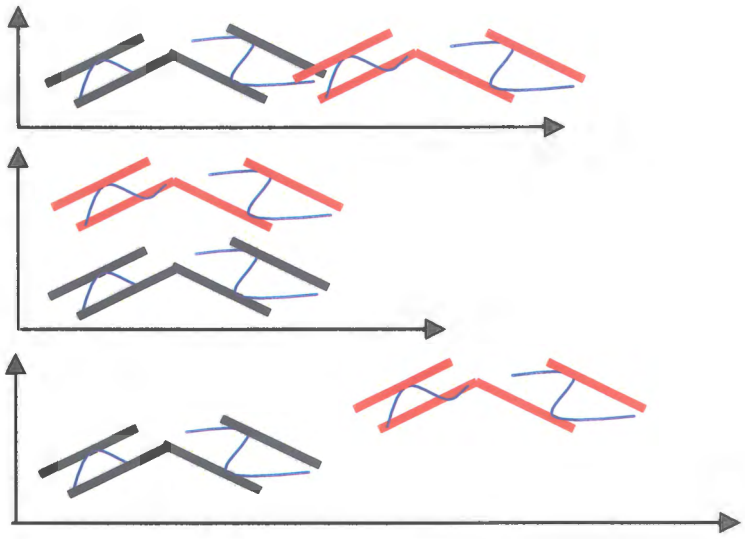


Figura 6. 26. Propiedades de la distancia DB representadas gráficamente. Las figuras mostradas en negro y en rojo siguen siendo similares según DB.

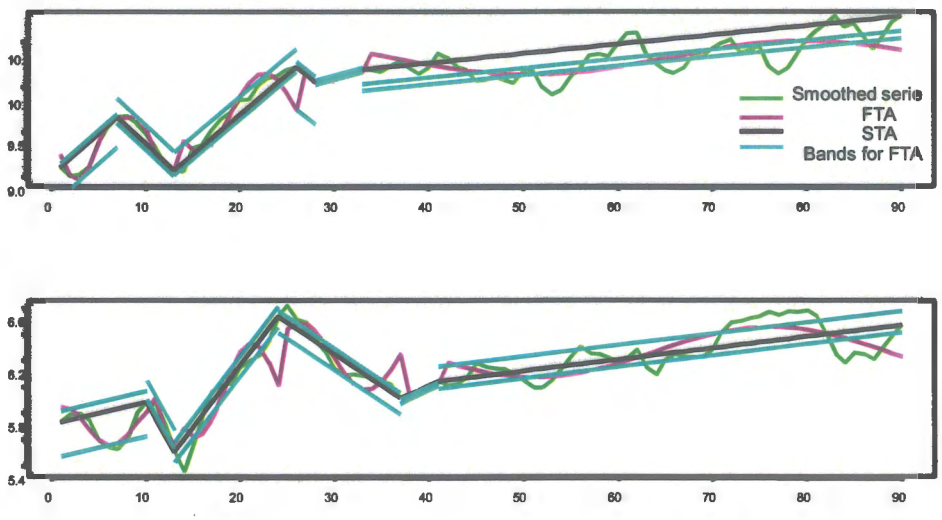


Figura 6. 27. Series similares según distancia DB.

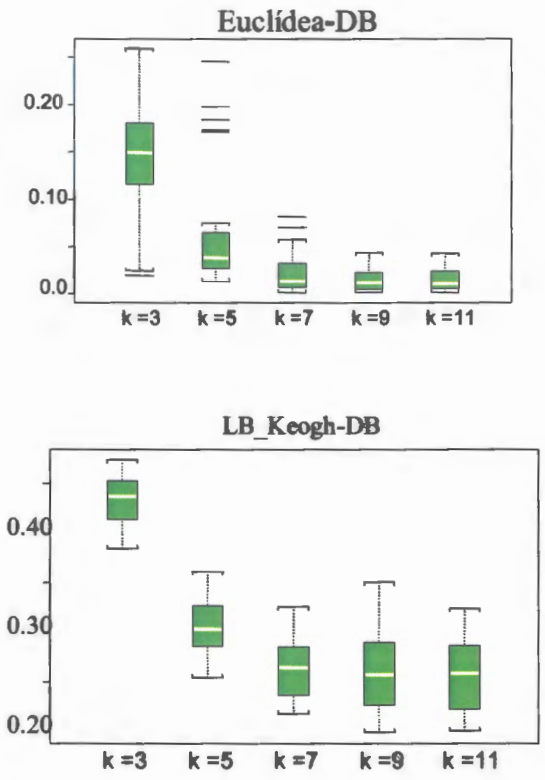


Figura 6. 28. Similitudes entre los clusters obtenidos usando DB y los clusters obtenidos mediante distancia Euclídea y LB_Keogh. La similitud entre clusters se ha definido según (98).

A continuación se muestran las similitudes entre los mismos clusters según (95) y (96) respectivamente.

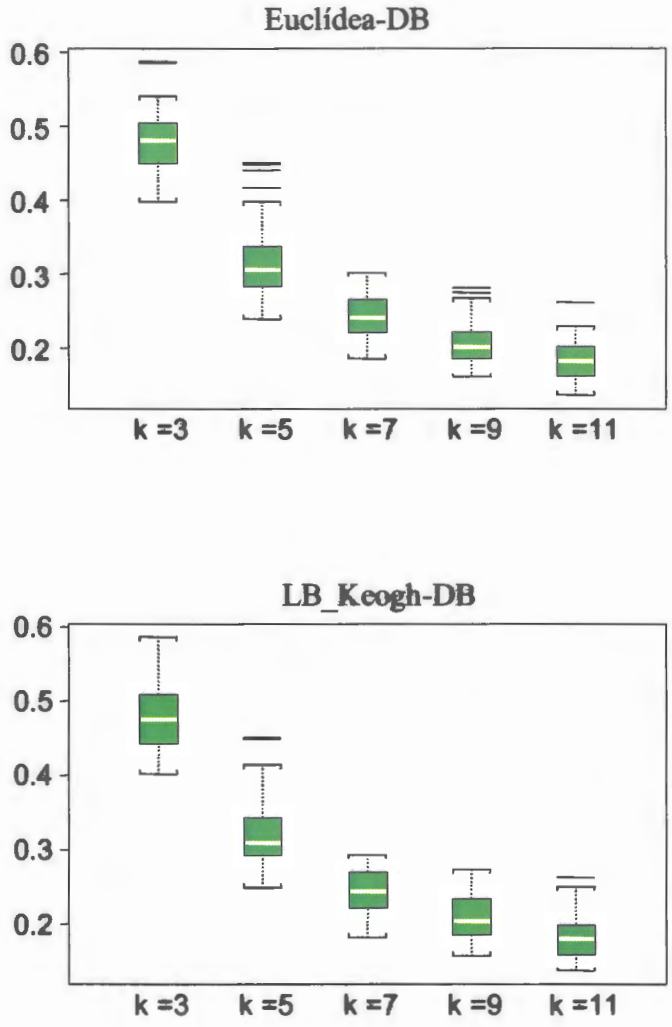


Figura 6. 29. Similitudes entre los clusters obtenidos usando DB y los clusters obtenidos mediante distancia Euclídea y LB_Keogh. La similitud entre clusters se ha definido según (95).

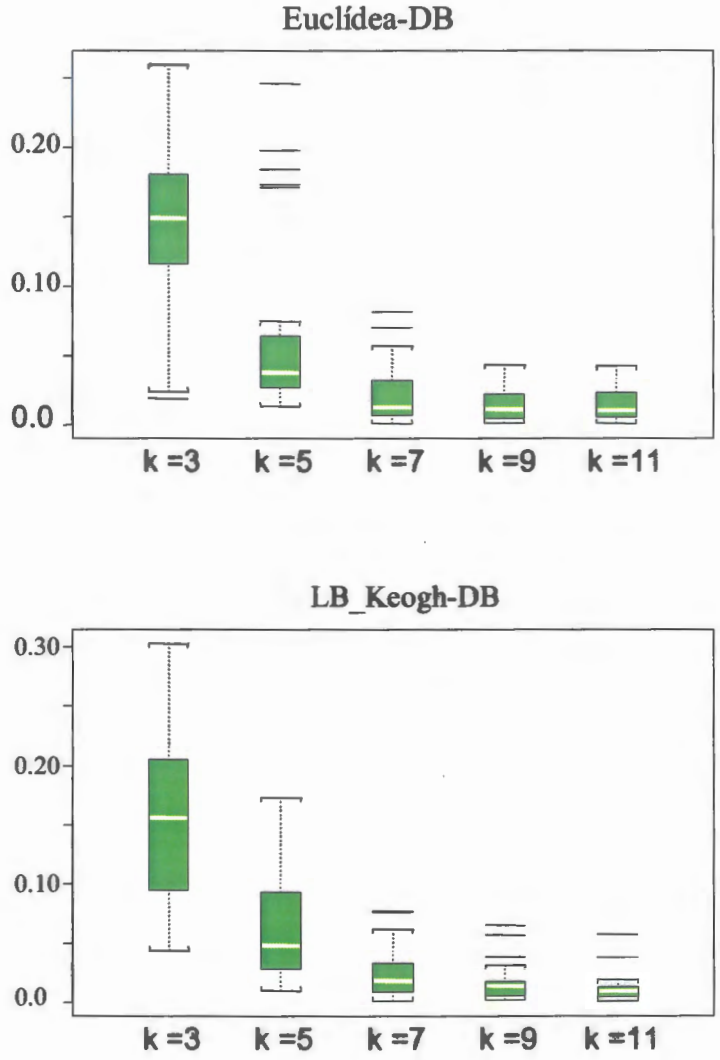


Figura 6. 30. Similitudes entre los clusters obtenidos usando DB y los clusters obtenidos mediante distancia Euclídea y LB_Keogh. La similitud entre clusters se ha definido según (96).

Por lo que se puede comprobar en las graficas correspondientes a las comparativas entre clusters obtenidos mediante DB con respecto a los clusters obtenidos con distancia Euclídea y LB_Keogh, los resultado no mejoran los obtenidos para la distancia DS.

Conclusiones parciales

Antes de exponer las conclusiones parciales correspondientes a la comparativa entre las distintas medidas de la distancia utilizadas, podemos recordar las ventajas y desventajas de cada una de éstas.

La distancia Euclídea es una distancia simple y rápida de calcular, exige un tiempo de cálculo lineal en función de la longitud de la series temporal. Además, puede ser utilizada en la generación de un índice junto con muchas de las técnicas de reducción de la dimensionalidad mencionadas en el capítulo 3. Es, de todas maneras, una distancia poco flexible y muy sensible a outliers, al ruido y a los desplazamientos a lo largo del eje de las x.

La distancia de alineamiento temporal, ADT, supone una mejora con respecto a la anterior puesto que permite desplazamiento en el eje de las x, aunque tiene un tiempo de cálculo prohibitivo para ser usado en grandes bases de datos. Es posible sortear la complejidad inherente al cálculo de la distancia usando cotas inferiores. Sigue siendo una distancia sensible a outliers.

La distancia DS es una distancia que requiere un tiempo lineal en función del número de segmentos utilizados para la representación de la serie temporal. No es

sensible a outliers siempre que la representación mediante segmentos lineales no esté sujeta a ellos, y permite desplazamientos en el eje de las x y en el eje de las y. Como cualquier otra representación basada en segmentos lineales permite el sopesado de los segmentos para el cálculo de la distancia.

La distancia DB es una distancia que comparte todos los atributos de la distancia DS, pero, además, pretende ser una distancia más específica al tomar en consideración como parte integrante de dicha distancia una banda, definida en función de las amplitudes de los harmónicos.

Se han utilizado a lo largo de este capítulo distintas distancias para obtener sucesivos clusterings y se han comparado éstos entre sí para poder tener un criterio objetivo de la repercusión del uso de la representación TA y sucesivas distancias (DS, DB) definidas sobre ella.

Según los resultados obtenidos, la distancia DS usada sobre la representación TA ofrece unos resultados bastante interesantes cuando los dendrogramas se cortan para un número de clusters pequeño, $k=3$, $k=5$ pero se degrada rápidamente cuando se aumenta dicho número a valores tales como $k=7$, $k=9$, $k=11$.

La longitud de las series tiene también cierta reprecusión en los resultados. De las 36 bases de datos exploradas, disponemos de 9 para cada longitud ($l=40,90,180,360$). Se muestran a continuación las similitudes entre los clusters obtenidos mediante DS y los obtenidos mediante la distancia Euclídea separando los resultados por longitudes. Atendiendo al criterio de coincidencia de conjuntos, los resultados para longitudes $l=40$ son ligeramente mejores que el resto.

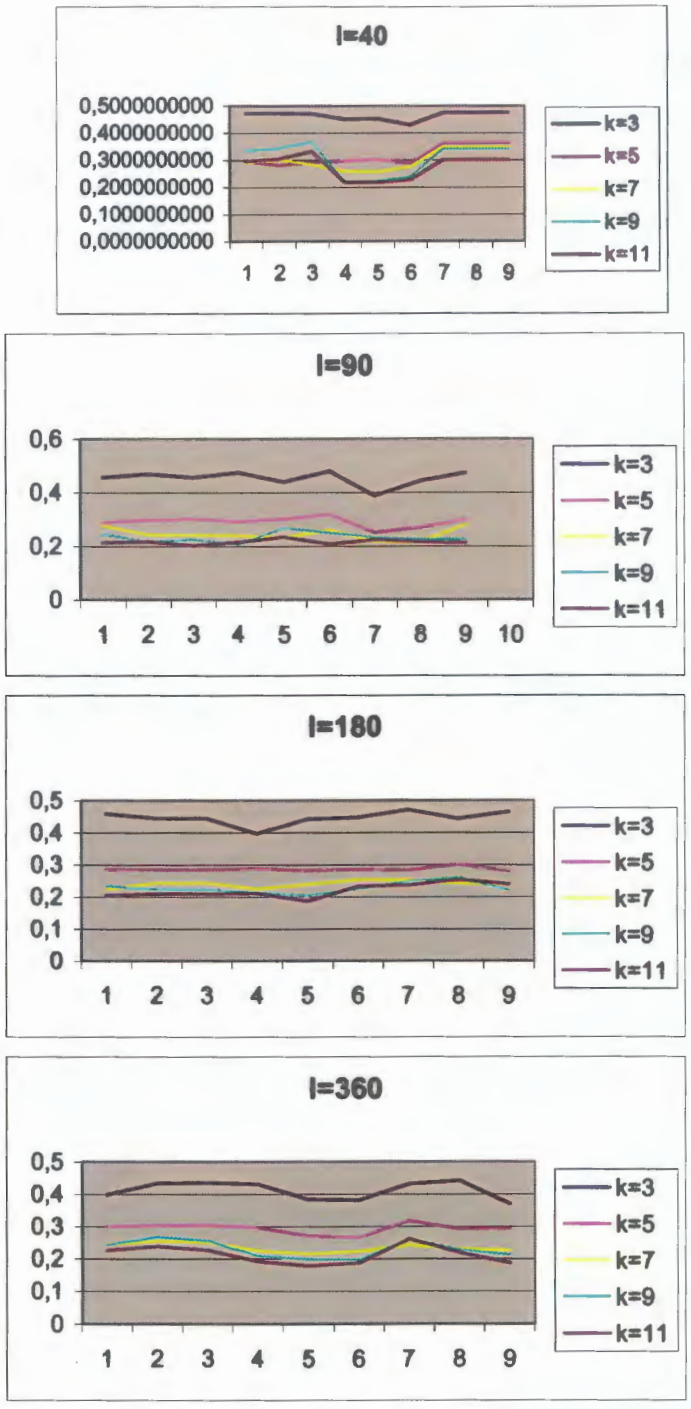


Figura 6. 31. Similitudes entre los clusters obtenidos usando la distancia Euclídea y la distancia DS, separando los resultados para distintas longitudes(*l=40,90,180,360*) y atendiendo al criterio de coincidencia de conjuntos.

CAPÍTULO 7

LA PARALELIZACIÓN

Fundamentos

La computación en paralelo es un conjunto de procesos que son capaces de trabajar cooperativamente para solucionar un problema computacional. La computación en paralelo es un proceso interesante porque ofrece el potencial de concentrar recursos

MPI: Message Passing Interface

MPI (*Message Passing Interface*) es un interfaz estandarizado para la realización de aplicaciones paralelas basadas en paso de mensajes. El modelo de programación que subyace tras MPI es MIMD (*Multiple Instruction streams,*

Multiple Data streams) aunque se dan especiales facilidades para la utilización del modelo SPMD (*Single Program Multiple Data*), un caso particular de MIMD en el que todos los procesos ejecutan el mismo programa, aunque no necesariamente la misma instrucción al mismo tiempo.

MPI suministra al programador una colección de funciones para que este diseñe su aplicación, sin que tenga necesariamente que conocer el hardware concreto sobre el que se va a ejecutar, ni la forma en la que se han implementado las funciones que emplea

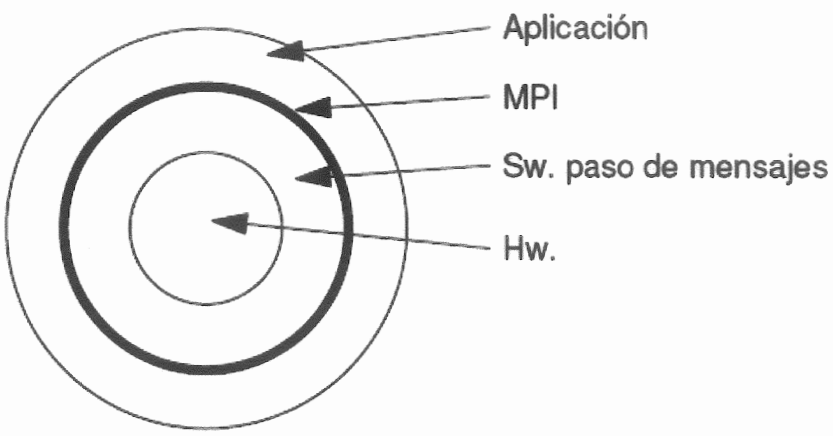


Figura 7. 1. MPI en el proceso de programación de aplicaciones paralelas.

MPI consta de, al menos, estos elementos:

- Una biblioteca de funciones para C, más el fichero de cabecera **mpi.h** con las definiciones de esas funciones y de una colección de constantes y macros.
- Una biblioteca de funciones para FORTRAN + **mpif.h**.

- Comandos para compilación, típicamente **mpicc**, **mpif77**, que son versiones de los comandos de compilación habituales (cc, f77) que incorporan automáticamente las bibliotecas MPI.
- Comandos para la ejecución de aplicaciones paralelas, típicamente **mpirun**.
- Herramientas para monitorización y depuración.

Cálculo de la distancia con alineación dinámica temporal usando procesamiento paralelo.

La distancia con alineación dinámica temporal es costosa de calcular, la paralelización es una alternativa posible para el cálculo de dicha distancia. Recientes trabajos, (Keogh. 2006), han realizado estudios más profundos sobre las características de la distancia de alineamiento.

En el particionamiento se ha establecido un paralelismo sobre los datos, de manera que se carga previamente la matriz de series temporales en memoria y se procede a su división. Dada una matriz de datos *MatG* con *NumFil* filas y con una serie en cada fila, el nodo maestro procede a la generación y posterior reparto de los pares de series temporales para los cuales se ha de calcular la distancia de alineamiento.

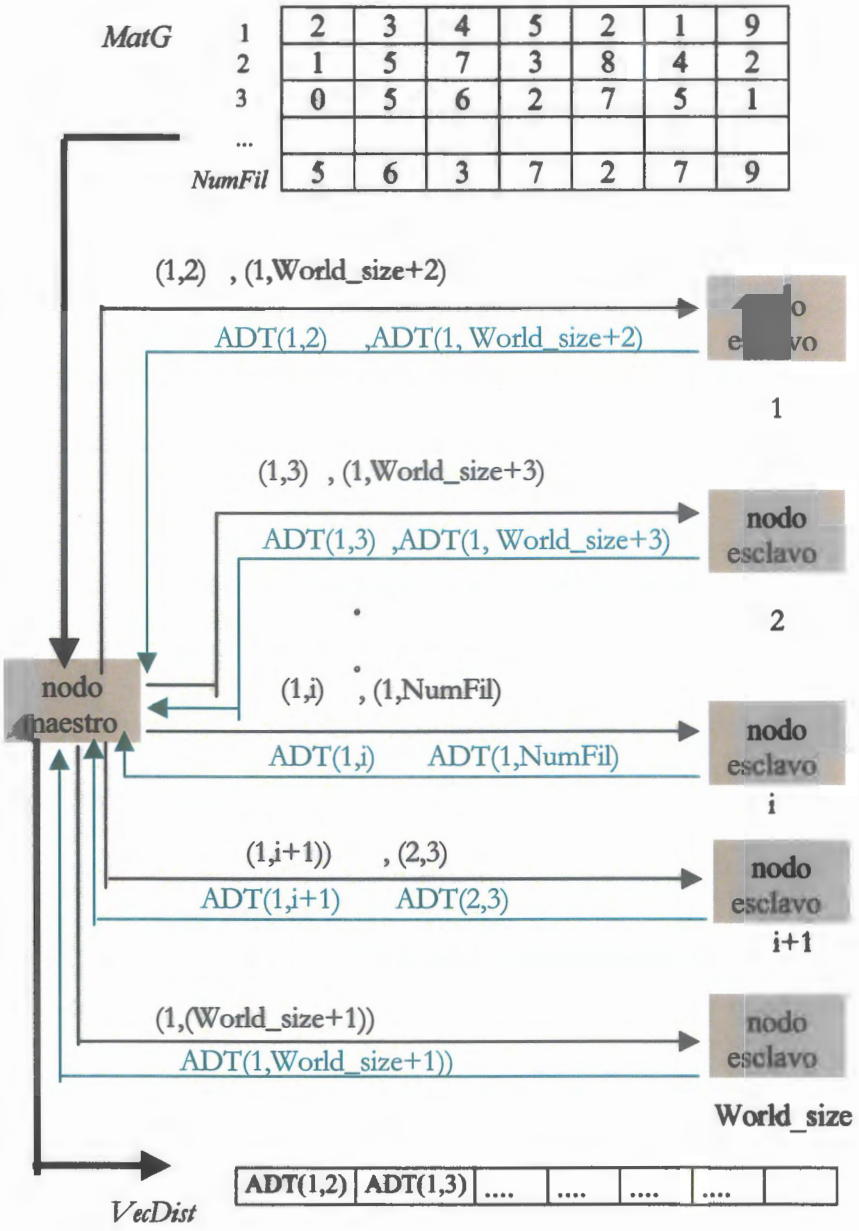


Figura 7. 2. Reparto de la matriz de datos *MatG* con una serie por fila. *VecDist* será el vector de distancias calculadas.

Si identificamos cada serie por el número de fila en el que se sitúa, los pares generados serán : (1,2), (1,3),(1,4), ..., (1,*NumFil*), (2,3), (2,4),.....(2,*NumFil*), ...,((*NumFil*-1), *NumFil*). En el reparto se ha procedido de la siguiente manera, a cada nodo se le da un par de series temporales, y así para todos los nodos del cluster, excepto para el nodo maestro que sólo se encarga de repartir y esperar la respuesta para colocarla en su sitio. Una vez que todos los nodos han realizado y devuelto los cálculos correspondientes se procede a realizar el siguiente reparto. El proceso continúa hasta que todas las distancias hayan sido calculadas.

El nodo maestro coloca cada resultado, una vez recibido, en la posición correspondiente del vector de distancias *VecDist*. Podemos verlo gráficamente en la figura 7.2.

MatG será la matriz de datos con una serie en cada fila y con *NumFil* filas. *VecDist* será el vector de distancias generado donde para cada número de fila *i* menor que *j*, la distancia entre la fila *i* y la fila *j* será el elemento *NumFil*(i-1) - i*(i-1)/2 + j-i –1* del vector de distancias.

Comparativa de clusters obtenidos mediante ADT y distancia Euclídea

Calculados los clusters mediante la distancia de alineamiento temporal, nos podríamos cuestionar hasta que punto son distintos de los clusters obtenidos mediante la distancia Euclídea. Con dicha comparativa podríamos evaluar la presencia del escalado longitudinal en los datos. Para comparar los clusters se utiliza la medida de la similitud dada en (94) realizando las mediciones para

distintos valores de k=(11,9,7,5,3). Los cálculos, como en las anteriores ocasiones, se realizan sobre las 36 bases de datos obtenidas sobre Ebay.

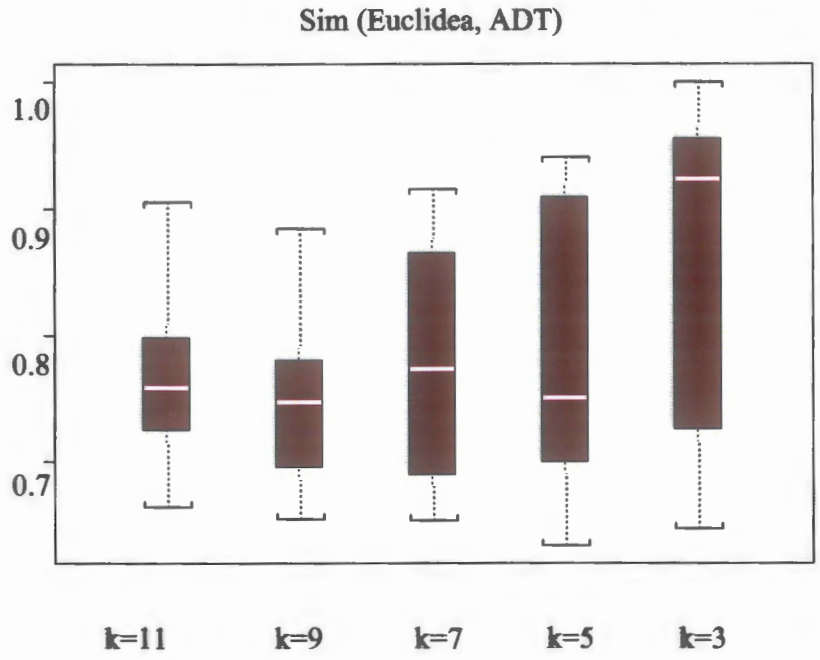


Figura 7. 3. Semejanza entre clusters generados por la distancia Euclídea y ADT. La medida de similitud entre clusters empleada es la dada en (95).

Los clusters presentan importantes semejanzas, lo cual quiere decir que la distorsión en el eje temporal no está, en estos datos, demasiado presente.

Podría también resultar interesante saber hasta que punto se mantienen las semejanzas entre los clusters utilizado la distancia real de alineamiento en lugar de utilizar su cota inferior. Los resultados se muestran en la siguiente gráfica.

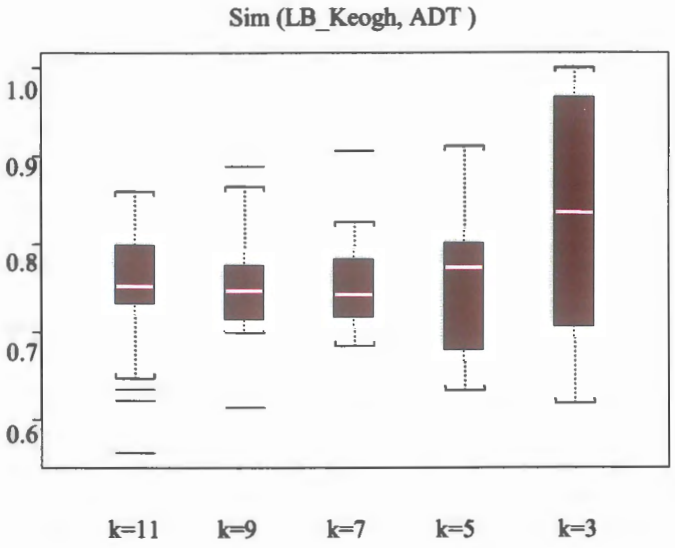


Figura 7. 4. Semejanza entre clusters generados por la distancia LB_Keogh y ADT.

Las semejanzas vuelven a ser importantes. La cota inferior LB_Keogh es una buena aproximación de ADT, siendo una distancia con tiempo de cálculo lineal.

En las dos gráficas que se muestran a continuación las pruebas vuelven a recuperar la distancia DS y retoman el uso que de dicha distancia se ha hecho en pruebas anteriores. Si comparamos clusters DS directamente con los clusters ADT los resultados son los mostrados en la figura 7.5. La distancia DS se ha utilizado sopesada y sólo sobre los segmentos que constituyen el patrón.

Para ver si la inclusión de éstos segmentos en el cálculo de DS mejora las similitudes entre los clusters se han realizado las pruebas pertinentes y se han obtenido los datos mostrados en la gráfica 7.7.

Cuando éstos algoritmos utilizan distancias sopesadas, como es el caso, es preciso plantearse si disponemos de un mecanismo correcto para ello, e incluso, si el sopesado de las series temporales es útil.

Se han realizado los cálculos pertinentes sin sopesado y las semejanzas entre clusters no arrojan grandes diferencias, el sopesado automático según el método utilizado, no mejora la comparativa con los clusters ADT. Las diferencias entre los resultados de la figura 7.6 y 7.7 son casi imperceptibles.

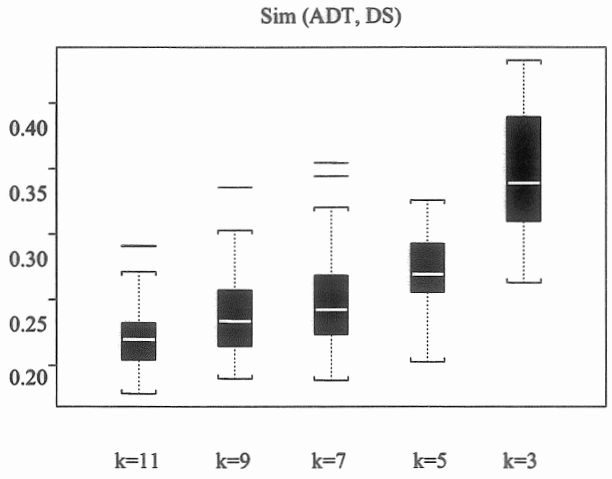


Figura 7. 5. Semejanza entre clusters generados por la distancia DS y ADT. La medida de similitud empleada es la dada en (95).

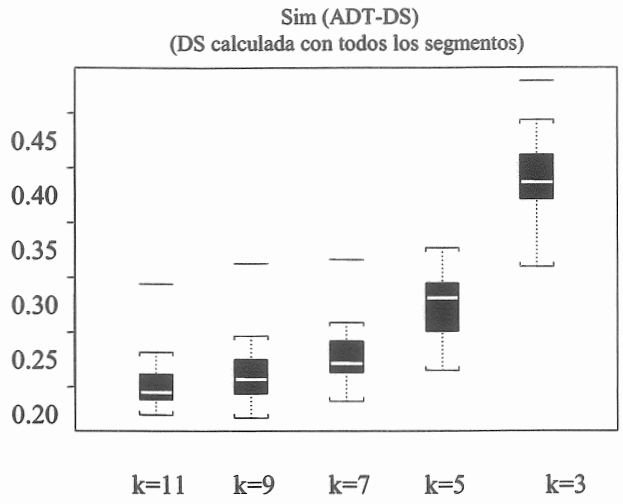


Figura 7. 6. Semejanza entre clusters generados por la distancia DS (usando todos los segmentos de la representación TA) y ADT.

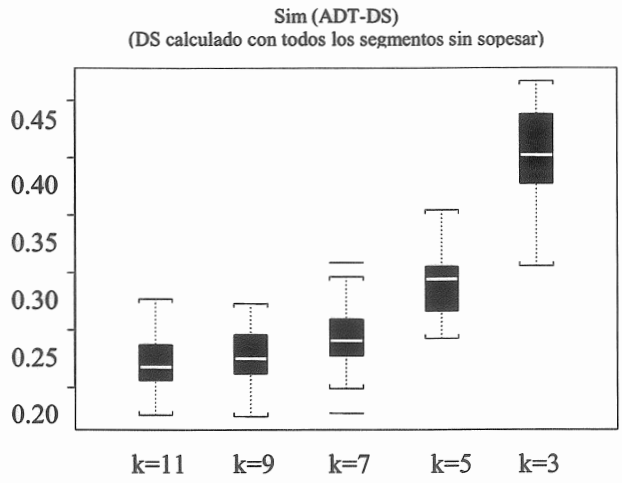


Figura 7. 7. Semejanza entre clusters generados por la distancia DS (usando todos los segmentos de la representación TA sin sopesado) y ADT. La medida de similitud entre clusters empleada es la dada en (95).

Beneficios de la paralelización

Hay ciertas leyes que limitan los beneficios que puedan derivar de la computación en paralelo. El objetivo en el área de cómputo a gran escala es el de conseguir realizar el máximo trabajo en el menor tiempo posible. La potencia de un sistema computacional puede ser definida como la cantidad de trabajo computacional que puede ser realizado, dividido por el tiempo que lleva el hacerlo. El "speedup" de un programa paralelo está definido según el porcentaje del tiempo en el que realiza un trabajo cuando es ejecutado con *n* procesadores con respecto al tiempo utilizado con un solo procesador. Será, por lo tanto, una función de *n*, el número de procesadores. Si *T(n)* es el tiempo requerido para completar la tarea en *n* procesadores , *S(n)* , el "speedup" , vendrá dado según (100).

$$S(n) = \frac{T(1)}{T(n)} \qquad (100)$$

En muchos casos (también en éste), el tiempo T(1) contiene una parte que es y será siempre serie, T_s , y una parte paralelizable, T_p . La parte serie no disminuye cuando la parte paralela se divide. En el mejor de los casos, la parte paralela decrecerá con un factor de $\frac{1}{n}$. El incremento en la velocidad que uno pueda esperar será por lo tanto el siguiente:

$$S(n) = \frac{T(1)}{T(n)} = \frac{Ts + Tp}{Ts + \frac{Tp}{n}} \qquad (101)$$

Esta expresión es conocida como la ley de Amdahl y se expresa normalmente como una desigualdad. El "speedup" logrado realmente será siempre menor o igual que dicha cantidad. Se grafican a continuación los resultados obtenidos del "speedup" y el tiempo real de cálculo para un solo nodo, 2 y 4 nodos de cálculo.

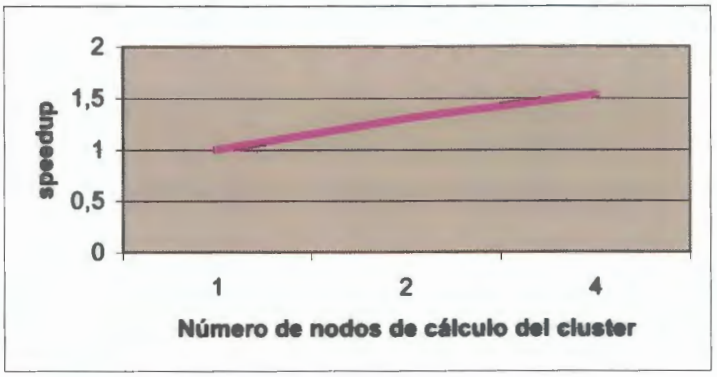


Figura 7. 8. Resultados obtenidos para la función *S(n)*, para una matriz de datos de 250 series de longitud 180 considerando 2, 3 y 5 nodos y empleando la fórmula dada en (100).

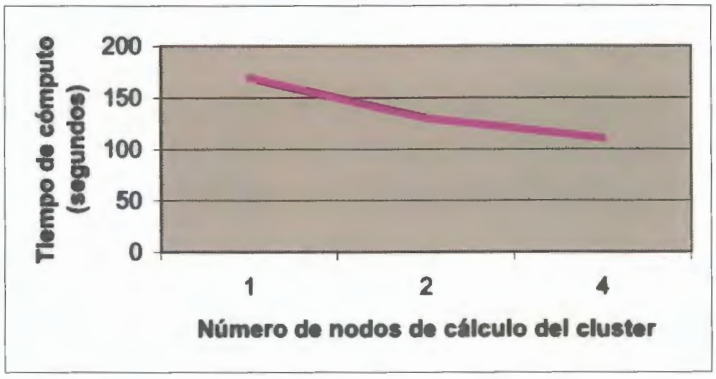


Figura 7. 9. Tiempo de cálculo real de la distancia de alineamiento para una matriz de datos de 250 series de longitud 180 para 2, 3 y 5 nodos en el cluster.

Conclusiones parciales

La distancia ADT es la distancia que mejores resultados ha obtenido en múltiples y variadas pruebas y la búsqueda de cotas inferiores ha sido un área que ha atraído mucha atención en los últimos años. Las cotas inferiores de la distancia ofrecen la posibilidad de realizar el cálculo de la verdadera distancia sólo cuando sea necesario siempre que busquemos series similares a una serie dada. Si deseamos utilizar la verdadera distancia y se desea realizar el cálculo en un tiempo razonable puede que la paralelización sea una de las alternativas posibles.

Las pruebas realizadas no son lo suficientemente amplias como para extraer un resultado definitivo pero pueden dar pistas sobre las ventajas que la paralelización pueda aportar en el cálculo de distancias costosas tal como ADT.

La paralelización será también una alternativa cada vez más utilizada en búsqueda de reglas de asociación en series temporales (Li et al. 2000).

CAPÍTULO 8

CONCLUSIONES Y LÍNEAS FUTURAS

El objetivo de esta tesis es explorar el área de minería de datos temporales y su uso en el análisis de series bursátiles. El dominio de aplicación ha determinado en gran parte el trabajo realizado, desde el momento en el que se ha dado prioridad al significado que tienen los patrones normalmente utilizados por los expertos en el área. El resultado logrado es un clustering de las series temporales que conserva similitudes semánticas entre los elementos del cluster. La metodología utilizada para lograr dicho objetivo ha sido la de evaluar la presencia de dichos patrones para poder, posteriormente, usar una medida de la similitud que, sin estar sujeta al dominio de aplicación, pueda lograr el objetivo deseado: juntar en el mismo

cluster series que, aún no siendo series con el mismo patrón, presentasen las suficientes semejanzas como para ser consideradas similares.

El dominio de aplicación presenta dificultades propias, varios expertos en el área no estarían, muchas veces, de acuerdo sobre la presencia de un cierto patrón en una serie temporal al ofrecerse ésta a múltiples interpretaciones.

La detección de los puntos de cambio, la búsqueda de la representación más adecuada para una serie temporal o bien la búsqueda de una medida de la similitud adecuada para la representación planteada, son los pilares sobre los que descansa la minería de series temporales. Las áreas de aplicación de esta disciplina están extendiéndose continuamente. Algunos métodos aplicados actualmente a las series temporales, pueden ser aplicados a otras áreas tales como las aplicaciones multimedia, imágenes de video, reconocimiento de firma ó clasificación de imágenes, siempre que los datos se conviertan a un formato similar. Por ejemplo, en (Vaklos, M 2004), se usan los términos empleados en el buscador MSN para generar una serie temporal en la que los valores se corresponden con el número de veces que un determinado término de búsqueda ha sido solicitado a lo largo del día.

El primer paso antes de que las series temporales se compriman, es transformarlas o normalizarlas de manera que los patrones similares puedan ser fácilmente identificados.

El preproceso aplicado a las series ha consistido en el suavizado con el fin de eliminar ruido. No era necesario tratar la translación puesto que las distancias DS y DB consideran dos series similares aunque se encuentren trasladadas en el eje de

ordenadas, y se decidió no tratar el escalado en amplitud y el crecimiento lineal, por considerar que dos series eran distintas entre sí cuando éstas transformaciones se encontrasen presentes.

En lo que respecta a la detección de los puntos de cambio, se han estudiado sucesivos valores del parámetro *h,* relacionándolos con el número de extremos obtenidos y la posterior detección de patrones, despejando algunas de las dudas sobre el valor óptimo del parámetro cuando éste sea utilizado para el proceso de detección de extremos.

En cuanto a la búsqueda de la representación más adecuada para las series temporales, se ha intentado conservar la que normalmente el experto usa, realizándose, posteriormente, valoraciones sobre la segmentación obtenida.

Es interesante tener en cuenta múltiples distancias en lugar de concentrarse en una sola y, de la misma manera, tener en cuenta varias representaciones. Se han realizado consideraciones con respecto a otras representaciones en lo que respecta al error de reconstrucción y hubiese sido posible explorar otras representaciones analizando, posteriormente, los resultados del clustering. Serán necesarios posteriores estudios para profundizar en este aspecto.

Es necesario recordar que ninguno de los métodos de reducción de la dimensionalidad es mejor que los demás en cualquier situación. De acuerdo a las características de la base de datos, un método puede dar mejores resultados que otro. La clave está en elegir el método que mejor se acomode al dominio de aplicación en el que se esté trabajando.

La distancia Euclídea y la distancia ADT son, hasta el momento, las mejores distancias y LB_Keogh es la cota inferior de ADT que más se aproxima a ella.

Las distancias DS y DB (realizados los estiramientos pertinentes en los segmentos constituyentes), permiten que dos series iguales entre sí pero desplazadas tanto en el eje x como en el eje y, sean consideradas similares, con lo que conservan ciertos de los atributos interesantes de ADT.

Comparando los clusters obtenidos con la representación TA, y distancias DS y DB, con los clusters obtenidos mediante la distancia Euclídea, ADT y LB_Keogh en la representación original, tenemos una referencia de las semejanzas que permanecen y desaparecen como consecuencia del proceso desarrollado. En lo que respecta a las distancias que interesan a la autora en la representación TA y que quedan por experimentar, se encuentran, al menos, la distancia con alineación temporal entre segmentos y aproximaciones de la distancia ADT.

Las aproximaciones de la distancia ADT se calcularían restringiendo la distorsión permitida. Se definiría para ello una región de coincidencia δ^2, de manera que la complejidad cuadrática en la longitud de la serie, $O(n^2)$, quedase reducida a $O(\delta n)$. Además, ésta puede ser una manera de mejorar la exactitud, impidiendo coincidencias no deseadas.

La región de coincidencia ó banda puede definirse deformada y sin deformar y es posible que el mejor harmónico encontrado para cada segmento pueda suministrar información sobre los anchos de banda de interés.

Por otro lado, la mayoría de la investigación se ha centrado, hasta el momento, en representaciones calculadas en modo batch, pero el desarrollo masivo de dispositivos móviles y sensores en tiempo real, ha creado la necesidad de obtener una representación que se pueda modificar de maneras incremental, aproximando los datos, por ejemplo, de acuerdo a su edad. Representaciones de este tipo se vuelven necesarias cuando las frecuencias a las que se generan los datos son tan altas que su manipulación o almacenamiento se vuelve impracticable. En éstos casos, puede que nos interese obtener más precisión en los datos más recientes.

Se denomina a este tipo de representaciones, representaciones amnésicas. Distintos dominios de aplicación podrían estar interesados en recordar los datos de distinta manera. Por ejemplo, si los datos fuesen el doble de viejos que los últimos disponibles, podría permitirme el lujo de conservarlos con un error que sea el doble de lo permitido en los datos más actuales.

Centrando nuestra mirada en el análisis técnico, sus gráficos muestran el escenario de una batalla entre compradores y vendedores. Para los analistas, es importante determinar la ocurrencia de patrones que puedan predecir subidas, bajadas y continuaciones de tendencias. El analista técnico trabaja en el dominio temporal y el trabajo mostrado en esta tesis también, puesto que se ha considerado que es el dominio más intuitivo y aquel en el que normalmente analizan los analistas los datos financieros. Todo ello, a pesar de que las Transformadas Wavelet y su capacidad de análisis a distintas escalas constituyen, actualmente, un mecanismo ampliamente utilizado en análisis de datos económicos. El proceso empleado de extracción de conocimiento útil a partir de las series temporales

utilizado ha sido semejante al que seguiría un analista técnico, pero la automatización del proceso de decisión no es fácil cuando los objetos con los que se trata son objetos complejos, como es el caso de las series temporales.

Decidir si una serie temporal contiene o no un patrón, incluso decidir qué patrón de todos los que contiene es el más adecuado para representarlo, es una acción que el analista realiza automáticamente empleando para ello toda la experiencia acumulada en su vida laboral. El experto deberá reconocer síntomas relacionados con el comportamiento y la psicología del mercado. Estos patrones se han repetido a sí mismos en sucesivos ciclos del mercado, la explicación debe ser que reflejan la psicología de las masas.

Una representación adecuada puede ser la llave para el descubrimiento de patrones válidos. La representación mediante segmentos lineales es muy utilizada y en este caso se ha considerado también la más apropiada. Es una representación sencilla, aunque el problema de la segmentación óptima no lo sea. Una representación basada en segmentos lineales permite un almacenamiento y un cómputo de los datos más eficiente. Además, usar segmentos lineales en la representación TA ha servido para conservar la percepción que el experto tiene al observar gráficos de series temporales bursátiles.

Los algoritmos que reciben series temporales y devuelven una representación lineal a tramos son los denominados algoritmos de segmentación. Se han utilizado los extremos detectados a lo largo de una serie temporal como los puntos candidatos para una segmentación, tal como se ha hecho en otros trabajos, (Chung, Fu-Lai 2004). Pruebas exploratorias dejaron claro que seleccionar unos

ciertos extremos de la serie temporal e imponer las restricciones respectivas a cada patrón podía generar patrones de poca calidad. La manera de mejorar esta calidad y de tener cierto control sobre el proceso de extracción ha sido el uso de parámetros: parámetro *h* que mide el suavizado de la serie, parámetro *l* que mide la resolución a utilizar en la extracción y el parámetro *lgp* que mide la longitud del patrón. El desafío, en este aspecto, es el descubrimiento de patrones frecuentes sin necesidad de especificar parámetros difíciles de determinar.

Los clusters obtenidos podrán ser utilizados para un acceso rápido a las series más semejantes a una serie dada en un espacio de baja resolución. La tarea de clustering, tan comúnmente usada como tarea previa en procesos de extracción de información, es, de todas maneras, una tarea que se debe emprender con cierto cuidado para poder obtener resultados significativos (Keogh and Lin 2003).

Futuras líneas de trabajo irán dirigidas a explorar otras tareas de minería de datos, con otras aplicaciones de la representación y medidas de la distancia planteadas. Es el caso de la minería de reglas de asociación, una tarea que utiliza como elementos las formas básicas y los patrones frecuentes. De la misma manera, ciertos algoritmos de clasificación de series temporales trabajan construyendo prototipos para cada clase y puede que los elementos de los clusters obtenidos puedan ser utilizados como prototipos para dicha labor.

Otras tareas, tales como la detección de anomalías, deben, inicialmente, modelar comportamientos comunes antes de poder detectar futuros patrones distintos a los patrones típicos.

El masivo uso de datos temporales ha hecho que el esfuerzo realizado en investigación y desarrollo en el área de minería de datos haya aumentado considerablemente la última década. Además del software específico para análisis y minería de series temporales, empresas proveedoras de productos para gestión de grandes bases de datos están realizando el esfuerzo de añadir a sus productos el software necesario para que estas tareas se integren dentro de las labores comunes a las bases de datos. Es el caso de Microsoft que está añadiendo a su software la capacidad de análisis de bases de datos temporales.

Es además un área intrínsecamente multidisciplinar donde la estadística, la computación paralela y el desarrollo de algoritmos de visualización, entre otros, tiene mucha importancia. El tratamiento de extensas bases de datos de series temporales de gran dimensionalidad obligará a utilizar algoritmos de paralelización que permitan obtener resultados en un tiempo razonable.

La computación en paralelo es un proceso interesante porque ofrece el potencial de concentrar los recursos computacionales, ya sea en procesamiento de datos, como en utilización de recursos de hardware y supone una alternativa a la hora de aligerar los costoso procesos de cálculo en minería de series temporales.

El estudio de modelos para series temporales tales como los modelos ARIMA(autoregressive integrated moving average) han resultado ser útiles en el modelado de series temporales tanto estacionarias como no estacionarias y los parámetros de dichos modelos utilizados para implementar medidas de la distancia entre series temporales. Estas aproximaciones adquieren más validez cuando se está tratando con series temporales largas donde las similitudes

definidas sobre la base de formas dan malos resultados. En éstos casos, será necesario definir similitudes sobre estructuras de más alto nivel, tal como pueden ser los modelos ARIMA, los parámetros de auto correlación o los parámetros de un modelo de Markov.

A veces, los algoritmos utilizados para el análisis de datos temporales resultan muchas veces inadecuadas para datos reales. Existen situaciones en las que la minería de datos temporales se tiene que aplicar sobre datos irregulares, o sobre datos temporales no se encuentran igualmente espaciados, que se encuentren distribuidos o que presenten cierta asincronía. A veces, es necesario realizar un análisis distribuido, de manera que el filtrado, la transformación y el análisis se haga lo más cerca posible de las fuentes de datos para evitar los tiempos prohibitivos de transmisión. Además, ciertos datos se almacenan sólo temporalmente, con lo cual, necesitan un análisis en tiempo real. Los propios datos pueden, a veces, ser heterogéneos, parte categóricos y parte numéricos.

En el caso concreto de las series temporales bursátiles los datos son afortunadamente fiables y son recogidos, normalmente, a intervalos regulares con lo cual no se plantean las problemáticas mencionadas anteriormente.

Se ha tratado, en ésta tesis, algunos de los problemas abiertos en el área del análisis multidimensional y minería de datos. Se han realizado algunas pruebas y expuesto los resultados pero la autora es consciente de que es sólo una pequeña contribución que espera pueda servir como introducción al extenso y vertiginoso área en estudio. Si resulta ser válido para que este excitante área despierte interés, el trabajo realizado verá su objetivo realizado.

BIBLIOGRAFÍA

1. **Agrawal R, Psaila G, Wimmers E and Zaït M.** Querying shapes of histories. *Proceedings of the 21 conference of VLDB*, 1995.

2. **Agrawal R.** Efficient similarity search in sequence databases, *Proc. of the Fourth Int'l Conference on Foundations of Data Organization and Algorithms, Chicago,* págs.. 69-84, 1993.

3. **Antunes CM and Oliveira AL.** Temporal Data Mining: an overview. *KDD 2001 workshop on temporal Data Mining* 2001.

4. **Basagoiti R.** Patrones de análisis técnico para la representación de series temporales bursátiles. *CEDI,* 2005.

5. **Basagoiti R and Juaristi E.** Clustering of time series using a similarity between segments and bans determined by patterns of technical analysis. *Data Mining 2006 Data, text and Web mining and their business application,* págs. 71-80, 2006.

6. **Beckmann N, Kriegel H and Schneider R.** The R*-tree an efficient and robust access method for points and rectangles. *ACM* 1990.

7. **Ben-hur A, Elisseff A and Guyon I.** A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing* 2002.

8. **Berchtold S, Böhm C and Kriegel H.** The Pyramid-Tree: breaking the curse of dimensionality. *SIGMOD Conference* 1998.

9. **Berndt DJ and Clifford J.** Using dynamic time warping to find patterns in time series. *KDD-94, AIII workshop on knowledge discovery in databases*, 1994.

10. **Bozkaya T and Ozsoyoglu ZM.** Distance based indexing for high dimensional metric spaces. *Proc. ACM SIGMOD International Conference on Management of data* 1997.

11. **Bozkaya T, Yazdani N and Ozsoyoglu ZM.** Matching and Indexing Sequences of different lengths. *Proc. Sixth Int. Conf on Information and Knowledge Management CIKM* 1997.

12. **Chu S, Keogh E, Hart D and Pazzani M.** Iterative deepening dynamic time warping for time series. *Proceedings of the Second SIAM Int. Conf. on Data Mining* 2002.

13. **Chu K and Wong M.** Fast time-series searching with scaling and shifting. *Proceedings of the 18 ACM Symposium on Principles of Database Systems*, págs. 126-133, 1999.

14. **Chung F, Fu T, Ng V and Luk RWP.** An evolutionary approach to pattern-based time series segmentation. *Evolutionary Computation, IEEE Transactions on Volume 8, Issue 5,* págs. 471-489, 2004.

15. **Das G, Gunopulos D and Mannila H.** Finding similar time series. *PKDD*, págs. 88-100, 1997.

16. **Elman JL.** Finding structure in time. *Cognitive Science* 1990.

17. **Faloutsos C.** Searching multimedia databases by content. Kluwer Academic Publishers, 1996.

18. **Faloutsos C, Ranganathan M and Manolopoulos Y.** Fast subsequence matching in time-series databases. págs. *Proceedings 1994 ACM SIGMOD Conference, Mineapolis, MN*, págs 419-429,1994.

19. **Faloutsos C and Lin K.** Fast Map: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *Proceedings of 1995 ACM SIGMOD, SIGMOD RECORD (June 1995), vol.24, no.2, SIGMOD Conference*, págs. 163-174, 1995.

20. **Faloutsos C, Jagadish H, Mendelzon A and Milo T.** A signature technique for similarity-based queries. *Proceedings of the Int. Conference on Compression and Complexity of Sequences* 1997.

21. **Fink E and Pratt KB.** Indexing of compressed time series. *Data Mining in time series databases, World scientific*, págs. 51-78, 2003.

22. **Fink E, Pratt KB and Gandhi HS.** Indexing of time series by major minima and maxima. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics.* 2003.

23. **Focardi SM.** Clustering economic and financial time series: exploring the existence of estable correlation conditions. *The Intertek group. Disscusion Paper* 2001-04.

24. **Fowkles EB and Mallows CL.** A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* 1983.

25. **Gavrilov M, Anguelov D, Indyk P and Motwani R.** Mining the stock market: wich mesure is best? *Proceedings of the KDD* 487, 2000.

26. **Ge X.** Segmental semi-markov models for change-point detection with applications to semiconductor manufacturing. *Technical Report UCI-ICS 00-08,* 2000.

27. **Ge X.** Pattern matching in financial time series data. 1998.

28. **Ge X and Smyth P.** Deformable markov model templates for time-series pattern matching. *KDD*, págs. 81-90, 2000.

29. **Gençay R, Selçuk F and Whitcher B, eds.** An introduction to Wavelets and other filtering methods in finance and economics. *Academic Press.*, 2002.

30. **Goffe WL.** Wavelets in macroeconomics: an introduction. *Computational techniques for Econometrics and Economic Analysis, Kluwer Academic Publishers*, págs. 137-149, 1994.

31. **Guralnik V and Srivastava J.** Event detection from time series data. *KDD-99,* 1999.

32. **Guttman A.** R-trees, A dynamic index structure for spatial searching. 1984.

33. **Han J.** Efficient mining of partial periodic patterns in time series databases. *Proc. 1999 Int. Conf. Data Engineering (ICDE'99), Sydney, Australia,* , págs. 106-115, 1999.

34. **Han J, Gong W and Yin Y.** Mining segment-wise periodic patterns in time-related databases. *In Proc. 1998 Int'l Conf. on Knowledge Discovery and Data Mining (KDD'98)* 1998.

35. **Han KP, Fu A and Yu C.** Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Transactions,* 2003.

36. **Härdel W and Müller M.** Applied nonparametric regression. *Cambridge University Press*, 1990.

37. **Hsu WH and Ray SR.** Quantitative model selection for heterogeneous time series learning. *Machine Learning AAAI Press,* 1998.

38. **Huang Y and Yu PS.** Adaptive query processing for time series data. *KDD-99,* 1999.

39. **Huhtala Y, Kärkkäinen J and Hannu T.** Mining for similarities in aligned time series using wavelets. *Data mining and Knowledge Discovery: Theory, Tools, and Technology, volume 3695 of SPIE Proc.,* págs. 150-160, 1999.

40. **Kahveci T and Singh A.** Variable length queries for time series data. *Proceedings of the 17 Int. Conference on Data Engineering.* 2001.

41. **Kalpakis K, Gada D and Puttagunta V.** Distance measures for effective clustering of ARIMA time series. *ICDM,* Págs. 273-280, 2001.

42. **Kanth, K. V. Ravi, Agrawal D and El Abbadi A.** Dimensionality reduction for similarity searching in dynamic databases. *Proc. Of 1998 ACM SIGMOD Int. Conf. On Management of Data* 1998.

43. **Keogh E.** Exact indexing of dynamic time warping. *VLDB* 2002.

44. **Keogh E.** An online algorithm for segmenting time series. *In Proceedings of the IEEE International Conference on Data Mining,* págs. 289-296, 2001a.

45. **Keogh E.** Derivative dynamic time warping. *First SIAM International conference on Data Mining,* 2001b.

46. **Keogh E.** Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems* 2000a.

47. **Keogh E.** Scaling up dynamic time warping for data mining applications. *In Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining,* págs. *285-289,* 2000b.

48. **Keogh E.** An indexing scheme for fast similarity search in large time series databases. *In Proc. Eleventh International Conference on Scientific and Statistical Database Management,* págs. *56-67,* 1999a.

49. **Keogh E.** Scaling up dynamic time warping to massive datasets. *Proc. Principles and Practice of Knowledge Discovery in Databases*, págs. *1-11,* 1999b.

50. **Keogh E.** An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. *Fourth International Conference on Knowledge Discovery and Data Mining (KDD)'98*, págs. 239-241, 1998.

51. **Keogh E.** A fast and robust method for pattern matching in time series data bases. *ICTAI,* págs. *578-584,* 1997a.

52. **Keogh E.** Fast similarity search in the presence of longitudinal scaling in time series. *ED Proceedings of the 9th Intr. Conf. On Tools with Arficial Intelligence*, págs. 578-584, 1997b.

53. **Keogh E.** A probabilistic approach to fast pattern matching in time series databases. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97)* , págs. 24-30, 1997c.

54. **Keogh E.** A decade of progress in indexing and mining large time series databases. *Proceedings of the 32nd international conference on Very large data bases* 32: 2006.

55. **Keogh E, Chakrabarti K, Pazzani M and Mehrotra S.** Locally adaptive dimensionality reduction for indexing large time series databases. *Proceedings of ACM SIGmod conference on Management of data*, págs. 151-162, 2001.

56. **Keogh E and Kasetty S.** On the need for time series data mining benchmarks: a survey and empirical demonstration. *SIGKDD* 2002.

57. **Keogh E and Pazzani M.** A simple dimensionality reduction tecnique for fast similarity search in large time series databases. *PKDD* 2000.

58. **Keogh E, Chu S, Hart D and Pazzani M.** Segmenting time series: a survey and novel approach. In: *data mining in time series databases*. Singapore: world scientific, 2004, chapt. 1, p. 1-21.

59. **Keogh E, Lin J and Fu A.** HOT SAX: efficiently finding the most unusual time series subsequence. *ICDM* 226-233, 2005.

60. **Keogh E and Lin J.** Clustering of time series subsequences is meaningless: implications for previous and future research. *ICDM* 2003.

61. **Kim S, Park S and Chu W.** An index-based approach for similarity search supporting time warping in large sequence databases. *ICDE 01*, págs. 607-614, 2001.

62. **King-pong Chan.** Efficient time series matching by wavelets. *ICDE*, págs. 126-133, 1999.

63. **Korn F, Jagadish H and Faloutsos C.** Efficiently supporting ad hoc queries in large datasets of time sequences. *Proceedings of the ACM SIGMOD Int Conference on Management of Data*, págs. 289-300, 1997.

64. **Le Gall D.** Mpeg: a video compression standard for multimedia applications. *Comm. Of ACM (CACM),34(4)*, págs. 46-58, 1991.

65. **Leach S.** Singular value decomposition-a primer. *Unpublished Manuscript, Department of Computer Science, Brown University,*

66. **Li C-S, Yu PS and Castelli V.** Hierarchyscan: A hierarchical similarity search algorithm for databases of long sequences. *Proc. Int Conf. On Data Engineering,* 1996.

67. **Li C, Yu PS and Castelli V.** MALM: A framework for mining sequence database at multiple abstraccion levels. *Proceedings of the 7 ACM CIKM Int Conference on Information and Knowledge Management*, 1998.

68. **Lin J, Keogh E, Lonardi S and Chiu B.** A symbolic representation of time series, with implications for streaming algorithms. *ACM SIGMOD* . 2003.

69. **Li Y, Lin X and Tsang CP.** An Efficient Distributed Algorithm for Computing Association Rules. *Proceedings of the First International Conference on Web-Age Information Management* 109-120, 2000.

70. **Lkhagva B, Suzuki Y and Kawagoe K.** Extended SAX: Extension of symbolic aggregate aproximation form finantial time series data representation. *DEWS* 2006.

71. **Lo AW, Mamaysky H and Wang J.** Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation. *The Journal of finance, vol. LV, n° 4.* 2000.

72. **Mendelzon AO and Milo T.** Similarity-based queries. *Int. Symposium on principles of Database Systems (PODS)* 1995.

73. **Ming Dong X.** Exploring the fuzzy nature of technical patterns of U.S. stock market. *Proc. ICONIP'02-SEAL'02-FSKD'02.* 2002.

74. **Murphy JJ.** Análisis técnico de los mercados financieros. *Ediciones Gestión*, 2000.

75. **Park S, Kim S and Chu W.** Segment-based approach for subsequence searches in sequence databases. *Proceedings of the 16 ACM Symposiun on Applied Computing*, págs. 248-252, 2001.

76. **Park S., and D. Lee and W. Chu.** Fast retrieval of similar subsequences in long sequence databases. *Technical report, University of California, Los Angeles, UCLA-CS-TR990028*, 1999.

77. **Patel P, Keogh E, Lin J and Lonardi S.** Mining motifs in massive time series databases. *Proceedings of IEEE International Conference on Data Mining (ICDM'02)* 2002.

78. **Pei J, Han J, Lu H, Tang S, Nishio S and Yang D.** H-mine: hyper-structure mining of frequent patterns in large databases. *In Intl. Conf. on Data Mining (ICDM),* 2001.

79. **Perng C, Wang H, Zhang SR and Parker DS.** Landmarks: a new model for similarity-based pattern querying in time series databases. *16th International Conference on Data Engineering (ICDE'2000)* 2000.

80. **Polly WPM and Wong MH.** Efficient and robust feature extraction and pattern matching of time series by a lattice structure. *Proceedings of the*

10 ACM CIKM Int Conference on Information and knowledge Management, 2001.

81. **Popivanov I.** Efficient similarity queries over time series data using wavelets. *Master's thesis, University of Toronto, Toronto, Canada*, 2001.

82. **Popivanov I and Miller RJ.** Similarity search over time series data using wavelets. *Proceedings of the 18 Int Conference on Data Engineering*, págs. 212-221, 2002.

83. **Pratt KB.** Locating patterns in discrete time-series. *Master's thesis, Computer Science and Engineering, University of South Florida*, 2001.

84. **Pratt KB and Fink E.** Search for patterns in compressed time series . *International Journal of Image and Graphics*, 2002.

85. **Rabiner L and Juang B.** Fundamentals of speech recognition. *Prentice Hall*, 1993.

86. **Rafiei D.** On similarity-Based Queries for Time Series Data. *IEEE* 1999.

87. **Rafiei D and Mendelzon A.** Similarity based queries for time-series data. *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data* 1997.

88. **Rafiei D and Mendelzon A.** Efficient retrieval of similar time sequences using DFT. *Proceedings of the 5 Int Conference on Foundations of Data Organizations and Algorithms,* 1998.

89. **Ratanamahatana CA and Keogh E.** Making time-series classification more accurate using learned constraints. *Proceedings of SIAM International Conference on Data Mining*, 2004.

90. **Rath TM and Manmatha R.** Lower-bounding of dynamic time warping distances for multivariate time series. *Technical Report MM-40, University of Massachusetts Amherst*, 2002.

91. **Roth WG.** MIMSY: A system for analyzing time series data in the stock market domain. *University of Wisconsin, Department of Computer Science* 1993.

92. **Salzberg B and Tsotras VJ.** Comparison of access methods for time evolving data. *ACM Computing Surveys 31(2),* págs. 158-221, Junio 1999.

93. **Shahabi C, Tiang X and Zhao W.** TSA-tree: A wavelet based approach to improve the efficiency of multi level surprise and trend queries on time series data. 2000.

94. **Shatkay H.** The Fourier transform-A primer. 1995.

95. **Shatkay H and Zdonik SB.** Approximate queries and representation for large data sequences. *Proceedings of the 12 IEEE International Conference on Data Engineering,* págs. 546-553, 1996.

96. **Smyth P and Keogh E.** Clustering and mode classification of engineering time series data. 1996.

97. **Srikant R and Agrawal R.** Mining sequential patterns: generalizations and performance improvements. *Procceedings of the Fifth Int'l Conference on Extending Database Technology (EDBT)* 1996.

98. **Stollnitz E, DeRose T and Salesin D.** Wavelets for computer graphics A primer. *IEEE Computer Graphics and Applications* , págs. 76-84, 1995.

99. **Struzik Z and Sibes A.** Measuring time series similarity through large singular features revealed with wavelets transformation. *In Proc. Of the 10th Intl. Workshop on Database and Expert Systems Applications*, págs. 162-166, 1999.

100. **Struzik ZR and Siebes A.** The haar wavelet transform in the time series similarity paradigm. *Principles of Data Mining and Knowledge Discovery* 1999.

101. **Vlachos M, Lin J, Keogh E and Gunopulos D.** A wavelet-based anytime algorithm for K-means clustering of time-series. *3rd SIAM*

International Conference on Data Mining. Workshop on Clustering High Dimensionality Data and Its Applications 2003.

102. **Vlachos M.** Similarity and indexing in multidimensional spaces. 2004.

103. **Wai-Chee Fu A, Tat-Wing Leung O, Keogh E and Lin J.** Finding Time Series Discords Based on Haar Transform. *ADMA* 31-41, 2006.

104. **Wand M and Jones M.** Kernel smoothing, monographs on statistical and applied probability. *Chapman & Hall*, 1995.

105. **Wang C and Wang XS.** Supporting content-based searches on time series via approximation. *Proceedings of the 12 Int Conference on scientific and Statistical Database Management*, págs. 69-81, 2000.

106. **Wu Y, Agrawal D and El Abbadi A.** A comparison of dft and dwt based similarity search in time-series databases. *Proceedings of the 9 ACM CIKM International Conference on Information and knowledge Management*, 2000.

107. **Yazdani O.** Sequence matches of images. *Proc. 8th International Conference on Scientific and Statistical Databases* 1996.

108. **Yi B, Faloutsos C and Biliris A.** Fast time sequence indexing for arbitrary lp norms. *VLDB Conf (VLDB20000)* 2000.

109. **Yi B, Jagadish HV and Faloutsos C.** Efficient retrieval of similar time sequences under time warping. *ICDE,* págs. 201-208, 1998.

110. **Zeira G, Maimon O, Last M and Rokach L.** Change point detection in classification models induced from time series data. In: *Data mning in time series databases*. Singapore: World Scientific, 2004, chapt. 5, p. 101-124.