UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Doble Grado en Ingeniería Informática y Matemáticas

TRABAJO FIN DE GRADO

CODIFICACIÓN EN RANDOM FOREST PARA EL PROCESAMIENTO DE SERIES TEMPORALES

Ana de Santos Martín

Tutor: Pablo Varona Martínez

JUNIO 2019

CODIFICACIÓN EN RANDOM FOREST PARA EL PROCESAMIENTO DE SERIES TEMPORALES

AUTOR: Ana de Santos Martín

TUTOR: Pablo Varona Martínez

Grupo de Neurocomputación Biológica

Dpto. Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Junio 2019

Resumen

Este Trabajo Fin de Grado tiene como objetivo mejorar la clasificación de series temporales multivariadas a partir de la selección de una codificación temporal óptima en el algoritmo conocido como Symbolic representation for Multivariate Time Series (SMTS), así como estudiar las dependencias con ciertas características que proporcionan las señales.

Actualmente existen grandes cantidades de datos con numerosa información temporal en multitud de campos, por lo que, las series, tanto univariadas como multivariadas, han ido adquiriendo mucha importancia a lo largo del tiempo. Es por ello que el problema de clasificación es cada vez más complejo, ya que ha aumentado la cantidad de datos a procesar, así como la calidad de los mismos o el uso de ciertos fatores desconocidos que afectan a la evolución temporal.

Uno de los algoritmos propuestos para abordar este problema es el SMTS, el cual permite clasificar las series multivariadas, considerando todos los atributos mencionados de forma simultánea para conseguir extraer la información contenida en las relaciones. A pesar de que este algoritmo consigue porcentajes altos en la clasificación, en este trabajo se tratará de optimizar el rendimiento presentado por el algoritmo para las MTS mediante una propuesta de codificación temporal con Random Forest.

Para conseguir la codificación temporal óptima en este algoritmo, primero se realizará un estudio de los conjuntos de datos de manera individual, con el fin de obtener las características temporales de las señales y de las clases por separado. Tras esto, se explorará el parámetro temporal óptimo para la clasificación en función de cada serie. Esta exploración se realiza con series temporales multivariadas de distinto tipo y estructura temporal.

Palabras clave

Clasificación de series temporales, Random Forest, Codificación temporal, SMTS

Abstract

The purpose of this Bachelor Thesis is to improve the time series classification from an optimal temporal coding selection in the algorithm called Symbolic representation for Multivariate Time Series (SMTS), as well as to study the dependencies with certain features provided by the signals.

There are currently large amounts of data with numerous temporal information in a wide array of fields which is why both univariate and multivariate series have become increasingly important over the years. Hence, the issue of classification is even more complex given that the amount of data to be processed as well as its quality have increased, and also the presence of certain unknown factors that can affect the temporal evolution.

One of the algorithms proposed to address this problem is the SMTS, which makes it possible to classify multivariate time series, considering all the attributes mentioned simultaneously in order to extract information contained in the associated relationships. Even though this algorithm attains high percentages in classification, in this project, we will attempt to optimise the performance to classify MTS through a proposal of temporal coding with Random Forest.

In order to achieve the optimal temporal coding in this algorithm, first, an individual study of the data sets will be conducted with the aim of characterizing the temporal features of the signals and the classes separately. Thereafter, the optimal parameters will be explored for the classification for each time series. This exploration will be done for multivariate time series of different types and different temporary structure.

Keywords

Time series classification, Random Forest, Temporal coding, SMTS

Agradecimientos

A mi padre, que no ha podido ver completada esta etapa. A mi madre, por guiarme y ayudarme durante este camino. Todo lo conseguido es gracias a vosotros.

A Ángel y Alejandro, por querer compartir mi camino con vosotros.

A mis compañeros de la carrera, por crear una familia.

A mi tutor, Pablo Varona por guiarme en este proyecto y por la confianza para realizarlo.

A ti, por leerlo.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 Introducción

1.1 Motivación

Actualmente existe una cantidad enorme de datos registrados con información temporal en una gran variedad de campos como la medicina, biotecnología, economía, medioambiente, marketing, etcétera. Las series temporales multivariadas (MTS) han adquirido mucha importancia en las últimas décadas, ya que el número de conjuntos de datos ha aumentado exponencialmente.

El problema de clasificación de series temporales multivariadas es complejo porque a menudo los datos son ruidosos, contienen deriva, dependencia no lineal de la historia previa y contienen factores desconocidos que afectan a la evolución temporal de cada señal de forma muy distinta (Abanda, Mori, & Lozano, 2019a; Baydogan & Runger, 2015a). Este problema de clasificación está siendo estudiado en profundidad cada vez más con el paso del tiempo; utilizando una gran cantidad de algoritmos para disminuir la tasa de error en la clasificación, atendiendo así a las características propias de cada señal.

Los algoritmos utilizados para abordar ese problema están típicamente diseñados para aprender a medida que el problema va avanzando y no tanto para resolver un problema específico. Por ello, la búsqueda de nuevos problemas en los diferentes campos en los que se estudian las series temporales será cada vez más amplio y con algoritmos que estén capacitados para resolver problemas de grados de dificultad más elevada a la actual.

En este momento, los problemas de clasificación son cada vez más complejos debido a que no solo se tienen series temporales que dependan únicamente de un atributo, sino que se necesita de más características para estudiar una señal. Uno de los algoritmos que permite clasificar series multivariadas es el SMTS (Symbolic representation for MTS), que considera todos los atributos de las MTS simultáneamente, en lugar de por separado, para extraer la información contenida en las relaciones. Los símbolos aprenden de un algoritmo supervisado que no requiere intervalos predefinidos ni características (Baydogan & Runger, 2014).

Para completar el algoritmo SMTS, se usa Random Forest, un método que, como todo clasificador, requiere de dotar al algoritmo de aleatoriedad para maximizar la independencia de los árboles en este caso, pero manteniendo la precisión en todo momento. Este clasificador es utilizado, en primer lugar, porque al clasificar series temporales multivariadas, Random Forest no excluye ningún atributo de las series.

Para clasificar series temporales no se ha parado a pensar en la estructura temporal, un clasificador va a mejorar el porcentaje de acierto si se estudia la variabilidad temporal de la señal. Una vez propuesto una mejora en este aspecto, se deberá estudiar una optimización de los parámetros, algo muy avanzado en otras ramas de la clasificación de series temporales.

1.2 Objetivos

El objetivo principal de este trabajo es estudiar la dependencia de la codificación temporal en relación con algunas de las características que proporcionan las señales y que mejoran el porcentaje de acierto de clasificación de las series temporales tanto univariadas como multivariadas teniendo en cuenta la codificación temporal.

Como aproximación a la clasificación de las series temporales, existen varios métodos para tratar este problema. Uno de ellos es el algoritmo SMTS, con el que se abordará el problema el estudio de la codificación temporal de la entrada, acorde a las señales, para mejorar la clasificación.

En primer lugar, se realizará un estudio de varios conjuntos de datos por separado, para ver las características temporales de cada señal y de cada clase. Después de esto, veremos cómo un método de clasificación basado en random forest, que tiene en cuenta la codificación temporal respetando la temporalidad de las señales, mejora la clasificación de las series temporales. El objetivo principal conlleva también elegir el parámetro óptimo para la clasificación dependiendo de cada serie temporal. Este parámetro tiene que escogerse en función de la señal ya que puedan llegar a proporcionar una pérdida de la información.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Estado del arte:** En esta sección del trabajo se presenta una introducción a la clasificación de las series temporales además de sus tipos, así como la explicación en el caso de Random Forest que preserva la estructura temporal de la señal procesada.
- **Diseño y desarrollo:** En este capítulo se describirá el algoritmo utilizado para la clasificación de las señales, además de la propuesta teniendo en cuenta las características que mejoran su clasificación. En esta sección también se explicarán los conjuntos de datos, diferenciándolos entre series multivariadas y univariadas.
- **Pruebas y resultados:** En este capítulo se discutirán los resultados de la clasificación de los diferentes conjuntos de datos.
- **Conclusión y trabajo futuro:** En esta sección se muestra las conclusiones del trabajo realizado, así como el trabajo que se puede llevar a cabo en el futuro.

2 Estado del arte

En este capítulo se estudiará el problema de la clasificación de series temporales, así como sus tipos, concluyendo con las ventajas de considerar codificación adaptadas a preservar estructura temporal en general y particularmente en el caso de los *Random Forest*.

2.1 Introducción a las series temporales

Una serie temporal es una secuencia de datos extraídos en orden en diferentes intervalos de tiempo. Estos pueden estar espaciados a intervalos desiguales (por ejemplo, en el control de constantes vitales de una persona a lo largo del tiempo en un centro de salud) o con periodicidad (medición de niveles diarios de CO_2 en una ciudad).

Dado un conjunto de datos de *n* series temporales, $\boldsymbol{T} = \{T_1, T_2, \ldots T_n\}$, cada serie T_i tiene *m* datos de valores ordenados y una clase c_i. Sea un *dataset* ***T***, la clasificación de las series temporales consiste en encontrar un algoritmo o una función que mapee en las posibles clases cada una de las series compuestas por esos instantes de tiempo (Hills, Lines, Baranauskas, Mapp, & Bagnall, 2014).

Para el análisis de series temporales, se usan métodos que permiten obtener información relevante sobre la estructura temporal de los datos de la serie o entre diferentes series para poder así caracterizar o pronosticar el comportamiento de una serie temporal, ya sea en el futuro, en instantes intermedios o en el pasado.

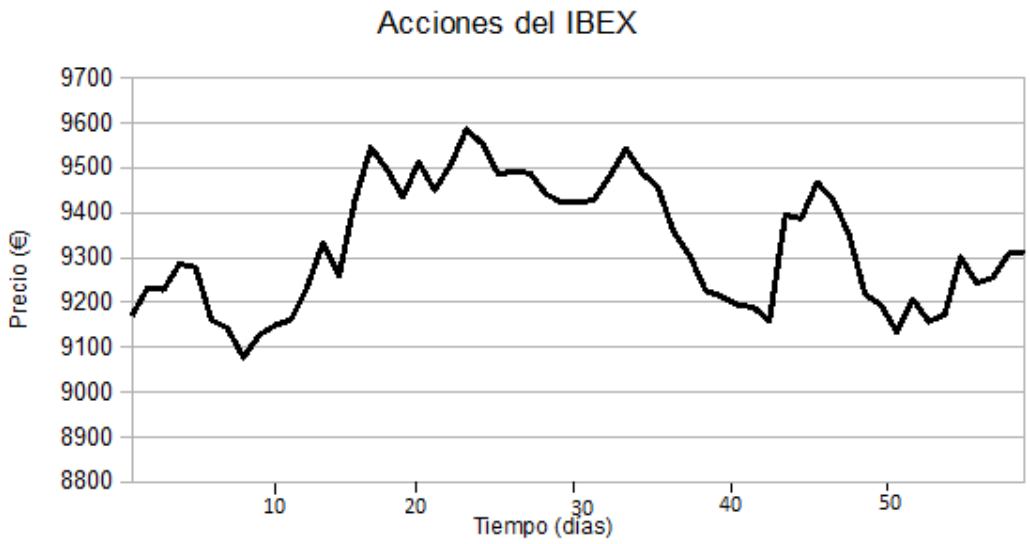

Figura 2-1: Representación de la serie Temporal del precio de las acciones del IBEX durante 60 días.

2.2 El problema de clasificación de series temporales

El problema de la clasificación de series temporales ha sido tema de numerosas investigaciones en estas últimas décadas (Cohen & Cohen, 1993; Xing, Pei, & Keogh, 2010a).

La clasificación de series temporales es un problema que ha crecido en la actualidad, debido a la gran cantidad de datos de la que se dispone y que han sido recolectados, fundamentalmente por el desarrollo de tecnologías de sensores, de almacenamiento masivo y la explosión de tecnologías *big-data*. Todos estos datos pueden ser obtenidos en distintas áreas como la ingeniería, finanzas, biomedicina, etc. La creación de algoritmos para su clasificación también han aumentado en las últimas décadas (Abanda, Mori, & Lozano, 2019).

Para la clasificación se utilizará, en este trabajo, una codificación que preserve la estructura temporal inicial y particularmente se hablará de una versión del algoritmo *Random Forest*.

2.2.1 Tipos de clasificación

Los métodos de clasificación se pueden dividir en 3 grandes categorías: (i) clasificación basada en características, (ii) clasificación basada en la distancia de las secuencias y (iii) clasificación basada en modelos (Abanda et al., 2019b).

En el primer tipo, basado en características, la serie temporal se transforma en un vector de características para más tarde aplicar los métodos usuales de clasificación como los árboles de decisión o las redes neuronales. En el segundo tipo, basado en la distancia, utiliza una función o métrica de distancia que mide la similitud o la diferencia entre las secuencias. Ejemplos de este método sería el uso de la distancia Euclídea, mientras que vecinos próximos (K-NN) o el soporte de Máquinas de vectores (SVM) son ejemplos que se pueden encontrar tanto en el tipo 1 como en el 2.(Xing, Pei, & Keogh, 2010b). Y, por último, en la llamada clasificación basada en patrón/modelos, se asume que todas las series temporales están generadas por el mismo patrón, y por lo tanto se asigna una nueva serie que mejor se adapte a la clase del modelo. Algunos de estos enfoques se crean utilizando modelos ocultos de Markov y modelos autorregresivos (Bagnall & Janacek, 2014), entre otros.

En la primera categoría, después de calcular varias funciones para cada serie temporal de un conjunto de datos, se seleccionan las categorías que han aportado más información de la estructura temporal de la clase mediante un clasificador lineal. Los clasificadores resultantes basados en características aprenden automáticamente de las diferencias entre las clases usando un número reducido de propiedades de las series temporales y evitan la necesidad de calcular distancias entre series de tiempo. De esta manera, en la representación de series temporales se reduce la dimensionalidad, lo que permite que este método tenga un buen desempeño en conjuntos de datos muy grandes o series de diferentes longitudes (Fulcher & Jones, 2014).

Por otro lado, una representación basada en distancia , puede reducir la dimensionalidad transformando el espacio original en uno construido mediante la concatenación de funciones distancia (López-Iñesta, Grimaldo, & Arevalillo-Herráez, 2015).

En la clasificación basada en modelos, en la fase de entrenamiento se aprenden los parámetros de los modelos de probabilidad (M). En la fase de clasificación, una nueva secuencia es asignada a la clase con la mayor probabilidad (Xing et al., 2010b). Un ejemplo de estrategia para realizar el análisis sería el método de Naive Bayes, ya que , este algoritmo es fácil de implementar y típicamente se obtienen buenos resultados además de que necesita un número pequeño de datos para el entrenamiento. Por otro lado, por el contrario, se debe asumir que las variables tienen dependencia condicional respecto a la clase y esto puede acarrear una falta de imprecisión en el análisis

2.3 Random Forest

La clasificación en *Random Forest* es una combinación de árboles de decisión, dónde cada árbol depende de los valores de un vector aleatorio probados independientemente y con la misma distribución para cada uno de estos (Breiman, 2008). La creación de una gran colección de árboles y más tarde promediándolos, modificando así el llamado *Bootstrap Aggregation o Bagging* (Empaquetamiento), da como resultado un clasificador muy poderoso.

2.3.1 Definición de *Random Forest*

Random Forest es un clasificador que consiste en una colección de clasificadores estructurados en árboles $\{h(\boldsymbol{x}, \boldsymbol{\Theta_k}, \boldsymbol{D}), k=1, ...\}$, dónde $\{\boldsymbol{\Theta_k}\}$ son vectores aleatorios independentes e idénticamente distribuidos, cada árbol emite un único voto para la clase más popular en la entrada $\boldsymbol{x}$.

Para cada *dataset* de entrenamiento $\boldsymbol{D} = \{(x_1,y_1), ...(x_n,y_n)\}$ dónde x_i son los predictores e y_i denota la respuesta(Cutler & Cutler, 2012).

En la figura que aparece a continuación, se muestra de forma esquemática el funcionamiento del clasificador *Random Forest*, donde podemos observar, que cada árbol de decisión, creado mediante la comparación aleatoria de los atributos, representa una clase en el conjunto de datos.

En la figura 2-2, se muestra como el camino seguido por los nodos amarillos nos va a definir la clase para su futura clasificación. En el momento de la predicción, se toman un promedio de las estimaciones de cada árbol.

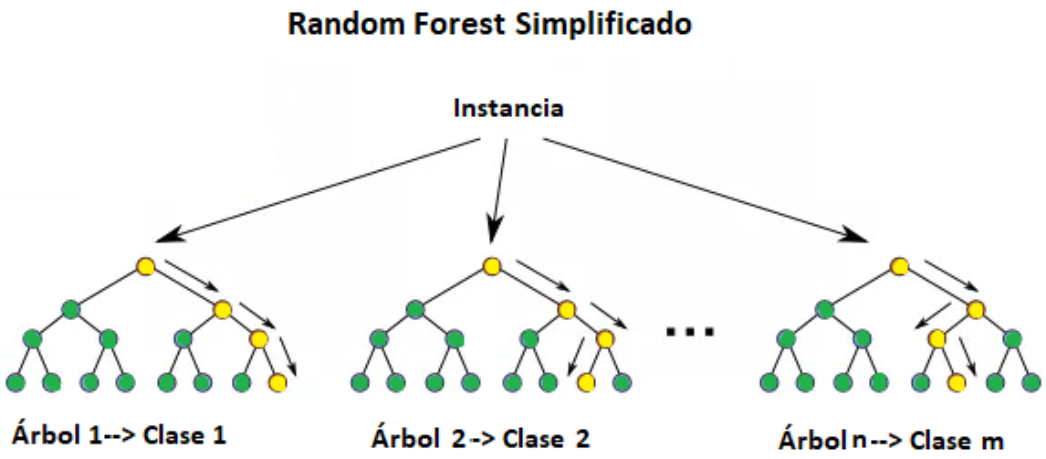

Figura 2-2: Visualización esquemática del RF.

2.3.2 Ajuste al algoritmo *Bagging*

Bagging es un algoritmo de aprendizaje diseñado para mejorar la estabilidad y precisión de los clasificadores usados para la clasificación y en regresión. Este algoritmo también disminuye la varianza y elude el sobreajuste. Aunque normalmente se aplica a los clasificadores de árboles de decisión, se puede usar con cualquier de método de clasificación (Breiman, 1996; Kotsiantis, 2014).

Un problema con los árboles de decisión como por ejemplo los árboles de clasificación y regresión (CART, por sus siglas en inglés) es que son bastante codiciosos. Típicamente eligen qué variable dividir usando un algoritmo que minimice el error. Por definición, incluso con *Bagging*, los árboles de decisión pueden tener gran cantidad de similitudes estructurales y a su vez hacen que tengan alta correlación en sus predicciones (Brownlee, 2016).

La combinación en conjunto de las predicciones por múltiples modelos funciona mejor si las predicciones de los submodelos no están correlacionadas o en el mejor de los casos lo están débilmente.

Random Forest cambia el algoritmo de la forma que los subárboles han aprendido, por lo que el resultado de las predicciones de todos los subárboles tiene menos correlación.
Este algoritmo realiza una pequeña modificación de los árboles CART. En estos últimos, el algoritmo de aprendizaje puede revisar todas las variables y sus valores para seleccionar el punto óptimo. Por su parte, el algoritmo *Random Forest*, modifica este procedimiento, limitando a una muestra aleatoria en la que buscar.

El número de características que pueden ser buscadas en cada punto **(*m*)**, debe ser especificado como parámetro del algoritmo.

2.3.3 Características y ventajas

A continuación, se expondrán los rasgos más ventajosos del algoritmo *Random Forest*.

Este algoritmo puede proporcionar un aprendizaje eficaz a partir de los datos de entrenamiento (para un conjunto de datos lo suficientemente grande produce un clasificador que se ajusta correctamente). Además, tiene la habilidad de manejar centenares de características de entrada sin tener que excluir ninguna y estimar de forma automatizada cuáles de estas características son importantes en la clasificación. Asimismo, este algoritmo tiene un método eficiente de estimar datos extraviados y mantiene la exactitud cuando existe un porcentaje de datos perdidos. Otra de las grandes ventajas de este clasificador, en este caso respecto a los árboles individuales es que evitan con mayor facilidad el sobreajuste en el entrenamiento.

3 Diseño y desarrollo

En esta sección se describirá el algoritmo utilizado para realizar las pruebas, así como las diferentes características de las señales que se han tenido en cuenta para los problemas de clasificación. Además, se explicarán en detalle los conjuntos de datos, al igual que de dónde provienen dichos datos.

3.1 Series temporales con dependencia de historia previa

La mayoría de las series temporales dependen de sus valores pasados. Los más recientes son buenos indicadores del comportamiento de una variable. Los valores anteriores de una señal, como una tasa de cambio, se registran sobre uno o más valores anteriores para predecir los valores actuales o futuros de una variable, por ejemplo, haciendo el promedio. Los datos que faltan, porque se hayan perdido o no se hayan registrado, a menudo se rellenan con datos pasados.

Luego se calculan las interrelaciones de los datos. Estas relaciones son desarrolladas en modelos donde son usadas para predecir puntos de tiempos futuros. Ocasionalmente, la suma ponderada de los valores presentes y pasados se usa para predecir los valores futuros. Cuando se trata de datos pasados para pronosticar valores futuros, es importante comprender que se utiliza un operador de retardo. Este operador permite que los modelos cuantifiquen cómo los valores pasados, presentes y futuros estén vinculados entre sí.

En este proyecto se clasificará usando el algoritmo *Random Forest*, pero considerando una codificación apta para preservar la estructura temporal particularmente en el caso de este algoritmo. En muchos sistemas de clasificación de series temporales se olvidan de la estructura temporal, que es lo que normalmente define a casi todas las series temporales.

3.2 SMTS

La clasificación de series temporales multivariadas (MTS por sus siglas en inglés) ha adquirido importancia en diferentes áreas. Este es un problema de aprendizaje supervisado en el cual, cada ejemplo consta de una o más series temporales.

Anteriormente se han dado estudios acerca de la clasificación de series temporales univariadas, donde un enfoque a este tipo de series sería K-NN, además de las aproximaciones obtenidas al obtener una representación rectangular de las MTS transformando el conjunto de la entrada de las series en un número fijo de columnas que usan diferentes enfoques de rectangularización (forma matricial) (Orsenigo & Vercellis, 2010).

Pero las MTS se caracterizan no solo por atributos individuales, sino también por las relaciones entre ellas. En esta sección se explicará el algoritmo SMTS (Symbolic representation for MTS).

Este algoritmo tiene en cuenta todos los atributos simultáneamente de las MTS, en vez de realizarlo por separado, para extraer la información de las relaciones de los atributos. Se utiliza una representación elemental que consiste en el índice de tiempo y los valores de las series temporales individuales como columnas. Otro reto importante en la clasificación de MTS es la alta dimensionalidad introducida por los múltiples atributos además de las series (Baydogan & Runger, 2015b).

3.2.1 Notación

Dada una MTS, X^n, es una serie temporal de M atributos diferentes cada uno de los cuales tiene T observaciones donde x_m^n es el m-ésimo atributo de la serie n y $x_m^n(t)$ denota la observación t.

La matriz X^n tiene dimensión T × M y tiene la siguiente forma: $X^n = [x_1^n, x_2^n, \ldots, x_M^n]$, donde $x_m^n = [x_m^n(1), x_m^n(2), \ldots, x_m^n(T)]'$ es la serie temporal en la columna *m*. Hay N MTS de entrenamiento asociadas a la clase $y^n \in \{0,1,2, \ldots, C\text{-}1\}$ correspondiente, para n desde 1 hasta N.

3.2.2 Algoritmo

A continuación se explicará de manera detallada el algoritmo SMTS (Baydogan & Runger, 2015b). Este algoritmo será la base para los resultados que se expondrán en las secciones posteriores.

3.2.2.1 Entrenamiento

En vez de extraer las características de cada serie temporal, cada fila de X^n es considerada para ser una instancia en la aproximación.

Crearemos la matriz *D* a partir la matriz *X*, está compuesta por el índice del tiempo, y las primeras diferencias de cada atributo numérico, por lo que cada fila de la matriz *D* por cada serie *n* en tiempo *t* es de la siguiente forma:

$$[t, x_1^n(t) , x_1^n(t) - x_1^n(t - 1), \ldots , x_M^n(t) , x_M^n(t) - x_M^n(t - 1)]$$

En la nueva implementación de este algoritmo propuesta en este trabajo, SMTS_extended, se añade un nuevo parámetro τ, utilizado para calcular las diferencias entre los valores, estas operaciones serían de la forma:

$$[t, x_1^n(t) , x_1^n(t) - x_1^n(t - \tau), \ldots , x_M^n(t) , x_M^n(t) - x_M^n(t - \tau)]$$

De esta forma, en cada serie temporal tendremos un τ óptimo por lo que podremos denotarlo como τ_X. Dentro de cada serie temporal, no todas las señales tienen la misma longitud ni variabilidad luego escogeremos τ en función de las señales con más cambios en su evolución temporal.

Por lo que veremos que, si dos puntos están muy cercanos en el tiempo y son muy similares, estas diferencias serán prácticamente 0, o, por el contrario, si se escoge un τ muy amplio y se perderán características temporales para la clasificación. Un valor óptimo preservará la estructura temporal necesaria para la clasificación.

Las diferencias en cada atributo aportan la inclinación en detalle de la serie, para que un árbol de decisión pueda obtener esta información relacionada con la clase. La diferencia no está disponible para la primera observación de la MTS, la cual se asume que está ausente, al igual que ocurriría en los atributos no numéricos, nominales, la primera diferencia sería 0 y no aportarían ninguna información.

En la tabla 3-1 presentada a continuación se expone un ejemplo de una serie univariante con clases binarias y con un índice de tiempo que está comprendido entre 1 y 3. Esta serie tiene un único atributo con sus primeras diferencias. Para representar las MTS, las columnas son generadas según el índice de tiempo, cada atributo, las primeras diferencias de cada atributo y de cada clase, es decir las columnas de la matriz D se irían multiplicando por 2 por cada característica añadida en el caso de que tuviese más atributos.

La tabla 3-1 sería un ejemplo de matriz D.

Ejemplo	Tiempo	Atributo	Diferencia	Clase
1	1	0.3	-	1
1	2	0.3	0	1
1	3	0.6	-0.3	0
2	1	0.5	-	0
2	2	0.2	0.3	1
2	3	0.9	-0.7	0
3	1	0	-	1
3	2	0.1	-0.1	0
3	3	0.3	-0.2	0

Tabla 3-1: Ejemplos de datos para una serie temporal univariada con clases.

Un árbol de decisión de *Random Forest* es entrenado sobre la matriz D asumiendo que cada observación tiene la misma clase que sus series temporales. Este algoritmo es conocido como *RFins* tiene J_{ins} árboles. Cada instancia de D es relacionada con un nodo terminal de cada árbol, g_j, j = 1, 2, . . ., J_{ins}**.** Aunque los árboles de *RFins* sin ninguna modificación no tienen una cota, se restringirá el número de nodos terminales de cada árbol a *R* y esto determinará el tamaño del alfabeto en nuestra aproximación.

Los árboles son entrenados mediante búsqueda en anchura, a continuación, se construye un nivel del árbol a la vez y se detiene el entrenamiento cuando hay R nodos terminales. Cada árbol de *RFins* aporta una representación simbólica de las series temporales.

Una manera simple de ver de qué manera cada punto de la serie temporal es clasificada en una u otra clase se reflejará en las siguientes figuras.

La primera es un gráfico que muestra las diferentes clases repartidas a lo largo del espacio 2-dimensional creado a partir de las observaciones y del tiempo, además de cómo cada observación es clasificada en una clase diferente al instante anterior. Esta clasificación depende de las características escogidas en el momento de la creación del árbol.

La segunda imagen es una representación esquemática de la creación de un árbol de decisión desde las series temporales realizando particiones de tiempo y de espacio, donde cada nodo terminal es una clase diferente.

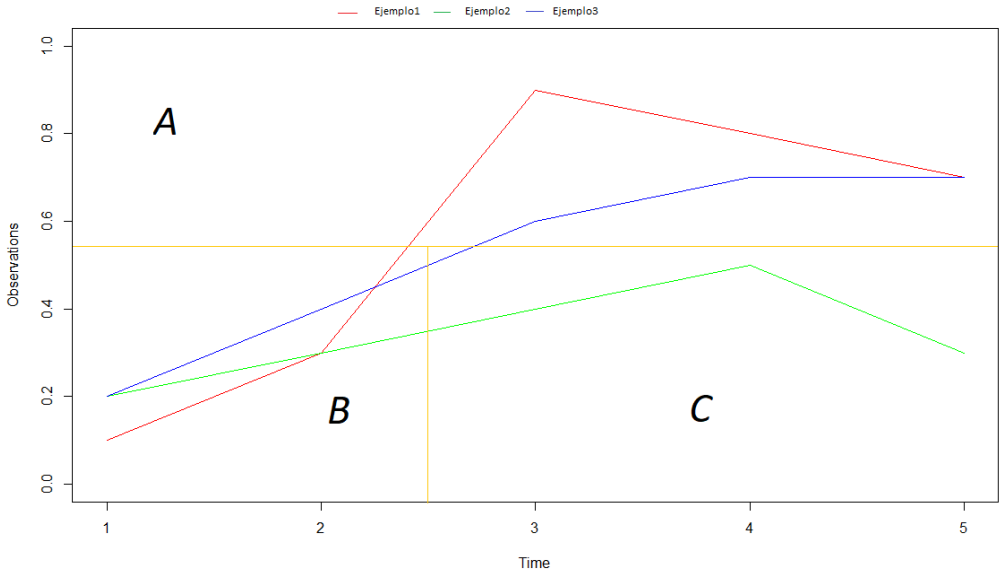

Figura 3-1: Serie temporal con 3 clases dibujada en el espacio.

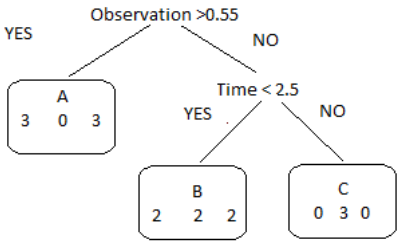

Figura 3-2: Árbol de decisión de la serie temporal con R=3.

3.2.2.2 Clasificación

La segunda fase de este algoritmo comienza con la representación simbólica generada de los árboles en el algoritmo *RFins*. Cada símbolo es considerado como una palabra y su vector de frecuencias relativas de los símbolos de cada árbol es concatenados y usados para clasificar las series temporales. Este vector frecuencias de cada árbol es normalizado por el número de instancias en las temporales para obtener el vector de frecuencias relativas. El vector de frecuencias relativas aparece en la imagen 3-2 en cada nodo terminal.

A continuación, se muestra un ejemplo visual de la representación basada en frecuencias simbólicas. El primer árbol coincidirá con el representado en la figura 3-2.

Árbol 1			Árbol 2			. . .	Árbol J_{ins}		
A1	B1	C1	A2	B2	C2	. . .	. . .	. . .	. . .
0,5	0,3	0	. . .	. . .	. . .	. . .	0,2	0,3	0,8
0	0,3	1	. . .	. . .	. . .	. . .	0,1	0,5	0,2
0,5	0,3	0	. . .	. . .	. . .	. . .	0,7	0,2	0

Tabla 3-2: Ejemplo visual de la representación de las frecuencias simbólicas normalizadas con J_{ins} árboles con R=3.

Cuando los vectores de las frecuencias normalizados ya están calculados llamaremos $H_j(X^n)$ al vector de cada nodo terminal de cada árbol g_j que contiene estos nuevos valores. Por lo que la matriz $H(X^n)$, *de dimensión* $R \times J_{ins}$, será la generada por todos los vectores.

Después se entrena un clasificador en la matriz $H(X^n)$. La cardinalidad de esta matriz puede ser grande según la configuración de los parámetros R y J_{ins}. Por lo tanto, para este inconveniente se prefiere un clasificador escalable que pueda manejar interacciones y correlaciones como *Random Forest.* Este RF se denomina RF_{ts} para el cual se entrenan J_{ts} árboles. Para clasificar una prueba de una MTS, se obtiene la matriz H y el RF_{ts} asigna la clase.

El algoritmo *Random Forest* es muy rápidos en la fase de ejecución, pero son lentos a la hora de entrenar. A priori, los *Random Forest* no serían muy buenos en el uso de la clasificación de las series temporales ya que este algoritmo mezcla todos los datos. Destruye todas las estructuras temporales.

3.3 Codificación de la estructura temporal

El procedimiento que se ha seguido para una propuesta en la mejora del porcentaje de acierto en la clasificación ha sido la modificación de la estructura temporal. Esta propuesta está basada en reconocer características de una señal, por lo que se va a elegir una codificación temporal en la cual quede reflejada la evolución temporal de la señal. En este proyecto esta codificación viene dada mediante los puntos de inflexión, máximos y mínimos relativos de cada señal.

Todas las series temporales que han sido probadas en este proyecto estaban grabadas en intervalos de uno en uno respecto a la serie original, donde no se ha especificado la unidad de tiempo de esos intervalos. Todos los intervalos tienen la misma duración. Esta mejora consiste en cambiar la codificación temporal en el algoritmo. Este nuevo algoritmo recibirá el nombre de SMTS_extended.

También se ha tenido en cuenta que, si el retraso es muy grande puede haber parte de la señal que se ha podido perder, por lo que el retardo no va a tener nunca más longitud que la duración de la serie temporal completa. Por tanto, a pesar de que el retardo sea amplio nunca se perderá la señal por completo.

Otra característica encontrada tenida en cuenta en la clasificación sería el tamaño del intervalo entre cada punto de la señal, es decir el distanciamiento entre diversos puntos de

la serie. Cuánto más distanciamiento tenga un punto de otro, en intervalos, más clara será la información que se obtiene de las señales. No obstante, separar mucho los puntos de la serie provocaría la perdida de la información en su variabilidad de la señal.

Se pueden considerar diferentes resoluciones a la hora de obtener resultados, el clasificador Random Forest baraja muchos de sus parámetros, entre otros el tiempo; pero con el algoritmo SMTS se evita que ocurra esto y sea perjudicial a la hora de clasificar series temporales.

El algoritmo propuesto SMTS es una adaptación del RF. Se propone este clasificador, RF, para mitigar la naturaleza codiciosa de los árboles univariados, se usa para dividir el espacio de características (Baydogan & Runger, 2014). Esto produce que la estructura temporal se respete y, por lo tanto, que la clasificación mejore.

3.4 Datasets

Inicialmente se realizaron las pruebas con las bases de datos halladas en las páginas *https://www.cs.ucr.edu/~eamonn/time_series_data/* , *http://archive.ics.uci.edu/ml/index.php* y *http://www.timeseriesclassification.com/*.

En estas bases de datos de las webs ya conocíamos los resultados de las tasas de error de la clasificación. Aquí debemos diferenciar dos grandes tipos de Series Temporales entre los distintos *Datasets*. El primer tipo, univariante, y multivariante el segundo.

3.4.1 Explicación detallada del dataset LIBRAS

Del segundo tipo vamos a explicar con mayor detalle el *dataset* LIBRAS. Este conjunto de datos contiene 15 clases de 24 instancias cada una, dónde cada clase hace referencia a un tipo de movimiento de la mano en LIBRAS (LÍngua BRAsileira de Sinais, lenguaje oficial de signos en Brasil). Cada instancia representa 45 puntos en un espacio bidimensional, donde se pueden representar de manera ordenada (de 1 a 45 en la coordenada X) para trazar el movimiento.

Se realiza un video de preprocesamiento, dónde se lleva a cabo una normalización de 45 fotogramas de cada video, de acuerdo con una distribución uniforme. En cada fotograma se encuentran los píxeles centrados de los objetos segmentados, en nuestro caso las manos, las cuales componen la versión discretizada de una curva con 45 puntos. Todas las curvas están normalizadas en el espacio unitario (Dias, Madeo, Rocha, Biscaro, & Peres, 2009).

Este conjunto de datos consta de 5 columnas:

- La primera indica el número de ejemplos que tiene el conjunto de datos, en este caso, esta serie tiene 180 tanto para el conjunto de prueba como para el de entrenamiento.
- La segunda columna muestra el índice de tiempo, el tamaño de la serie temporal, en este ejemplo 45.
- La tercera columna indica la clase, en este caso este *dataset* tiene 15 clases.
- Las dos últimas columnas indican el número de variables o atributos que posee la serie, en este caso al ser únicamente dos, nos indica que el espacio es

bidimensional donde la 4 columna representa los valores de las coordenadas en el eje de abscisas y la columna número 5 en el eje de ordenadas.

Por lo que tal y como se ha ejemplificado en la tabla 3-1 este conjunto de datos (tanto TRAIN como TEST) tendría un total de 180*45 = 8100 filas.

A continuación, se expondrán los distintos movimientos de cada clase de este conjunto de datos:

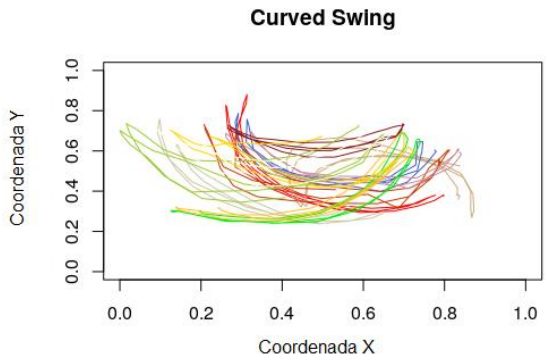

Figura 3-3: Representación Clase 1.

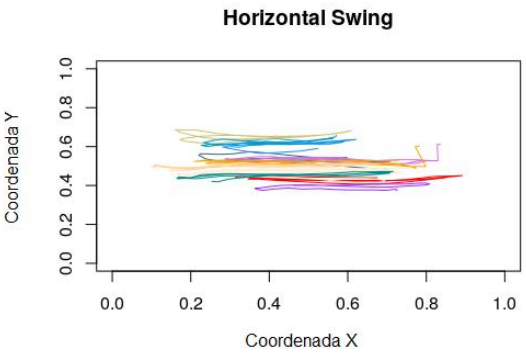

Figura 3-4: Representación Clase 2.

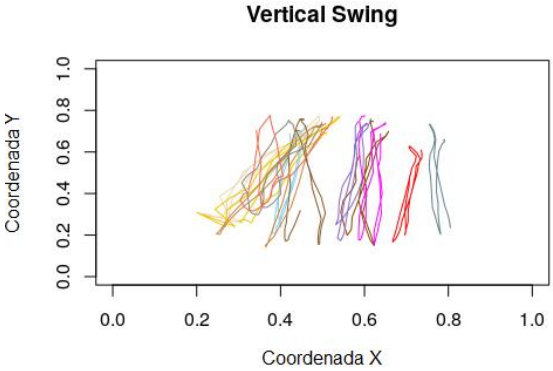

Figura 3-5: Representación Clase 3.

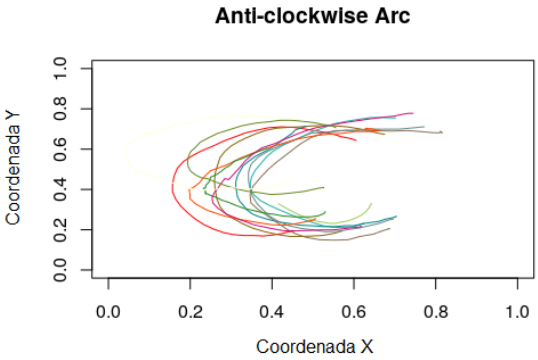

Figura 3-6: Representación Clase 4.

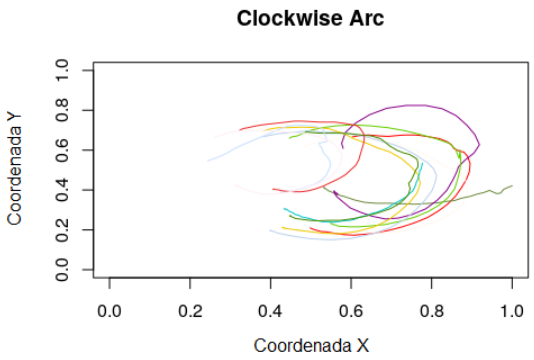

Figura 3-7: Representación Clase 5.

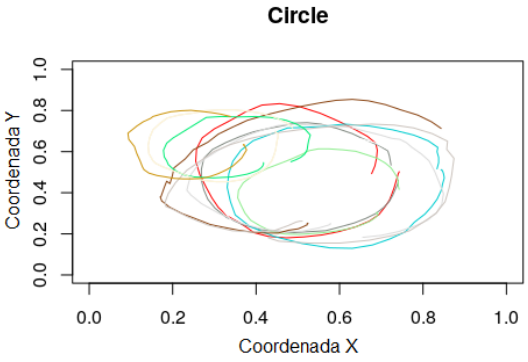

Figura 3-8: Representación Clase 6.

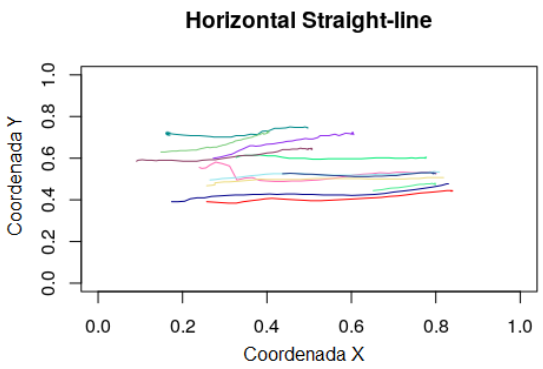

Figura 3-9: Representación Clase 7.

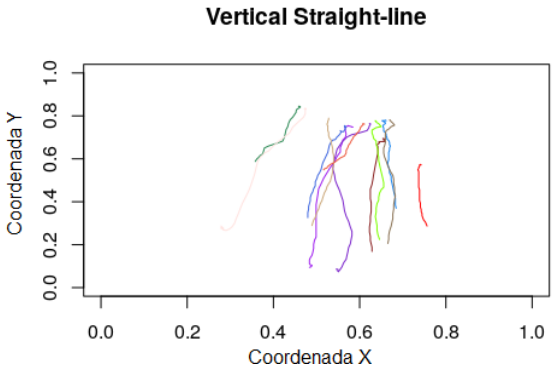

Figura 3-10: Representación Clase 8.

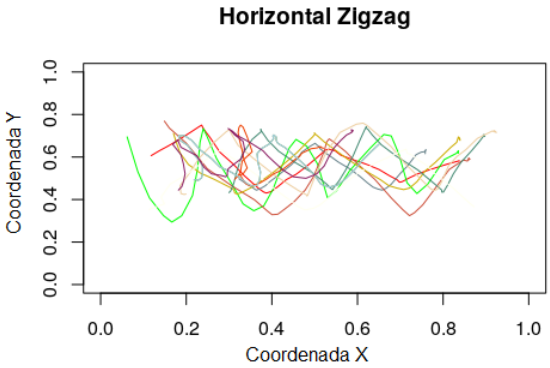

Figura 3-11: Representación Clase 9.

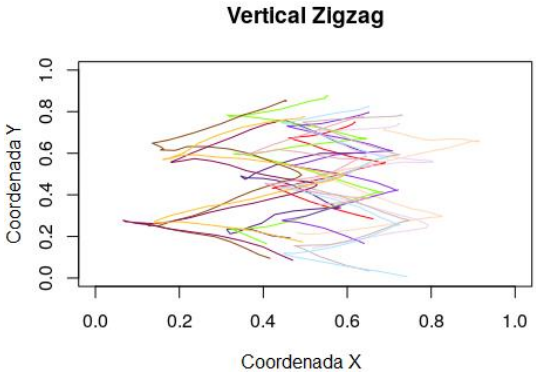

Figura 3-12: Representación Clase 10.

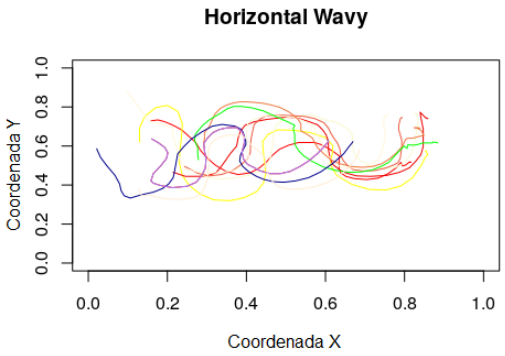

Figura 3-13: Representación Clase 11.

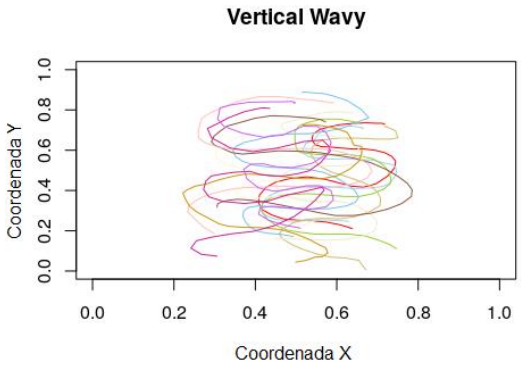

Figura 3-14: Representación Clase 12.

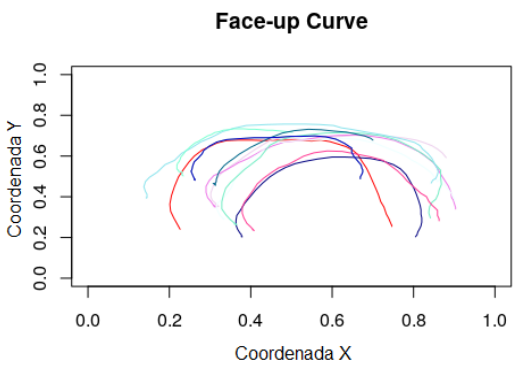

Figura 3-15: Representación Clase 13.

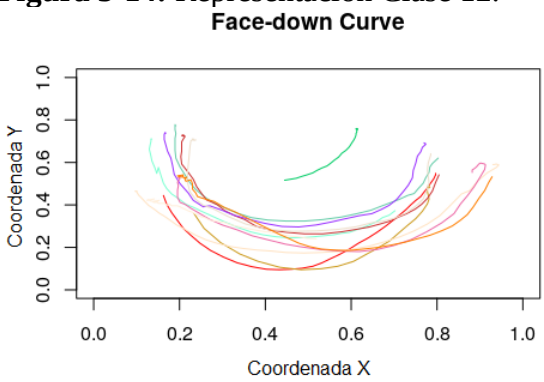

Figura 3-16: Representación Clase 14.

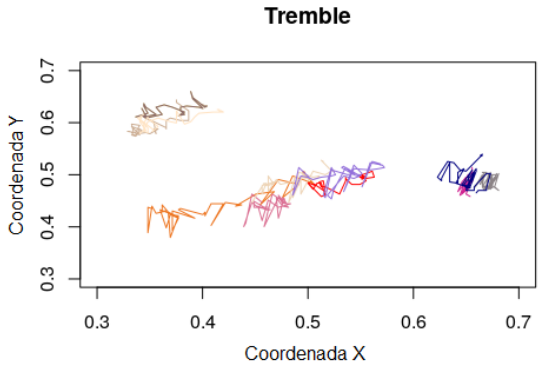

Figura 3-17: Representación Clase 15.

Dentro de estas, tenemos que encontrar cuáles tienen estructura temporal característica y cuáles no. Y una vez encontradas, habría que ver, en cada una de ellas, cual podría ser el retraso para tener en cuenta de cara a una mejora en la tasa de error en su clasificación.

Inicialmente se comprobará que con la implementación propuesta se consiguen los mismos resultados que los expedidos por Baydogan y Runger, una vez reproducido esto, se cambiará la estructura temporal para ver si esta propuesta mejora o empeora la tasa de error del algoritmo. En realidad, la estructura temporal depende de los propios datos que tiene la evolución temporal. Algo que evoluciona muy lentamente, y que seguramente mejorará el reconocimiento si codificamos con más intervalos de tiempo.

3.4.2 Explicación detallada del dataset GunPoint

Del primer tipo, series temporales univariadas, explicaremos el *dataset* GunPoint. Este conjunto de datos contiene dos clases y los 150 datos restantes de cada fila son valores de la serie temporal individual.

Este *dataset* involucra a un actor y una actriz, los cuales realizan un movimiento con su mano. Las dos clases son el dibujo de una pistola y apuntar: en ambos casos los actores tienen sus manos a los lados.

Para la primera clase los actores dibujan el movimiento de sacar una pistola de imitación de una funda, que se encuentra a la altura de sus caderas aproximadamente, durante un segundo, después devuelven la pistola a la funda y las manos de nuevo a los lados.

Para la clase punto, los actores tienen su arma a los lados. Señalan hacia un objetivo durante aproximadamente un segundo, y luego vuelven a dejar sus manos a los lados. Para ambas clases, se rastrea el centroide de las manos derechas del actor en los ejes X e Y, que parecen estar altamente correlacionados. Los datos del conjunto de datos son sólo del eje X (William Vickers., 2019).

Figura 3-18: Representación real del movimiento de la primera clase.

Figura 3-19: Representación real del movimiento de la segunda clase.

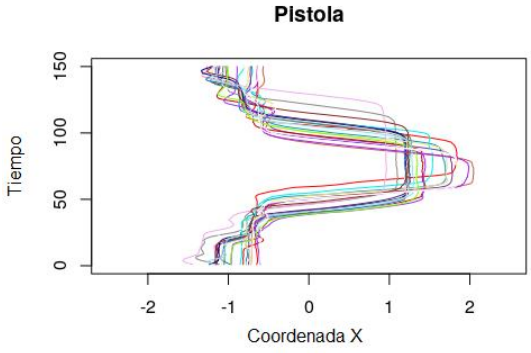

Figura 3-20: Representación clase 1

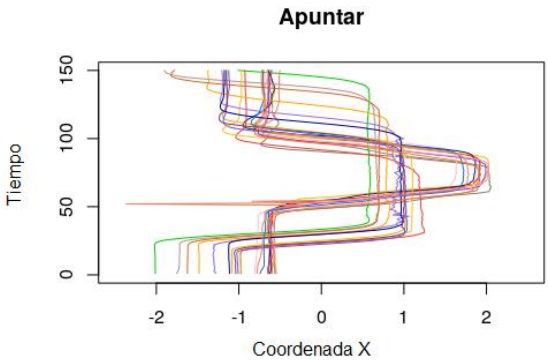

Figura 3-21: Representación clase 2

3.4.3 Explicación dataset ECG

Esta base de datos ha sido estudiada y publicada para el avance en la clasificación de arritmias en el latido cardiaco (Olszewski, 2001).
La base de datos ECG comprende una colección de conjuntos de datos de series temporales donde cada archivo contiene la secuencia de mediciones registradas por un electrodo durante un latido cardiaco. Cada latido tiene una clasificación asignada normal, 1, y anormal, 2.

El latido anormal es representativo de una patología cardiaca conocida como latido prematuro supraventricular (SVPB).

Los archivos de datos contenidos en esta base de datos se obtuvieron de la base de datos de arritmia (SVDM) disponibles en MIT-BIH y PhisioNet. Esta base datos contiene un subconjunto seleccionado al azar de los latidos cardiacos normales y todos los latidos anormales diagnosticados con SVPB.

ECG tiene un tamaño de 100 series temporales para el conjunto de entrenamiento y 100 para el de prueba.

3.4.4 Explicación dataset Middle Phalanx TW

Por último, se realizaron las pruebas con un *dataset* que no había sido probado antes con el algoritmo SMTS. Este conjunto de datos ha sido donado por L. Davis y A. Bagnall. Fue un problema que se creó en una colección de 11, como resultado de la tesis del primero, *"Predictive Modelling of Bone Ageing"* (Davis, 2013). Todas las imágenes han sido extraídas de *"Digital hand atlas and web-based bone age assessment: system design and implementation"* (Cao, Huang, Pietka, & Gilsanz, n.d.).

Estas bases de datos han sido diseñadas para probar la eficacia de la detección de contorno de manos y huesos y ver si existiría la posibilidad de predecir la edad ósea. Este trabajo ha cogido la base de datos usada para uno de los últimos problemas de clasificación de las imágenes óseas. Middle Phalanx TW supone predecir la puntuación Tanner-Whitehouse en las falanges medias.

Esta base de datos contiene series temporales multivariadas sacadas de 1300 imágenes. Cada serie temporal tiene una longitud de 80 atributos. Lo que se ha hecho para adaptarla al modelo del algoritmo SMTS es colocarlas en un eje de coordenadas, donde tenemos ahora dos atributos, eje X y eje Y, al igual que se ha hecho en (Baydogan & Runger, 2014) con la serie LIBRAS. El tamaño del conjunto de datos para entrenamiento es de 399 series temporales, que, contando los 40 instantes de tiempo, nos quedará una matriz D de 399*40 filas. Para el conjunto de prueba, tendremos 154*40 filas en la matriz.

Esta base de datos ha alcanzado un porcentaje de acierto en la clasificación de 58,69% con SVML (Máquinas de Vectores con Soporte Lineal) como algoritmo.

3.5 Vecinos próximos

Los algoritmos, que involucran al clasificador vecinos próximos, se usan, a menudo, en seres temporales univariadas. En este trabajo se han probado tanto series multivariadas como univariadas. Las primeras no solo se caracterizan por el número de atributos que tengan sino además por la relación entre ellos.

En este trabajo además de comparar las mejoras obtenidas respecto del porcentaje original, también se comparará con el clasificador KNN con distancias obtenidas del algoritmo DTW (Dynamic Time Warping), debido a que en el artículo original de Baydogan y Runger aparecen estas comparaciones. Este algoritmo se utiliza en el reconocimiento de similitudes en varias señales.

3.6 Ejecución

El medio para la ejecución de las pruebas ha sido en todo momento el clúster de la Universidad Autónoma. Cada prueba se ejecutaba en un nodo aleatorio, por lo que la velocidad de cada prueba depende del nodo en el que se ejecuten ya que no todos tenían la misma velocidad.

Además del medio de ejecución, un factor importante a la hora de lanzar las pruebas será que el algoritmo SMTS está paralelizado, por lo que de esta manera como varias operaciones se realizarán simultáneamente, el tiempo también disminuirá.

4 Pruebas y resultados

En esta sección se mostrarán los resultados obtenidos gracias a la clasificación usando *Random Forest* mediante el algoritmo STMS, explicado en la sección anterior. Con estos resultados se comparará el acierto de este algoritmo con la mejora en la estructura temporal propuesta, SMTS_extended, así como las pruebas realizadas con otros clasificadores.

El porcentaje de acierto de este algoritmo se ha extraído realizando 10 reproducciones de la prueba y calculando su media de estos 10 resultados. En cada prueba lanzada se ejecutan 10 repeticiones, por lo que se obtienen 100 resultados diferentes. Cada conjunto de datos contiene un fichero para entrenar y otro para clasificar.

Para el clasificador Random Forest se van a listar a continuación una serie de parámetros que se han tenido en cuenta a la hora de realizar las pruebas:

- Noftreelevels: Número de árboles en la generación simbólica (Jins) explicada en la sección 3.2.2. Este valor será un número aleatorio entre 20, 50 o 100.
- Nofnodelevels: tamaño del alfabeto, es decir, el número de nodos terminales (R). Este valor será un número aleatorio entre 20, 50 o 100.
- Maxiter: Número máximo de iteraciones en los árboles de entrenamiento. Este valor será 20.
- Noftree_step: Número de árboles para entrenamiento por iteración. Este valor será 50.
- Tolerance: Valor establecido en la comparación del error relativo como forma posible de parada. Este valor será 0,05.

Los conjuntos de datos en los que se han basado las pruebas han sido sacados de amplios campos. En este trabajo los *datasets* serán en su mayoría del campo de la medicina. Pero esta propuesta se puede aplicar en la economía, en movimientos geológicos, deportes y una larga lista de temas diferentes.

CONJUNTO DE DATOS	NÚMERO DE CLASES	NÚMERO DE VARIABLES	LONGITUD	TAMAÑO DE TRAIN	TAMAÑO DE TEST
GUNPOINT	2	1	150	50	150
LIBRAS	15	2	45	180	180
ECG	2	2	39-152	298	896
MIDDLE PHALANX-TW	6	2	40	399	267

Tabla 4-1: Explicación esquemática de cada conjunto de datos.

En una primera aproximación se empezaron a realizar los retardos de las pruebas con valores muy conservadores y con conjuntos de datos que tenían una dependencia de sus datos anteriores y con conjuntos de datos que no tenían. A partir de este momento, y teniendo en cuenta la mejora propuesta, se verá como esto solo mejora *Datasets* con dependencia a valores anteriores y multivariados.

Todas las bases de datos tienen como medida de tiempo intervalos de uno en uno, no se especifica si son en segundos o minutos.

4.1 Dataset GunPoint

Como prueba en una serie univariada, se tomará el conjunto de datos GunPoint, explicado en la sección 3.3,

A continuación, veremos en la siguiente imagen los diferentes intervalos del estudio realizado para las diferentes clases de esta serie. El primer retardo se relacionaría con el primer y el segundo punto de inflexión apreciable en la primera clase, 20 unidades de tiempo, el segundo retardo, el segundo y el más apreciable sería de 65 unidades de tiempo y, por último, 30. Para la segunda gráfica tendremos solamente dos puntos de inflexión, luego el único retardo a destacar sería 60.

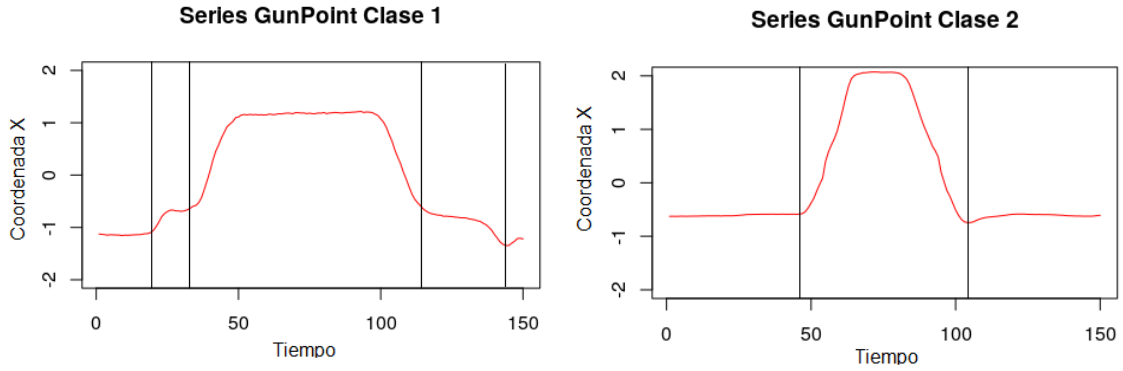

Figura 4-1: Representación de ambas clases con los diferentes intervalos apreciables.

A continuación, se muestra la tabla con los diferentes valores acierto obtenidos, la primera fila se mostraría sería el resultado devuelto por el artículo *Learning a symbolic representation for multivariate time series classification* (Baydogan & Runger, 2015), SMTS, y el resultado de ejecutarlo en el clúster de la UAM en la segunda fila sería el resultado de ejecutarlo en el clúster de la UAM, llamado SMTS_local.

MÉTODO	PORCENTAJE DE ACIERTO
SMTS	98,90%
SMTS_local	98,20%

Tabla 4-2: Ratios de error del conjunto de datos GunPoint sin mejoras.

La tabla que se muestra a continuación es el resultado de realizar las pruebas con las modificaciones en la codificación temporal del algoritmo SMTS (SMTS_extended), especificado anteriormente.

RETARDO	POCENTAJE DE ACIERTO
20	99,50%
30	99,30%
60	99,40%
65	99,50%

Tabla 4-3: Ratios de acierto del conjunto de datos GunPoint con STMS_extended.

4.2 Dataset Libras

Este conjunto de datos, tal y como ha sido explicado anteriormente, está compuesto por 15 clases y representa el movimiento de las manos de la lengua de signos brasileña.

En la imagen 4-2 que se observa a continuación se muestra la primera serie de las 15 clases con sus dos atributos, en esta serie temporal podemos destacar que la clase con más variación en su movimiento en ambos atributos es la clase 1. Como se puede observar en las figuras 4-3 y 4-4. Por lo que para definir su retraso tomaremos estas clases para hallar sus puntos de inflexión y extremos relativos. La clase 1 en la parte más inferior del gráfico.

En la figura 4-2 se muestra el atributo de las coordenadas X en la parte inferior del gráfico y en la superior el eje Y de la señal. El eje Y está representado con un offset.

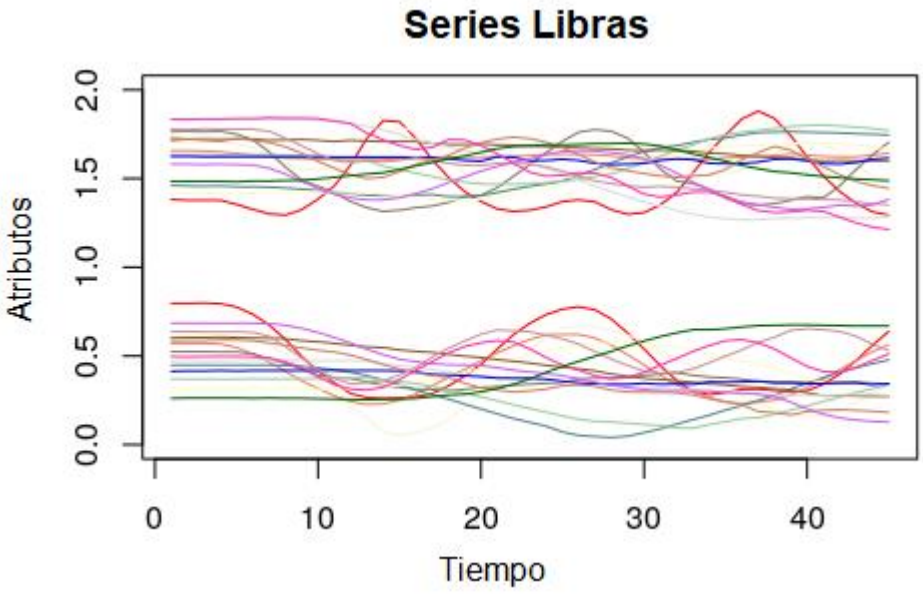

Figura 4-2: Representación de las clases del dataset libras.

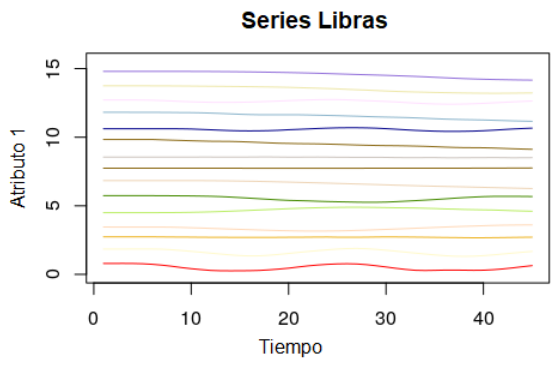

Figura 4-3: Representación atributo X.

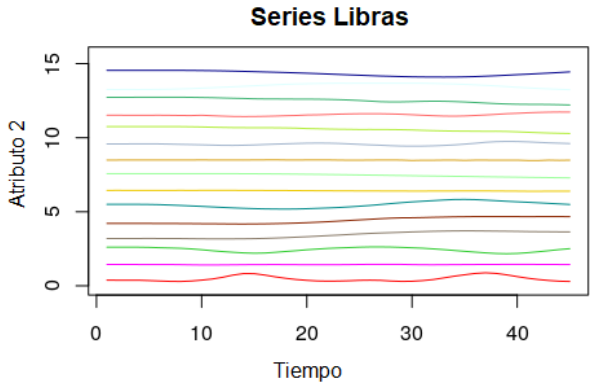

Figura 4-4: Representación atributo Y.

La tabla 4-4 muestra los resultados del algoritmo sin realizar ninguna mejora realizando la comparación con KNN con DTW (NNDTW) (Baydogan & Runger, 2015). Esta comparación con otro clasificador aparece en el artículo original.

MÉTODO	PORCENTAJE DE ACIERTO
SMTS	90,90%
SMTS_local	89,30%
NNDTW	80%

Tabla 4-4: Porcentajes de acierto para el dataset de LIBRAS sin mejoras.

Tal y como se explicó en la sección previa, tanto los retardos como la ampliación en los intervalos viene según la representación de las señales. En concreto en esta base de datos nos fijaremos en la Clase 1, debido a que tiene una mayor variación en su señal. En la figura 4-4 se muestra la clase 1 con sus separaciones.

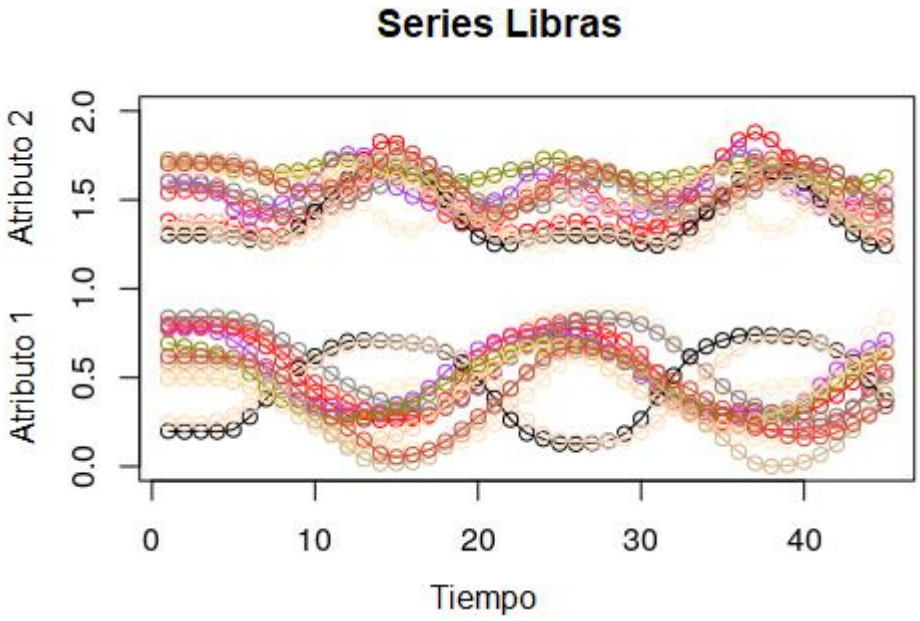

Figura 4-5: Representación de la clase 1 del dataset LIBRAS.

A continuación, se muestran los porcentajes resultantes de la ejecución del algoritmo con las mejoras propuestas. Como vemos, aunque la mejora no es muy apreciable, conseguimos que en 20 (intervalo entre máximos) sea el retraso óptimo.

RETRASO	PORCENTAJE DE ACIERTO
1	90,60%
7	90,90%
8	90,60%
10	90,40%
20	91,40%
25	90,90%

Tabla 4-5: Porcentajes de acierto para el dataset de LIBRAS con SMTS_extended.

En la tabla 4-5, se muestran las mejoras en el acierto con la variación en los intervalos de tiempo para la codificación temporal óptima (20).

TAMAÑO DE LOS INTERVALOS	PORCENTAJE DE ACIERTO
2	91,60%
5	91,20%
10	91,30%
20	91,20%

Tabla 4-6: Porcentajes de acierto para el dataset de LIBRAS variando los intervalos en T=-20.

4.3 Dataset ECG

Este conjunto de datos, tal y como ha sido presentado anteriormente, está compuesto por dos atributos y dos clases, diferenciando el latido normal del anormal en la recogida de los latidos de corazón mediante un electrodo.

En la tabla que se muestra a continuación, aparecen los porcentajes de acierto de la clasificación del dataset que originalmente aparece en artículo *Learning a symbolic representation for multivariate time series classification* (Baydogan & Runger, 2015), seguido del devuelto por el clúster, ambos con Random Forest, además de la comparativa con el clasificador vecinos próximos como aparece en el artículo original.

MÉTODO	PORCENTAJE DE ACIERTO
SMTS	81,80%
SMTS_local	82,20%
NNDTW	85%

Tabla 4-7: Porcentajes de acierto para el dataset de ECG.

Las series de esta base de datos tienen longitudes diferentes, por lo que en la figura 4-6 que se mostrará a continuación, aparecerán varias series de ambas de la clase 1 y en la figura 4-7 las primeras series de la clase 2.

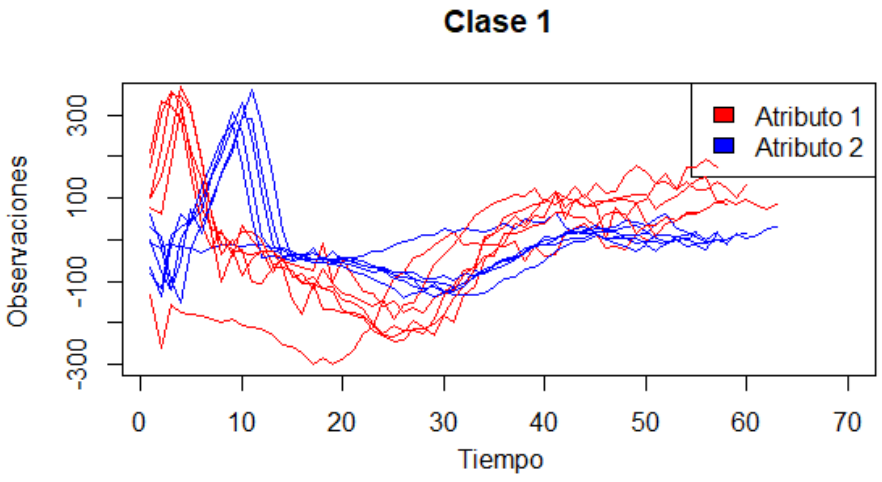

Figura 4-6: Representación de ambos atributos de la clase 1.

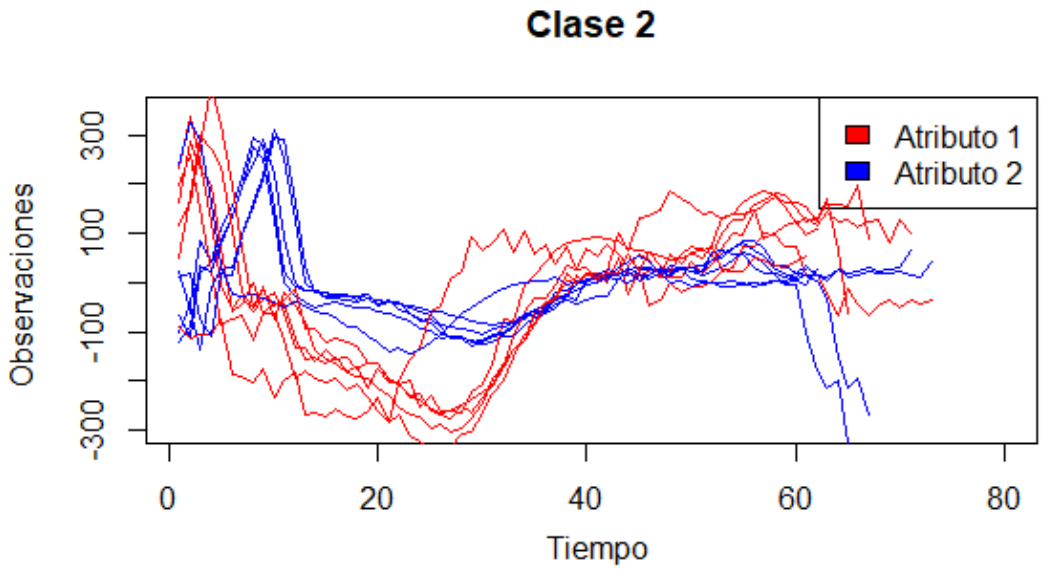

Figura 4-7: Representación de ambos atributos de la clase 2.

Este problema tiene más dificultad para encontrar los retrasos óptimos debido al movimiento tan irregular de las señales. Debido a esto, se han suavizado los picos menos significativos con el fin de encontrar los extremos más relevantes. Con esto obtenemos la siguiente tabla.

RETRASO	PORCENTAJE DE ACIERTO
15	82,80%
30	82,20%
35	83,70%

Tabla 4-8: Porcentajes de acierto para el dataset de ECG con SMTS_extended.

En las tablas 4-9, 4-10 y 4-11 se muestran los porcentajes teniendo en cuenta ambas características, el retraso y la separación en los intervalos.

TAMAÑO DE LOS INTERVALOS	PORCENTAJE DE ACIERTO
2	84,00%
10	83,00%
30	81,30%

Tabla 4-9: Porcentajes de acierto para el dataset de LIBRAS variando los intervalos en T=-15.

En esta tabla, se muestra como al ser los intervalos más amplios que el retraso la pérdida de la señal es evidente por lo que el porcentaje de acierto baja.

TAMAÑO DE LOS INTERVALOS	PORCENTAJE DE ACIERTO
2	82,00%
10	81,80%
30	82,70%

Tabla 4-10: Porcentajes de acierto para el dataset de LIBRAS variando los intervalos en T=-30.

TAMAÑO DE LOS INTERVALOS	PORCENTAJE DE ACIERTO
2	83,20%
10	83,00%
30	82,80%

Tabla 4-11: Porcentajes de acierto para el dataset de LIBRAS variando los intervalos en T=-35.

4.4 Dataset Middle Phalanx TW

Como se ha especificado antes, el dataset Middle Phalanx TW mide la eficacia en la detección de contorno de manos y huesos y ver si existiría la posibilidad de predecir la edad ósea.

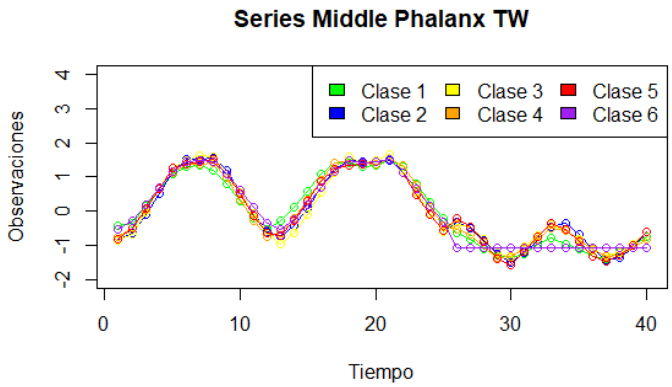

Figura 4-8: Representación de cada clase del dataset Middle Phalanx TW.

En la tabla 4-12 se mostrarán los porcentajes de acierto, tanto del algoritmo SVML, como del SMTS ejecutado en el clúster de la UAM.

MÉTODO	PORCENTAJE DE ACIERTO
SVML	58,69%
SMTS_local	54,20%

Tabla 4-12: Porcentajes de acierto en el dataset Middle Phalanx TW sin mejoras.

Para calcular los retardos, cogemos la Clase 1, la señal más variable y tomamos sus extremos relativos.

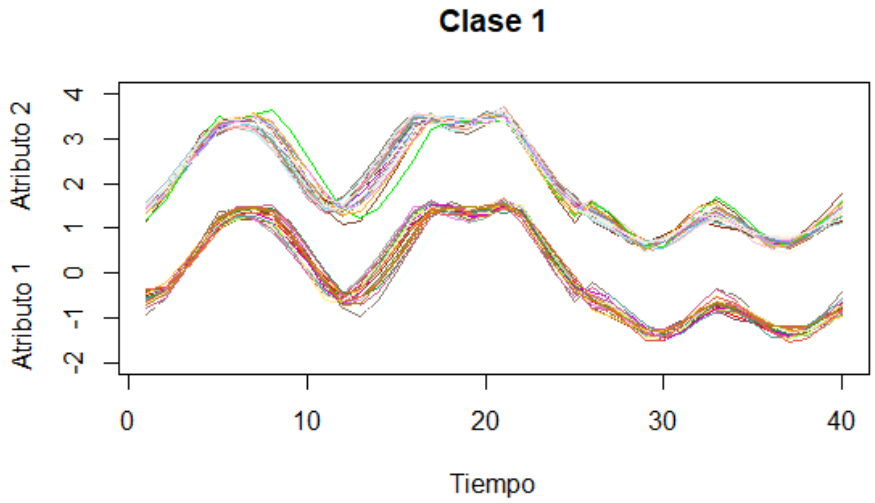

Figura 4-9: Representación de la clase 1 del dataset Middle Phalanx TW.

En la tabla que se muestra a continuación, se han calculado los porcentajes de acierto en función del retardo calculado mediante los extremos relativos y puntos de inflexión de las clases.

RETARDO	PORCENTAJE DE ACIERTO
12	55,10%
25	56,20%
15	55,50%

Tabla 4-13: Porcentajes de acierto en el dataset Middle Phalanx TW con SMTS_extended.

Al igual que en los anteriores conjuntos de datos, también se han realizado pruebas variando la longitud de los intervalos con el retardo óptimo.

TAMAÑO DE LOS INTERVALOS	PORCENTAJE DE ACIERTO
3	56,20%
5	59,70%
10	55,80%
15	59,70%
18	57,80%
25	59,70%

Tabla 4-14: Porcentajes de acierto en el dataset Middle Phalanx TW variando los intervalos en T=-25.

4.5 Tiempos de ejecución

Como se mencionó en la sección 3.6, todas las pruebas fueron ejecutadas en el clúster, cada prueba se realizaba en un nodo aleatorio lo que cada prueba termina en un tiempo diferente a las demás. En estos problemas el tiempo no es tan importante ya que no exceden mucho de tiempo.

A continuación, se muestra una tabla con la diferencia máxima entre los tiempos de cada prueba en cada conjunto de datos.

CONJUNTO DE DATOS	TIEMPO MÁXIMO	TIEMPO MÍNIMO	DIFERENCIA
GUNPOINT	2,12 s	1,79 s	0,33 s
LIBRAS	11,17 s	8,85 s	2,32 s
ECG	4,92 s	3,87 s	1,13 s
MIDDLE PHALANX TW	26,61 s	21,04 s	5,57 s

Tabla 4-15: Tiempos de ejecución de las pruebas.

5 Conclusiones y trabajo futuro

En esta sección se describirán las conclusiones de trabajo y el posible futuro trabajo que se puede realizar a partir de este proyecto.

5.1 Conclusiones

El clasificador Random Forest es, a pesar de mezclar sus entradas y por ello de la destrucción de la temporalidad de las series temporales, eficiente con el algoritmo SMTS ya que atenúa la destrucción de las relaciones temporales. En este contexto, se ha propuesto el SMTS_extended, una codificación que tiene en cuenta la resolución temporal de las series. En este proyecto se ha comprobado que, en efecto, SMTS_extended es adecuado para clasificar series temporales puesto que optimiza el rendimiento como se ha mostrado en el análisis realizado.

En el artículo *Learning a symbolic representation for multivariate time series classification* (Baydogan & Runger, 2015) aparecen conjuntos de datos con porcentajes muy elevados del acierto en la clasificación, por lo que las mejoras presentadas en esos conjuntos de datos no han sido muy elevadas, 2% o 3%. Por otro lado, en el caso del conjunto Middle Phalanx TW, se partía de un porcentaje menor de acierto en la clasificación, y, se ha conseguido una mejora de un 6% calculando la media de las pruebas realizadas con la codificación temporal, por lo que en algunas ejecuciones se llegaba hasta una mejora de un 10%.

Los clasificadores paramétricos en *machine learning* admiten una optimización en sus parámetros de entrada. Como consecuencia, el algoritmo SMTS admite mejoras mediante la optimización de la codificación temporal que proporcione al clasificador la mejor información sobre la estructura temporal de las series temporales que se deseen clasificar.

En el caso de la serie temporal ECG, las series estudiadas no tienen la misma duración, por lo que se han realizado las pruebas con las señales más representativas. Además, este conjunto de datos tenía mucha variabilidad en las señales, por lo que se ha optado por una suavización de los datos mediante la función smooth.spline. El objetivo de esta función es minimizar la función de error, que va variando según la distancia a la que se encuentren los puntos y la varianza entre ellos. Es decir, los datos se han filtrado para estimar mejor los valores de τ.

Como se ha visto, para cada serie temporal se ha utilizado un τ para cada serie temporal dependiendo de la variabilidad de la señal. A partir de aquí, se escoge el τ óptimo en la clasificación para cada serie temporal. Como discutió en la sección anterior, este parámetro se escoge en función de la clase que tenga más variabilidad. Un ejemplo en este contexto sería una serie temporal medioambiental medida mediante sensores que recogen, por ejemplo, parámetros como la temperatura, humedad y nivel de CO_2. Los dos primeros no varían mucho en unos pocos segundos, pero el tercero sí puede presentar variaciones significativas de sus niveles en instantes de tiempo más breves, luego el τ óptimo sería el que modelice mejor la señal representada por el CO_2. La elección de τ puede considerar la estructura de la variabilidad de las series temporales, así como su duración.

5.2 Trabajo futuro

Este trabajo ha estudiado la clasificación de las series temporales con una versión del clasificador Random Forest llamada SMTS_extended, ya que previamente se conocía que por los estudios realizados por Baydogan y Runger, que el algoritmo STMS ofrecía un mejor porcentaje de acierto respecto al clasificador KNN en distintos ejemplos de series temporales. Como cualquier clasificador paramétrico, el porcentaje resultante de acierto se puede optimizar en función de sus parámetros.

Una de las mejoras que se pueden llegar a desarrollar en un futuro sería la elección automática de los valores de τ, dependiendo de la temporalidad de cada una de las series que componen el conjunto multivariado, atendiendo a la caracterización de su duración y estructura temporal.

Finalmente se pueden introducir nuevas propuestas en el algoritmo SMTS, más complejas que las detalladas en este proyecto, orientadas a representar mejor la estructura temporal de las series multivariadas en su conjunto, debido a que el objetivo es reducir la tasa de error de los clasificadores cualquier mejora de la codificación de la temporalidad puede dar lugar a mejores resultados. En este trabajo se han analizado series temporales relacionadas, en su mayoría, con la medicina; pero este método se puede llevar a cabo en una amplia gama de tipos de datos como los procedentes del sector energético o sensores ambientales. Otro de los trabajos posibles sería el estudio de este problema de codificación temporal con otros clasificadores como redes neuronales y analizar la posibilidad de la disminución del error de clasificación con una estrategia parecida a la llevada a cabo en este estudio.

Referencias

Abanda, A., Mori, U., & Lozano, J. A. (2019a). A review on distance based time series classification. *Data Mining and Knowledge Discovery*, *33*(2), 378–412. https://doi.org/10.1007/s10618-018-0596-4

Abanda, A., Mori, U., & Lozano, J. A. (2019b). A review on distance based time series classification. *Data Mining and Knowledge Discovery*, *33*(2), 378–412. https://doi.org/10.1007/s10618-018-0596-4

Bagnall, A., & Janacek, G. (2014). A Run Length Transformation for Discriminating Between Auto Regressive Time Series. *Journal of Classification*, *31*(2), 154–178. https://doi.org/10.1007/s00357-013-9135-6

Baydogan, M. G., & Runger, G. (2014). Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, *29*(2), 400–422. https://doi.org/10.1007/s10618-014-0349-y

Baydogan, M. G., & Runger, G. (2015a). Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, *29*(2), 400–422. https://doi.org/10.1007/s10618-014-0349-y

Baydogan, M. G., & Runger, G. (2015b). Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, *29*(2), 400–422. https://doi.org/10.1007/s10618-014-0349-y

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2008). Random Forests - Original Paper. *Vasa*, 1–33. Retrieved from http://oz.berkeley.edu/~breiman/randomforest2001.pdf%0Ahttp://medcontent.metapress.com/index/A65RM03P4874243N.pdf

Brownlee, J. (2016). Bagging and Random Forest Ensemble Algorithms for Machine Learning. Retrieved May 20, 2019, from https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/

Cao, F., Huang, H. K., Pietka, E., & Gilsanz, V. (n.d.). Digital hand atlas and web-based bone age assessment: system design and implementation. *Computerized Medical Imaging and Graphics : The Official Journal of the Computerized Medical Imaging Society*, *24*(5), 297–307. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/10940607

Cohen, W., & Cohen, W. (1993). Efficient Pruning Methods for Separate-and-Conquer Rule Learning Systems. *13th International Joint Conference on Artificial Intelligence*, 988–994. Retrieved from http://www.cs.cmu.edu/~wcohen/postscript/ijcai-93.ps

Cutler, A., & Cutler, D. R. (2012). *Ensemble Machine Learning*. (January). https://doi.org/10.1007/978-1-4419-9326-7

Davis, L. M. (2013). *Predictive Modelling of Bone Ageing*. Retrieved from

https://ueaeprints.uea.ac.uk/45085/

Dias, D. B., Madeo, R. C. B., Rocha, T., Biscaro, H. H., & Peres, S. M. (2009). Hand movement recognition for Brazilian Sign Language: A study using distance-based neural networks. *2009 International Joint Conference on Neural Networks*, 697–704. https://doi.org/10.1109/IJCNN.2009.5178917

Fulcher, B. D., & Jones, N. S. (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, *26*(12), 3026–3037. https://doi.org/10.1109/TKDE.2014.2316504

Hills, J., Lines, J., Baranauskas, E., Mapp, J., & Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, *28*(4), 851–881. https://doi.org/10.1007/s10618-013-0322-1

Kotsiantis, S. (2014). Bagging and boosting variants for handling classifications problems: a survey. *Knowledge Eng. Review*, *29(1)*, 78–100.

López-Iñesta, E., Grimaldo, F., & Arevalillo-Herráez, M. (2015). Classification similarity learning using feature-based and distance-based representations: A comparative study. *Applied Artificial Intelligence*, *29*(5), 445–458. https://doi.org/10.1080/08839514.2015.1026658

Olszewski, R. T. (2001). *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*. Retrieved from https://www.cs.cmu.edu/~bobski/pubs/tr01108-twosided.pdf

Orsenigo, C., & Vercellis, C. (2010). Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, *43*(11), 3787–3794. https://doi.org/10.1016/j.patcog.2010.06.005

William Vickers. (2019). Time Series Classification. Retrieved June 1, 2019, from http://www.timeseriesclassification.com/description.php?Dataset=GunPoint

Xing, Z., Pei, J., & Keogh, E. (2010a). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, *12*(1), 40. https://doi.org/10.1145/1882471.1882478

Xing, Z., Pei, J., & Keogh, E. (2010b). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, *12*(1), 40. https://doi.org/10.1145/1882471.1882478

Glosario

Bagging	Algoritmo de aprendizaje automático diseñado para mejorar la estabilidad y precisión de algoritmos
ECG	Conjunto de datos que recoge arritmias en el latido cardiaco
MTS	Serie temporal multivariada
KNN	K Vecinos próximos
RF	Random Forest
SMTS	Algoritmo de clasificación simbólica de series temporales multivariadas propuesto por Baydogan and Runger.
SMTS_extended	Algoritmo de clasificación simbólica de series temporales multivariadas con codificación temporal.
SMTS_local	Algoritmo de clasificación simbólica de series temporales multivariadas ejecutado en local.
TW	Puntuación Tanner-Whitehouse