

UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



**Clasificación semi-supervisada de series  
temporales**

MEMORIA QUE PRESENTA

**Mabel González Castellanos**

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Programa Oficial de Doctorado en Tecnologías  
de la Información y la Comunicación

DIRECTOR:

**José Manuel Benítez Sánchez**

Editor: Universidad de Granada. Tesis Doctorales

Autor: Mabel González Castellanos

ISBN: 978-84-9163-043-3

URI: <http://hdl.handle.net/10481/44303>

La doctoranda Mabel González Castellanos y el director de la tesis Dr. José Manuel Benítez Sánchez garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por la doctoranda bajo la dirección del director de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, octubre de 2016

La Doctoranda

El Director

Fdo: Mabel González Castellanos

Fdo: José Manuel Benítez Sánchez



# Agradecimientos

Agradezco a mi director de tesis, José Manuel, por todo su apoyo a lo largo de estos años. A los compañeros que colaboraron en esta investigación, Christoph e Isaac, por todos los consejos y conocimientos que han compartido conmigo. Les agradezco por haber dedicado tanto tiempo a las innumerables revisiones que tanto han aportado a mi formación.

A los miembros del Departamento de Computación de la Universidad Central de las Villas que contribuyeron en mi formación profesional. En especial a Yanet, y a todos los miembros del Laboratorio de Inteligencia Artificial.

Muchas gracias a los miembros del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, en especial al Curro y a todos los profesores que han formado parte en los tribunales donde se han evaluado nuestros avances en la investigación.

Mi agradecimiento a las instituciones y proyectos que han posibilitado la realización de esta tesis. En específico al “Proyecto de Investigación de Excelencia de la Junta de Andalucía, P12-TIC-2958” y al “Proyecto de Investigación del Ministerio de Economía y Competitividad, TIN2013-47210-P”. Un agradecimiento especial a la “Asociación Universitaria Iberoamericana de Postgrado” (AUIP) por auspiciar el programa de doctorado, así como las estancias de investigación realizadas en la Universidad de Granada.

A todos los amigos tanto de Cuba como de Granada por hacerme más grato el camino. A mi familia por darme el valor para llegar a la meta.

iii MUCHAS GRACIAS !!!



# Resumen

La presente investigación se centra en el estudio, diseño y evaluación de métodos semi-supervisados de auto-etiquetado para abordar la clasificación de series temporales, considerando las características distintivas implicadas en el tratamiento de este tipo de dato. En ese sentido los objetivos planteados fueron:

- Efectuar un estudio comparativo experimental entre varios métodos de auto-etiquetado de probada efectividad en aras de identificar cuáles ofrecen mejores resultados en la clasificación de series temporales.
- Desarrollar un estudio sobre la influencia que ejerce la complejidad de los problemas temporales, desde el punto de vista de la clasificación, en el desempeño de los métodos de auto-etiquetado.
- Proponer y validar criterios de parada para el algoritmo Self-training, orientados al aprendizaje semi-supervisado de una clase, que resulten más efectivos en comparación con los reportados en la literatura.
- Implementar un paquete de código abierto en lenguaje R para la aplicación de métodos de auto-etiquetado en la solución de problemas de clasificación semi-supervisada.

La primera contribución de la tesis consiste en un estudio comparativo entre varias técnicas de auto-etiquetado donde se exploraron diversos clasificadores base adaptados específicamente al tratamiento de series temporales. Los resultados obtenidos permitieron alcanzar varias conclusiones sobre la aplicabilidad de este tipo de técnicas al dominio temporal. Respecto al desempeño de los clasificadores base, se ratificó el 1NN como una opción robusta en el contexto semi-supervisado por ofrecer las clasificaciones de mayor exactitud. Los

restantes clasificadores base evaluados presentan resultados competitivos cuando se combinan sus hipótesis siguiendo un esquema de multiaprendizaje. Respecto a los métodos semi-supervisados evaluados, se destacaron particularmente tres métodos de auto-etiquetado: SETRED, Tri-training y Democratic. Estos métodos presentaron resultados significativamente superiores al resto en términos de sus capacidades tanto inductivas como transductivas. El enfoque multclasificador, al que pertenecen los métodos Tri-training y Democratic, se destacó como una solución prometedora al problema de la clasificación semi-supervisada en el contexto de las series temporales.

La segunda contribución de la tesis consiste en un estudio experimental acerca de la relación entre la complejidad asociada a los problemas de clasificación de datos temporales y los resultados obtenidos con las técnicas de auto-etiquetado estudiadas. Como parte de dicho estudio se evaluaron varias medidas de complejidad, las cuales resultaron informativas sobre las características de los conjuntos de datos temporales incluso para conjuntos pequeños de ejemplos etiquetados. Como resultado de este estudio se constató una fuerte correlación entre la complejidad estimada y la exactitud obtenida mediante los métodos semi-supervisados evaluados. De esta forma se identificaron las medidas de complejidad más relevantes y en correspondencia los aspectos relacionados con la complejidad que ejercen mayor influencia en la clasificación obtenida. En cuanto a la competencia de los métodos semi-supervisados en relación a la complejidad de los conjuntos temporales, se identificaron los métodos Democratic y TriT por presentar un mejor comportamiento frente a problemas de baja complejidad. Por el contrario, el método SETRED se mostró competitivo al enfrentar problemas de mediana y alta complejidad.

La tercera contribución de la tesis consiste en la propuesta de cinco criterios de parada para el algoritmo P1NNPU, el cual ha sido ampliamente utilizado en el dominio de las series temporales, específicamente para tratar la clasificación de problemas de una clase en el contexto semi-supervisado. Este algoritmo es una variante del Self-training donde el criterio de parada juega un papel crucial para impedir la inclusión errónea de ejemplos negativos en la clase positiva. El uso de la técnica de análisis gráfico, aplicada a las distancias mínimas obtenidas por el clasificador base 1NN, resultó una técnica efectiva para identificar la frontera entre clases. Específicamente, de los cinco criterios propuestos se identificó el



criterio  $SC4$  como el más robusto a partir de la experimentación realizada. La capacidad transductiva del método  $CBD-GA-SC4$  superó los métodos existentes del estado del arte, alcanzando resultados competitivos tanto para la distancia euclidiana como para DTW.

Por último, todos los métodos semi-supervisados utilizados durante el desarrollo de la tesis se incorporaron en un paquete de R llamado `ssc`, el cual permite resolver problemas de clasificación tanto inductivos como transductivos. Dicho paquete resulta fácil de extender con nuevos algoritmos semi-supervisados y ofrece diferentes formas de introducir los datos de entrenamiento, incluyendo la especificación de funciones de distancias que se encuentren implementadas en otros paquetes de R. El paquete también incluye la posibilidad de definir nuevos clasificadores base lo cual añade flexibilidad a esta propuesta.



# Índice

<b>Introducción</b>	<b>1</b>
A Planteamiento . . . . .	1
B Objetivos . . . . .	3
C Resumen . . . . .	4
<b>1 Aprendizaje semi-supervisado para series temporales</b>	<b>7</b>
1.1 Contexto y formalización del aprendizaje automático . . . . .	8
1.1.1 Aprendizaje supervisado . . . . .	9
1.1.2 Aprendizaje no supervisado . . . . .	10
1.1.3 Aprendizaje semi-supervisado . . . . .	11
1.2 Métodos semi-supervisados de auto-etiquetado . . . . .	13
1.2.1 Mecanismo de adición . . . . .	14
1.2.2 Esquema de clasificación . . . . .	15
1.2.3 Criterio de parada . . . . .	16
1.2.4 Métodos de auto-etiquetado representativos . . . . .	18
1.2.5 Hipótesis principales . . . . .	21
1.3 Series temporales . . . . .	22
1.3.1 Definición de series temporales . . . . .	22
1.3.2 Análisis de series temporales . . . . .	23
1.3.3 Minería de datos para series temporales . . . . .	25

1.4	Aprendizaje automático en el dominio temporal . . . . .	27
1.4.1	Medidas de disimilitud . . . . .	27
1.4.2	Métodos de clasificación . . . . .	32
1.4.3	Métodos de agrupamiento . . . . .	36
1.4.4	Métodos de clasificación semi-supervisada . . . . .	36
1.5	Sumario . . . . .	38
<b>2</b>	<b>Estudio sobre métodos de auto-etiquetado aplicados al dominio temporal</b>	<b>41</b>
2.1	Métodos seleccionados para el estudio experimental . . . . .	42
2.2	Marco experimental . . . . .	44
2.2.1	Conjuntos de datos experimentales . . . . .	44
2.2.2	Medidas de evaluación . . . . .	46
2.2.3	Consideraciones sobre los algoritmos . . . . .	47
2.3	Resultados experimentales . . . . .	49
2.3.1	Resultados transductivos . . . . .	50
2.3.2	Resultados inductivos . . . . .	55
2.3.3	Discusión sobre los resultados de clasificación . . . . .	60
2.3.4	Análisis de los tiempos de ejecución . . . . .	61
2.3.5	Efectividad de los métodos de auto-etiquetado . . . . .	62
2.4	Sumario . . . . .	73
<b>3</b>	<b>Influencia de la complejidad de los conjuntos de datos en el desempeño de los métodos de auto-etiquetado</b>	<b>75</b>
3.1	Complejidad de los datos . . . . .	76
3.2	Evaluación de las medidas de complejidad seleccionadas . . . . .	79
3.2.1	Análisis de los tiempos de ejecución . . . . .	86
3.3	Influencia de la complejidad en los resultados de clasificación . . . . .	87

---

3.3.1	Análisis de la exactitud . . . . .	88
3.3.2	Análisis de la ganancia de exactitud . . . . .	94
3.4	Sumario . . . . .	100
<b>4</b>	<b>CBD-GA: criterios de parada para el algoritmo 1NNP</b>	<b>103</b>
4.1	Aprendizaje PU y su aplicación al dominio temporal . . . . .	104
4.1.1	Formalización del aprendizaje PU . . . . .	104
4.1.2	Aprendizaje PU aplicado a la tarea de clasificación de series temporales . . . . .	106
4.1.3	Criterios de parada existentes para el Self-training . . . . .	109
4.2	Criterios de parada CBD-GA para el algoritmo P1NNPU . . . . .	112
4.2.1	Descripción de la frontera entre clases mediante tres curvas	113
4.2.2	Criterios de parada basados en CBD-GA . . . . .	117
4.2.3	Medidas estadísticas para la evaluación del criterio de parada	121
4.3	Marco experimental . . . . .	122
4.4	Resultados experimentales . . . . .	124
4.4.1	Comparación entre los criterios de parada propuestos . . . . .	125
4.4.2	Comparación con las propuestas de la literatura . . . . .	129
4.5	Sumario . . . . .	134
<b>5</b>	<b>ssc: paquete de R para clasificación semi-supervisada</b>	<b>137</b>
5.1	Lenguaje de programación R . . . . .	138
5.1.1	Repositorio CRAN . . . . .	140
5.1.2	Estructura de un paquete de R . . . . .	141
5.2	Paquetes y bibliotecas relacionados con la SSC . . . . .	142
5.3	Arquitectura del paquete <code>ssc</code> . . . . .	143
5.3.1	Funciones de entrenamiento . . . . .	144
5.3.2	Funciones de clasificación . . . . .	146

5.3.3	Medidas de evaluación . . . . .	147
5.3.4	Especificación de un nuevo clasificador base . . . . .	147
5.4	Ejemplos de uso del paquete <code>ssc</code> . . . . .	148
5.4.1	Preparación de los datos de entrenamiento . . . . .	149
5.4.2	Fase de entrenamiento . . . . .	152
5.4.3	Clasificación transductiva e inductiva . . . . .	154
5.4.4	Comparación entre los modelos entrenados . . . . .	155
5.4.5	Otras facilidades . . . . .	159
5.5	Sumario . . . . .	161
<b>Conclusiones y trabajos futuros</b>		<b>163</b>
A.	Resumen y Conclusiones . . . . .	163
B.	Publicaciones Asociadas a la Tesis . . . . .	166
C.	Trabajos Futuros . . . . .	166
<b>A</b>	<b>Parámetros específicos de los métodos semi-supervisados estudiados</b>	<b>169</b>
<b>B</b>	<b>Resultados de los métodos semi-supervisados estudiados</b>	<b>171</b>
B.1	Resultados transductivos e inductivos de los métodos semi-supervisados . . . . .	171
B.2	Resultados obtenidos con el 1NN como cota de referencia . . . . .	199
<b>C</b>	<b>Resultados de las medidas de complejidad</b>	<b>203</b>
C.1	Evaluación de las medidas de complejidad . . . . .	203
C.2	Dispersión entre la ganancia de exactitud y la complejidad . . . . .	210
<b>D</b>	<b>Demostración de las propiedades de la curva estable</b>	<b>217</b>
<b>E</b>	<b>Detalles sobre la implementación del paquete <code>ssc</code></b>	<b>219</b>

**Bibliografía**

**221**





# Índice de Siglas

- DT: *Decision Trees*, Árboles de decisión
- DTW: *Dynamic Time Warping*, Distorsión dinámica del tiempo
- *k*NN: *k-Nearest Neighbors*, *k* vecinos más cercanos
- PU: *Positive Unlabeled learning*, Aprendizaje semi-supervisado de clase positiva
- SSC: *Semi-Supervised Classification*, Clasificación semi-supervisada
- SSL: *Semi-Supervised Learning*, Aprendizaje semi-supervisado
- SVM: *Support Vector Machines*, Máquinas de soporte vectorial



# Introducción

## A Planteamiento

Resulta difícil imaginar una rama de la ciencia en la que no aparezcan datos que puedan ser modelados como series temporales. Según su definición [30, 144, 76], las series temporales se obtienen mediante la medición de variables a través del tiempo. Algunos de los dominios donde se ha hecho habitual el trabajo con datos temporales son: procesamiento de señales, reconocimiento de patrones, econometría, matemática financiera, pronóstico climático, electroencefalografía, ingeniería y comunicaciones, entre otros.

Por lo general, los datos almacenados a través del tiempo contienen información valiosa para su dominio de procedencia. Esto ha condicionado el surgimiento de una rama de la minería de datos que se encarga exclusivamente de las series temporales [59, 52]. Algunas de las características distintivas de las series temporales dificultan su análisis en el contexto de los métodos tradicionales de la minería de datos. Entre sus principales características se encuentran: alta numerosidad, gran número de dimensiones y una constante actualización de sus datos al transcurrir el tiempo. Uno de los enfoques de la minería de datos temporales consiste en adaptar los métodos tradicionales para tratar las series temporales como una clase especial de dato [84, 113].

La clasificación [159] constituye una de las principales tareas de la minería de datos y persigue como objetivo asociar datos con grupos predefinidos o clases. La mayoría de los algoritmos de clasificación asumen algún conocimiento de los datos o realizan fases de entrenamiento para estas clasificaciones. Muchos de los

algoritmos de clasificación conocidos se han adaptado para ser utilizados con series temporales, entre ellos se destacan: los árboles de decisión [49, 45], las máquinas de soporte vectorial [169, 105] y los  $k$  vecinos más cercanos [162, 125]. Este último, a pesar de su simplicidad, es uno de los más populares en la minería de datos temporales debido a los buenos resultados que ofrece [85]. Además, las redes neuronales artificiales [116] y la programación lógica inductiva [139] son otros de los métodos empleados en la clasificación de series temporales.

Todos los algoritmos previamente mencionados pertenecen al paradigma de la clasificación supervisada, el cual asume que existe una cantidad suficiente de ejemplos previamente etiquetados para ser utilizados por los algoritmos durante el entrenamiento. En el dominio temporal resulta usual contar con grandes volúmenes de datos sin etiquetar [66, 124, 25]. Por el contrario, la obtención de datos etiquetados resulta, en ocasiones, un proceso que consume tiempo y esfuerzo de los expertos. La utilización durante el aprendizaje, tanto de datos etiquetados como no etiquetados, se conoce en la literatura como aprendizaje semi-supervisado [29, 173]. Este paradigma permite emplear los datos no etiquetados con el objetivo de obtener resultados superiores que los obtenidos en caso de haber utilizado sólo los datos etiquetados. El aprendizaje semi-supervisado constituye una opción factible para aquellos problemas donde existe un número limitado de datos etiquetados.

Una de las primeras propuestas de este tipo de algoritmo, orientada a la clasificación de series temporales, es desarrollada por Wei y Keogh [157]. Esta se basa en un esquema de auto-etiquetado conocido como Self-training [165]. A partir de ese momento surgen otras propuestas basadas en varios métodos de aprendizaje semi-supervisado adaptados para el tratamiento de series temporales [136, 118, 119]. Algunas de estas propuestas pertenecen a una clase particular de problema donde sólo se conocen ejemplos etiquetados de una de las clases. En este tipo de aprendizaje aún existen problemas abiertos como el caso del criterio de parada en el algoritmo Self-training. De forma general, los resultados obtenidos por los métodos de aprendizaje semi-supervisado en el dominio temporal aún son deficientes, en el trabajo de Chen y colaboradores [33] se hace un análisis al respecto.

Además del Self-training existen otros métodos de auto-etiquetado [150]

con características atractivas que hacen factible su aplicación al dominio temporal. Específicamente, los métodos de auto-etiquetado basados en múltiples clasificadores [171, 172] ofrecen soluciones adecuadas y aún no se ha extendido su uso en la clasificación de datos temporales. Hasta el momento, los trabajos existentes sobre esta temática aún son escasos y poco concluyentes sobre cuáles enfoques resultan más efectivos para el trabajo con series temporales. Además, las características particulares de las series temporales añaden complejidad adicional a la tarea de clasificación, aspecto que debe tenerse en cuenta al estudiar la aplicabilidad de los métodos de auto-etiquetado en este dominio.

En esta tesis centramos nuestra atención en la aplicación de métodos de auto-etiquetado a la clasificación de series temporales. Existe un gran interés en impulsar el desarrollo de algoritmos semi-supervisado en esta tarea ya que se pueden emplear tanto para aumentar el conjunto de entrenamiento compuesto por series etiquetadas como para obtener modelos de clasificación. Por esta razón, en la presente tesis nos proponemos profundizar en la temática presentada con el fin de mostrar que los métodos de auto-etiquetado constituyen una alternativa viable para solucionar problemas de clasificación de datos temporales a partir de cantidades limitadas de ejemplos etiquetados.

## B **Objetivos**

Dada la necesidad de abordar la naturaleza temporal de los problemas de clasificación en el contexto socio-tecnológico actual y las evidencias existentes sobre la adecuación del paradigma semi-supervisado para su solución, se plantea como objetivo general de esta tesis: estudiar, diseñar y evaluar métodos semi-supervisados de auto-etiquetado para abordar la clasificación de series temporales, considerando las características distintivas implicadas en el tratamiento de este tipo de dato.

A grandes rasgos, para alcanzar este objetivo, se estudiará en primer lugar cómo adaptar métodos semi-supervisados prometedores que nunca se han aplicado al contexto temporal. En segundo lugar, se realizará una evaluación de varios métodos de auto-etiquetado aplicados a la clasificación de series tempora-

les. Finalmente, se harán propuestas de nuevos métodos basados en los resultados previos.

Los objetivos específicos en los que se desglosa el objetivo general son:

- Efectuar un estudio comparativo experimental entre varios métodos de auto-etiquetado de probada efectividad en aras de identificar cuáles ofrecen mejores resultados en la clasificación de series temporales.
- Desarrollar un estudio sobre la influencia que ejerce la complejidad de los problemas temporales, desde el punto de vista de la clasificación, en el desempeño de los métodos de auto-etiquetado.
- Proponer y validar criterios de parada para el algoritmo Self-training, orientados al aprendizaje semi-supervisado de una clase, que resulten más efectivos en comparación con los reportados en la literatura.
- Implementar un paquete de código abierto en lenguaje R para la aplicación de métodos de auto-etiquetado en la solución de problemas de clasificación semi-supervisada.

## C Resumen

Las tareas desarrolladas para alcanzar los objetivos planteados se describen en cinco capítulos, una sección de comentarios finales y cinco apéndices. La estructura de cada una de estas partes se introduce a continuación.

En el Capítulo 1 se exponen los fundamentos del aprendizaje semi-supervisado y la conveniencia de su aplicación en el dominio temporal. Inicialmente se introducen las nociones fundamentales del aprendizaje automático, enfatizando el contexto en el que se inserta el aprendizaje semi-supervisado. Además, se ofrece una descripción de los métodos de auto-etiquetado que son utilizados en las contribuciones de la memoria. Luego, se formaliza la definición de series temporales y se analiza el enfoque que aporta la minería de datos temporales. Posteriormente, se exponen los principales avances que ha experimentado el apren-

dizaje automático aplicado al dominio temporal y para finalizar se presenta un breve resumen del capítulo.

En el Capítulo 2 se realiza un estudio experimental que abarca varios métodos semi-supervisados de auto-etiquetado combinados con diversos esquemas de aprendizaje supervisado que se utilizan como clasificadores base. Inicialmente se resumen los métodos involucrados en el estudio tanto supervisados como semi-supervisados. Posteriormente, se describen los detalles de la experimentación, los cuales incluyen: los conjuntos de datos seleccionados, los parámetros prefijados para los distintos algoritmos utilizados y las medidas empleadas para la evaluación de los resultados. Una vez presentados los resultados experimentales, se realiza un análisis comparativo de los mismos tanto a nivel transductivo como inductivo. Al finalizar el capítulo se presenta un breve resumen con las principales conclusiones del capítulo.

En el Capítulo 3 se desarrolla un estudio experimental sobre la influencia que ejerce la complejidad, asociada a los problemas de clasificación de datos temporales, en el desempeño de las técnicas de auto-etiquetado estudiadas. Inicialmente se mencionan los principales aspectos que aportan complejidad a la tarea de clasificación y se describen varias medidas de complejidad seleccionadas para desarrollar el estudio. Luego, se presentan los resultados obtenidos de la evaluación de dichas medidas para diferentes proporciones de ejemplos etiquetados. La siguiente parte del capítulo está dedicada al análisis de la relación existente entre la complejidad de los datos temporales y el desempeño de los métodos de auto-etiquetado. Para finalizar, se presenta un breve resumen de los resultados más relevantes del capítulo.

En el Capítulo 4 se propone un método de clasificación semi-supervisada para series temporales enfocado en un caso especial donde sólo se encuentran disponibles ejemplos etiquetados de una clase. Inicialmente se formaliza este tipo de aprendizaje y su aplicación en el dominio temporal. Luego, se describe la propuesta y la familia de criterios de parada que esta incluye. Posteriormente, se describe el diseño experimental utilizado para la evaluación de la propuesta y se ofrecen algunos detalles de los conjuntos de datos experimentales. La siguiente parte del capítulo está dedicada al análisis y discusión de los resultados así como la comparación con las propuestas existentes del estado del arte. Por último, se

presenta un breve resumen con los resultados más importantes.

En el Capítulo 5 se presenta el paquete `ssc` desarrollado en lenguaje R, el cual implementa una colección de métodos de clasificación semi-supervisados de auto-etiquetado. Inicialmente se realiza un resumen de las principales características del lenguaje R y las facilidades que ofrece para el desarrollo de paquetes. Posteriormente, se describen las principales funciones del paquete `ssc`. Estas incluyen el proceso de entrenamiento desde la preparación de los datos de entrada hasta la etapa de clasificación donde se utiliza el modelo entrenado. De todas estas etapas se muestran ejemplos donde se utiliza un problema clásico del dominio temporal. Para finalizar, se presenta un breve resumen del capítulo.

En la Sección “Conclusiones y trabajo futuro” se resumen los principales resultados obtenidos en esta memoria, las conclusiones que se pueden extraer de los mismos y se comentan algunas líneas de trabajo futuro.

Para finalizar, se incluyen cinco apéndices con información complementaria a la presentada en los capítulos anteriores. En el primer apéndice se recogen los detalles de cada uno de los parámetros que poseen los métodos de auto-etiquetado estudiados. En el segundo apéndice se muestran los resultados de los métodos de auto-etiquetado evaluados durante el estudio presentado en el Capítulo 2. En el tercer apéndice se muestran los resultados de la evaluación de las medidas de complejidad involucradas en el estudio del Capítulo 3. El cuarto apéndice se dedica a la demostración de una de las propiedades que se enuncian en el Capítulo 4. El último apéndice aporta información adicional sobre el esfuerzo requerido durante el desarrollo del paquete `ssc`. La memoria termina con una recopilación bibliográfica donde se recogen las principales contribuciones de la materia estudiada.



# Capítulo 1

## Aprendizaje semi-supervisado para series temporales

En el presente capítulo se exponen los fundamentos del aprendizaje semi-supervisado y la conveniencia de su aplicación en el dominio específico de las series de tiempo. En la sección 1.1 se presentan las nociones fundamentales del aprendizaje automático y el contexto en el que se inserta el aprendizaje semi-supervisado. En la sección 1.2 se describe una categoría de métodos de clasificación semi-supervisada que resultan de especial interés para el presente estudio. Luego, en la sección 1.3 se formaliza la definición de series temporales y los tipos de análisis existentes para este tipo de dato, haciendo especial énfasis en el enfoque que aporta la minería de datos temporales. Por su importancia, se dedica la sección 1.4 a la exposición de los principales avances que ha experimentado el aprendizaje automático aplicado al dominio temporal. Por último, la sección 1.5 concluye el capítulo con un sumario donde se resaltan los principales aspectos tratados.

## 1.1 Contexto y formalización del aprendizaje automático

Desde el surgimiento de las computadoras se han realizado intentos para que estas aprendan. Aunque las computadoras aún no pueden aprender en la misma forma en que las personas lo hacen, se han desarrollado algoritmos que resultan efectivos para enfrentar ciertas tareas de aprendizaje. El aprendizaje automático o aprendizaje de máquinas [112] es un campo multidisciplinario que tiene como objetivo que programas de computadora mejoren su funcionamiento en cierta tarea a partir de la experiencia. Específicamente, se trata de crear modelos capaces de generalizar hipótesis a partir de la experiencia suministrada, fundamentalmente, en forma de ejemplos. La minería de datos [159] ha contribuido al desarrollo del aprendizaje automático ya que este tipo de algoritmos ha sido ampliamente utilizado en el descubrimiento de información valiosa a partir de datos almacenados. Con frecuencia el aprendizaje automático se solapa con la estadística y resulta difícil establecer una línea divisoria entre ambos.

Los ejemplos de entrenamiento, también conocidos como casos del problema, son utilizados como entrada en la construcción de un modelo para luego poder realizar predicciones frente a nuevos datos. Podemos definir un ejemplo de entrenamiento de la siguiente forma:

**Definición 1.1.** Un ejemplo o caso  $x$  se suele representar como una  $D$ -upla  $x = (x_1, x_2, \dots, x_D)$  donde cada elemento  $x_a$  representa el valor que toma el caso  $x$  en el atributo  $a$ . A la longitud  $D$  se le conoce como *dimensión* de  $x$ .

Dichos ejemplos, pueden o no estar acompañados de una salida durante el entrenamiento. Considerando esta diferenciación, los algoritmos de aprendizaje se han clasificado de manera tradicional en dos amplias categorías:

- aprendizaje supervisado (sección 1.1.1)
- aprendizaje no supervisado (sección 1.1.2)

Las principales características de cada categoría y el surgimiento de una tercera son objetos de análisis en esta sección.

### 1.1.1 Aprendizaje supervisado

En el aprendizaje supervisado, para cada ejemplo  $x^i$  existe asociada una salida  $y^i$ . El objetivo de este tipo de aprendizaje consiste en ajustar un modelo que relacione el valor de salida  $y^i$  con los valores de los atributos predictores en  $x^i$ . Formalmente podemos definir el aprendizaje supervisado de la siguiente forma:

**Definición 1.2.** Sea  $\mathcal{X}$  el dominio de los ejemplos de entrenamiento y sea  $\mathcal{Y}$  el dominio de las salidas. Dado un conjunto de  $l$  ejemplos de entrenamiento  $\{(x^i, y^i)\}_{i=1}^l$ , el aprendizaje supervisado tiene como objetivo entrenar una función  $f : \mathcal{X} \mapsto \mathcal{Y}$  capaz de predecir el valor correcto de  $y$  para un futuro valor de  $x$ .

En dependencia del dominio al que pertenece el valor  $y$ , entonces se puede categorizar el problema de aprendizaje en clasificación o regresión. Cuando  $\mathcal{Y}$  representa un dominio discreto de etiquetas o clases entonces consideramos la función  $f$  como un clasificador. Por el contrario, cuando  $\mathcal{Y}$  representa un dominio continuo la función  $f$  se denomina función de regresión.

Existe una amplia gama de enfoques que han sido utilizados tradicionalmente en la obtención de modelos supervisados. El enfoque basado en casos [112], también conocido como aprendizaje perezoso, simplemente almacena los ejemplos de entrenamiento. La generalización a partir de estos ejemplos se pospone hasta el momento en que un nuevo caso del problema debe ser clasificado. En ese instante se analiza la relación entre el nuevo caso y los ejemplos previamente etiquetados con el objetivo de asignar un valor de salida a dicho caso. El método de los  $k$  vecinos más cercanos [38, *k-Nearest Neighbors (kNN)*], es uno de los más conocidos dentro de este enfoque y se utiliza tanto en tareas de clasificación como de regresión. El  $k$ NN, en presencia de un nuevo caso, estima el valor de la función objetivo a partir de los valores de salida almacenados en los  $k$  ejemplos más cercanos.

El aprendizaje de árboles de decisión [129, *Decision Trees (DT)*] es un método de inferencia inductiva en el cual la función aprendida es representada mediante un árbol de decisión. Este tipo de método clasifica nuevos casos a partir del camino que sigue el caso cuando es orientado hacia abajo desde la raíz hasta un nodo hoja, el cual provee la clasificación de dicho caso. Cada nodo del árbol

especifica alguno de los atributos que conforman el problema y las ramas que descienden de un nodo se corresponden con cada uno de los posibles valores que puede tomar el atributo en cuestión. La forma en que se construye el árbol distingue los diferentes métodos existentes. El ID3 [129] y el C4.5 [130] son algunos de los más utilizados.

Otro método que ha ganado gran popularidad en los últimos años son las máquinas de soporte vectorial [153, *Support Vector Machines* (SVM)]. Este método fue desarrollado para resolver problemas de clasificación aunque ha sido extendido al dominio de los problemas de regresión. Su formulación se basa en el principio de Minimización del Riesgo Estructural (*Structural Risk Minimization*), el cual ha demostrado ser superior que el principio tradicional de Minimización del Riesgo empírico (*Empirical Risk Minimization*). Esta diferencia provee a las SVMs de una mayor habilidad para generalizar. Un modelo de SVM construye un hiperplano o conjunto de hiperplanos en un espacio característico de alta dimensionalidad, obteniendo una separación óptima entre las clases. En la búsqueda del hiperplano óptimo sólo se necesita calcular los productos escalares de los vectores en el espacio característico. Las funciones núcleo son utilizadas con este fin, permitiendo el cálculo de los productos escalares en el espacio de entrada en lugar del espacio característico.

Los métodos de aprendizaje supervisado descritos anteriormente se encuentran entre los más influyentes de la minería de datos y así lo expresa la investigación realizada en el año 2007 por Wu y colaboradores [161].

### 1.1.2 Aprendizaje no supervisado

El aprendizaje no supervisado es considerado un reto aún mayor que el supervisado debido a que los ejemplos de entrenamiento no están asociados con una salida. Esta situación se considera no supervisada debido a que no existe una variable de respuesta que pueda controlar el aprendizaje. Formalmente podemos definir el aprendizaje no supervisado de la siguiente forma:

**Definición 1.3.** El aprendizaje no supervisado tiene como entrada un conjunto de entrenamiento compuesto por  $u$  ejemplos de la forma  $\{x^j\}_{j=1}^u$ . No existe ningún tipo de supervisión sobre la forma en que los ejemplos deben ser manejados.

Entre las principales tareas que se incluyen dentro del aprendizaje no supervisado se encuentra la búsqueda de conglomerados (*clustering*) o agrupamiento. El objetivo de este enfoque es definir grupos de forma tal que los ejemplos incluidos dentro de un mismo grupo resulten similares y los ejemplos en diferentes grupos sean disimilares entre sí.

Dado que en este tipo de métodos no se conoce una salida definida, no hay una medida exacta para estimar el error de la solución obtenida. Esta característica lo diferencia del aprendizaje supervisado, donde sí es posible evaluar los resultados, ya que el algoritmo se prueba sobre un conjunto de ejemplos donde se conoce de antemano su salida asociada.

### 1.1.3 Aprendizaje semi-supervisado

Para lograr un buen desempeño de las técnicas de aprendizaje supervisado es necesario contar con una cantidad suficiente de ejemplos que le permitan al modelo generalizar de manera adecuada la hipótesis subyacente en el conjunto de entrenamiento. En los problemas de clasificación resulta en ocasiones difícil obtener ejemplos clasificados o etiquetados, pues esto requiere del tiempo y esfuerzo de personal especializado en un determinado dominio. Por el contrario, la disponibilidad de grandes volúmenes de datos sin etiquetar, por lo general, es mucho mayor. Estos contienen información valiosa acerca de la estructura y características del problema a analizar. La posibilidad de mejorar los resultados al explotar los ejemplos no etiquetados en combinación con los etiquetados provocó el surgimiento del paradigma de aprendizaje semi-supervisado [29, *Semi-Supervised Learning* (SSL)]. La Figura 1.1 ilustra cómo puede resultar útil este paradigma para obtener una hipótesis más acertada de un problema donde sólo se conoce un número limitado de ejemplos etiquetados.

Como su nombre lo indica, el SSL ocupa un área intermedia entre el aprendizaje supervisado y el no supervisado. Se reconoce el trabajo de [111] como el primero en acuñar el término **semi-supervisado** aplicado a la tarea de clasificación a partir de ejemplos etiquetados y no etiquetados. Se reconocen dos vertientes principales del SSL: clasificación semi-supervisada (*Semi-Supervised Classification* (SSC)) y el agrupamiento con restricciones (*constrained clustering*). Consi-

derando que este trabajo se enfoca en el primero a continuación presentamos su definición formal:

**Definición 1.4.** La clasificación semi-supervisada se considera una extensión de la clasificación supervisada donde el conjunto de entrenamiento está formado por un conjunto de  $l$  ejemplos etiquetados  $L = \{(x^i, y^i)\}_{i=1}^l$  y  $u$  ejemplos sin etiquetar  $U = \{x^j\}_{j=l+1}^{l+u}$ . Típicamente se asume que el número de los ejemplos no etiquetados es mucho mayor que los etiquetados, o lo que es lo mismo,  $u \gg l$ . El objetivo de la clasificación semi-supervisada es entrenar un clasificador  $f$  a partir de los conjuntos  $L$  y  $U$ , de manera que se obtenga un clasificador más exacto que el obtenido únicamente a partir de los ejemplos supervisados.

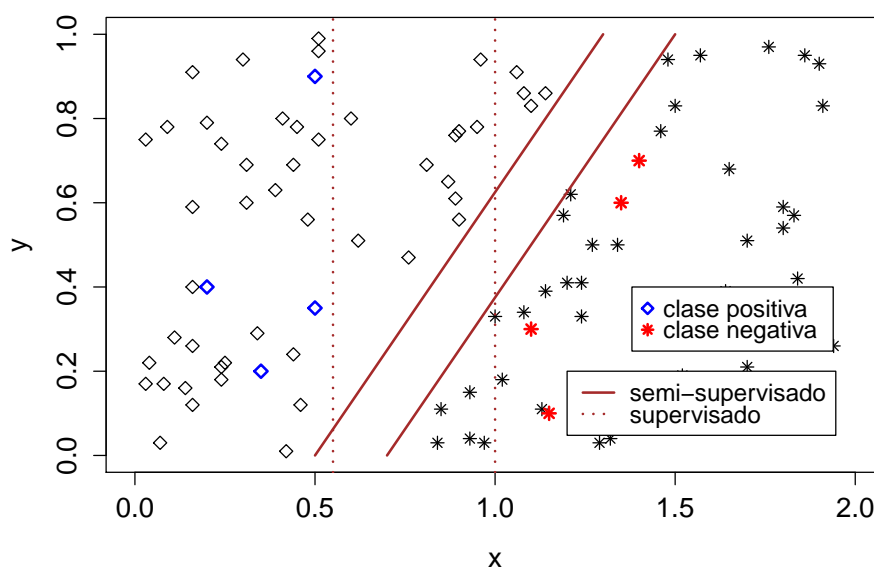


Figura 1.1: Los ejemplos no etiquetados (color negro) pueden ayudar a definir mejor la estructura del problema. La frontera inferida utilizando sólo los ejemplos etiquetados (color azul y rojo) es incorrecta y provoca errores en la clasificación.

En dependencia del objetivo que se persiga con las técnicas de SSC, estas se dividen en dos categorías [29]: inductivas y transductivas. El aprendizaje transductivo tiene como objetivo predecir las clases de los ejemplos no etiquetados que forman parte del conjunto de entrenamiento. Por el contrario, el aprendizaje inductivo tiene como objetivo crear un modelo capaz de generalizar la hipótesis

aprendida durante el entrenamiento y de esta forma clasificar nuevos casos que no hayan sido utilizados en el proceso de aprendizaje.

Existen diferentes enfoques que responden al paradigma de la SSC, entre ellos se encuentran: los modelos generativos [60], los modelos basados en grafos [18] y las SVMs semi-supervisadas [82, *Semi-Supervised SVMs* (S3VM)]. Cada enfoque asume determinados supuestos [173] acerca de la distribución que siguen los ejemplos etiquetados y los no etiquetados, así como la relación entre estos. Cuando se violan las suposiciones que un determinado método de SSC realiza sobre los datos, lejos de obtenerse una mejora, la adición de ejemplos no etiquetados degrada el desempeño en comparación con el aprendizaje supervisado [35, 14].

Otro enfoque exitoso, pertenecientes al paradigma SSC, basa su funcionamiento en la utilización de métodos tradicionales de clasificación supervisada. Los métodos que siguen este enfoque, denominados métodos de auto-etiquetado (*self-labeled*), intentan aumentar el conjunto original de ejemplos etiquetados  $L$  a partir de las clasificaciones que con mayor certeza se realicen de los ejemplos en  $U$ . Tradicionalmente, sólo se consideraban dentro de esta categoría los métodos Self-training [165] y Co-training [19], pero en la taxonomía realizada en el 2014 por Triguero y colaboradores [150] queda establecida que es una familia de métodos mucho más amplia.

## 1.2 Métodos semi-supervisados de auto-etiquetado

Los métodos de auto-etiquetado siguen una metodología de envoltorio a partir de la utilización de uno o varios algoritmos supervisados de clasificación (clasificadores base) con el objetivo de determinar la clase más probable de los ejemplos no etiquetados. Es por esto que los clasificadores base juegan un papel fundamental en la estimación de los ejemplos en  $U$  que pueden ser clasificados con una mayor certeza. Con la ampliación del conjunto  $L$  original se pretende modificar o reformar las hipótesis obtenidas únicamente con los datos etiquetados. Los métodos de auto-etiquetado presentes en la literatura se diferencian princi-

palmente en un conjunto de aspectos, los cuales serán objeto de análisis en las próximas secciones:

- Mecanismo de adición (sección 1.2.1)
- Esquema de clasificación (sección 1.2.2)
- Criterio de parada (sección 1.2.3)

### 1.2.1 Mecanismo de adición

El mecanismo de adición permite que el conjunto  $L$  inicial aumente con nuevos ejemplos, formando un conjunto extendido  $EL$ , a partir de un proceso iterativo que tiene lugar dentro de los algoritmos de auto-etiquetado. Este mecanismo se puede clasificar en tres tipos [150]:

- **Incremental:** Un enfoque estrictamente incremental comienza con un conjunto inicial de ejemplos etiquetados que paso a paso es incrementado con los ejemplos de  $U$  que cumplen cierto criterio. Un aspecto crucial de este mecanismo es la forma en la cual se determina la pertenencia a cada una de las clases de los ejemplos en  $U$ . La hipótesis aprendida es modificada con la adición de estos ejemplos en  $EL$  y es por esto que resulta determinante el orden en que se añaden dichos ejemplos. Otro aspecto relacionado con este enfoque es el número de ejemplos que son añadidos en cada iteración. Este número puede definirse como una constante independiente de la distribución que sigan las clases del problema a resolver. Otro enfoque consiste en definir este número de forma proporcional a la cantidad de ejemplos, por cada clase, presentes en el conjunto inicial  $L$ . Una ventaja del enfoque incremental es que puede resultar más rápido que los enfoques no incrementales. No obstante, la adición de ejemplos incorrectamente clasificados puede deteriorar considerablemente la hipótesis aprendida causando el problema conocido como reforzamiento del error.
- **Lote:** Este enfoque decide cuáles ejemplos en  $U$  cumplen el criterio de adición antes de adicionar cualquiera de ellos en  $EL$ . Posteriormente, todas los ejemplos seleccionadas son añadidos de una sola vez. En este sentido, las



técnicas basadas en lote no asignan una clase definitiva a los ejemplos no etiquetadas durante el proceso de aprendizaje. Este tipo de procesamiento puede resultar más costoso en tiempo que el enfoque incremental.

- **Modificable:** Los modelos modificables surgen como una solución a las desventajas que presentan los modelos estrictamente incrementales. Este enfoque basa su funcionamiento en la posibilidad de incluir o excluir ejemplos que cumplan determinado criterio, de manera iterativa. Este mecanismo permite realizar rectificaciones en operaciones de adición ya realizadas, con lo cual puede mejorar la exactitud del conjunto *EL*. Típicamente estos métodos han sido diseñados con el objetivo de evitar la introducción de ejemplos ruidosos en *EL* durante cada iteración [96, 151]. De manera general cualquier método incremental o de lote que incluya una fase de limpieza de ejemplos dudosos puede ser considerado un modelo de tipo modificable. La principal desventaja de este enfoque es que usualmente requiere un costo computacional más elevado que el de los modelos anteriores.

### 1.2.2 Esquema de clasificación

El esquema de clasificación utilizado por los métodos de auto-etiquetado comprende el número de clasificadores utilizados siguiendo una metodología de envoltorio durante el proceso de obtención del conjunto *EL*. Se pueden identificar dos modelos posibles [150]:

- **Monoclasificador:** Este modelo de clasificación está formado por un único clasificador el cual asigna la clase más probable a los ejemplos no etiquetados. Esto implica que las probabilidades por cada clase deben ser calculadas de forma explícita. La manera en que se calculan las probabilidades depende del tipo de clasificador empleado. Por ejemplo en los DT las probabilidades se calculan a partir de la certeza mediante la cual la hoja determinó la predicción [23]. Otros métodos, como el *k*NN, aproximan la confianza de las predicciones a partir de las distancias. De manera general, este tipo de modelo de clasificación se destaca por su simplicidad y la posibilidad de obtener las probabilidades sin necesidad de emplear un gran costo computacional.

- **Multclasificador:** Este modelo utiliza de manera combinada la hipótesis aprendida por varios clasificadores. La idea subyacente del modelo basado en multclasificadores, para un escenario de SSL con un número limitado de ejemplos etiquetados, es que la combinación de varios clasificadores débiles puede adquirir una mayor capacidad de generalización que un único clasificador [22]. Para agregar la confianza de los clasificadores se utilizan comúnmente dos enfoques: consenso entre clasificadores (voto mayoritario) y la combinación de las probabilidades obtenidas por cada clasificador. Un efecto colateral derivado de la utilización de múltiples clasificadores es el aumento del costo computacional.

Otro aspecto a tener en consideración en los métodos de auto-etiquetado es el esquema de aprendizaje utilizado por los clasificadores. Independientemente de la cantidad utilizada estos pueden o no ser del mismo tipo:

- **Monoaprendizaje:** Este enfoque se puede relacionar tanto con el modelo de clasificación simple como con el modelo basado en múltiples clasificadores. Un único clasificador responde a un tipo de aprendizaje único. La idea de utilizar un único tipo de aprendizaje para varios clasificadores está inspirada en el éxito obtenido con la agregación de múltiples versiones de un predictor en el escenario supervisado, por ejemplo el *Bagging* [22].
- **Multiaprendizaje:** Este enfoque sólo se corresponde con el modelo basado en múltiples clasificadores donde cada clasificador proviene de un esquema de aprendizaje diferente. En un modelo multiaprendizaje la confianza de las predicciones se calcula a partir de integrar esquemas de aprendizaje con diferente naturaleza. Este tipo de aprendizaje funciona bajo la hipótesis de que diferentes esquemas de aprendizaje generan modelos localmente distintos, incluso cuando son generados a partir del mismo conjunto de entrenamiento [171]. Un inconveniente específico de este enfoque es la elección de los esquemas de aprendizaje que resulten más adecuados.

### 1.2.3 Criterio de parada

Los métodos de auto-etiquetado basan su funcionamiento en un proceso iterativo que continúa mientras no se satisfaga un determinado criterio de parada.

Este criterio resulta de gran importancia pues define el tamaño del conjunto  $EL$  que se obtendrá y por consiguiente también influye en la hipótesis a aprender. Para abordar esta problemática se han definido los criterios de parada respecto a tres elementos, fundamentalmente:

- **Tamaño de  $U$ :** Un primer enfoque, generalmente asociado al algoritmo Self-training, consiste en iterar sobre el conjunto  $U$  hasta que este haya quedado totalmente agotado [173]. La principal desventaja de este enfoque es que aumenta considerablemente el riesgo de incluir ejemplos erróneamente clasificados en  $EL$ , con la consecuente afectación en la exactitud de la clasificación. Esto se debe a que en cada iteración los ejemplos que son eliminados de  $U$  e insertados en  $EL$  son aquellos que han sido clasificados con mayor certeza. Por tanto, los ejemplos de entrenamiento no etiquetados que resulten más dudosos al algoritmo quedarán relegados en  $U$  hasta las iteraciones finales. Una variante de este enfoque, que tiene como objetivo aliviar esta situación, consiste en agotar sólo un por ciento de la cantidad inicial de ejemplos en  $U$ . En este caso habría que definir este por ciento mediante un parámetro del algoritmo [148].
- **Número de iteraciones:** Un segundo criterio de parada [20] consiste en establecer un número limitado de iteraciones. A lo anterior se añade la selección de los ejemplos de entrenamiento a partir de un subconjunto  $U'$  y no directamente de  $U$ . El subconjunto  $U'$  se forma seleccionando ejemplos de  $U$  de manera aleatoria en cada iteración. La principal dificultad que presenta este enfoque es que generalmente la cantidad de iteraciones es fijada con antelación y no es adaptativa respecto al tamaño del conjunto de entrenamiento utilizado.
- **Cambios en la hipótesis:** Un tercer criterio se establece cuando la hipótesis aprendida por los clasificadores utilizados deja de sufrir cambios durante el proceso de auto-etiquetado [172]. Este criterio aunque limita el número de ejemplos que se añaden a  $EL$  no garantiza la exclusión de ejemplos incorrectamente clasificados en este conjunto.

### 1.2.4 Métodos de auto-etiquetado representativos

En la literatura se han propuesto decenas de métodos de tipo auto-etiquetado [150]. A continuación se presenta una breve descripción de un grupo representativo de este tipo de métodos.

- **Self-training estándar:** este es un método monoclificador y por consiguiente monoaprendizaje que adiciona en  $EL$  los ejemplos que, con mayor certeza, son clasificados y extraídos de  $U$  en un proceso iterativo. El mecanismo de adición utilizado es de tipo incremental y el proceso se detiene cuando se alcanza el criterio de parada, por ejemplo que se agote un determinado por ciento del tamaño inicial de  $U$ . El esquema de la Figura 1.2 muestra el funcionamiento de este método.

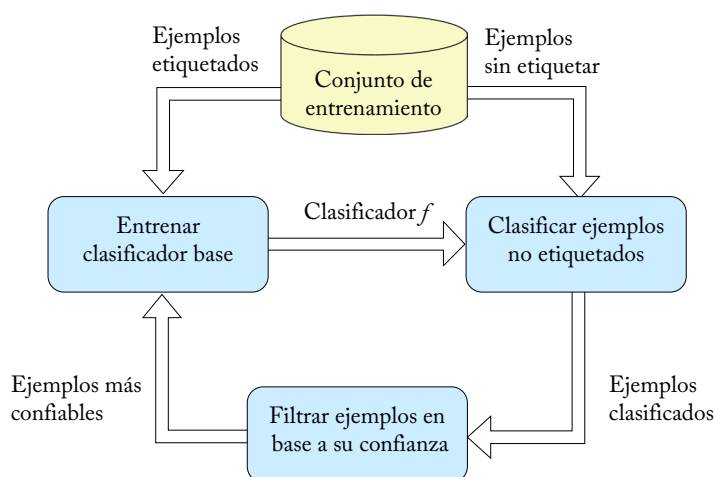


Figura 1.2: Método Self-training estándar.

- **Co-training estándar:** este es un método multiclificador y multiaprendizaje, el cual asume que existen dos vistas o conjuntos de atributos, que describen el problema de manera independiente, las cuales son suficientes por sí mismas para efectuar el aprendizaje. Se entrena un clasificador por cada vista y los ejemplos que con mayor certeza son seleccionados y etiquetados pasan a formar parte del conjunto de entrenamiento perteneciente a la otra vista. El mecanismo de adición es incremental con un número prefi-

jado de iteraciones. El esquema de la Figura 1.3 muestra el funcionamiento de este método.

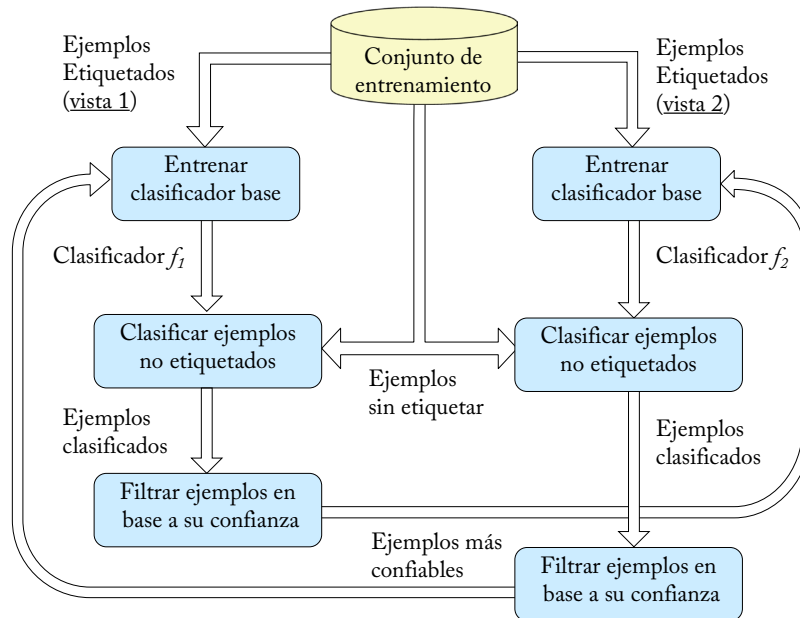


Figura 1.3: Método Co-training estándar.

- **SETRED** [96, *Self-TRaining with EDiting*]: este método constituye una variante del Self-training y tienen en común que son modelos de tipo monoclasificador y monoaprendizaje. La principal diferencia radica en el mecanismo de adición. SETRED introduce una técnica de edición para filtrar los ejemplos ruidosos que ya han sido etiquetados por el método. En cada iteración SETRED realiza un filtrado de los ejemplos extraídos de  $U$  utilizando información local proporcionada por el grafo de vecindades [174]. Este proceso iterativo finaliza cuando se alcanza un número prefijado de iteraciones.
- **SNNRCE** [155, *Self-training Nearest Neighbor Rule using Cut Edges*]: este método constituye una variante de SETRED que incluye una primera etapa donde se identifican, mediante el grafo de vecindades, aquellos ejemplos que pueden ser etiquetados con mayor certeza. Estos ejemplos son añadidos al conjunto  $EL$  antes de pasar a la segunda etapa donde se

aplica el Self-training estándar y la regla del vecino más cercano como esquema de clasificación. En la tercera etapa se modifican las etiquetas de los ejemplos añadidos que se consideran mal etiquetados si cumplen con una determinada prueba estadística. El criterio de parada que se aplica para la segunda etapa consiste en satisfacer la cantidad de ejemplos esperados para la clase minoritaria. Esta cantidad se establece de manera proporcional a las cantidades observadas en  $L$ .

- **Tri-training** [172]: este método sigue un estilo similar al Co-training pero en lugar de entrenar dos clasificadores entrena tres. A diferencia del Co-training, este método no necesita dos vistas diferentes de los datos y los clasificadores comparten el mismo esquema de aprendizaje. La diversidad inicial de los clasificadores se logra a partir de una manipulación del conjunto inicial de ejemplos etiquetados, generalmente utilizando el *Bagging*. En cada iteración, los ejemplos seleccionados y etiquetados sólo son añadidos al conjunto de entrenamiento de un clasificador si existe consenso en los otros dos clasificadores y se cumplen ciertas condiciones. El criterio de parada se satisface cuando, durante una iteración, no se modifica la hipótesis de ningún clasificador.
- **Co-bagging** [53, *Co-training by committee: bagging*]: como su nombre lo indica este método es una variante del Co-training que no necesita tener diferentes vistas sobre los datos. La cantidad de clasificadores se establece como un parámetro del método y todos comparten el mismo esquema de clasificación. Una vez más, la diversidad inicial de los clasificadores puede lograrse usando *Bagging*. La cantidad de ejemplos que se añaden de cada clase depende de las cantidades iniciales observadas en  $L$ . El proceso iterativo termina cuando se alcanza una cantidad prefijada de iteraciones.
- **Democratic** [171, *Democratic co-learning*]: este método es multclasificador y multiaprendizaje. La cantidad de clasificadores a utilizar se establece como un parámetro del método aunque se especifica un número mínimo de tres. Durante la etapa inicial del método todos los clasificadores son entrenados utilizando el conjunto  $L$ . En cada iteración se propone una clase para cada ejemplo no etiquetado utilizando el criterio del voto mayoritario. A los clasificadores que no estimaron la clase de acuerdo a la mayoría, para un

determinado ejemplo, se les incluye este en su conjunto de entrenamiento si se satisfacen ciertas condiciones. El proceso iterativo termina cuando no ocurren cambios durante una iteración en ninguno de los conjuntos de entrenamiento pertenecientes a los clasificadores. La hipótesis final retornada por el modelo se basa en un voto mayoritario pesado de los clasificadores finales obtenidos.

### 1.2.5 Hipótesis principales

A diferencia de otros enfoques los métodos de auto-etiquetado no asumen supuestos específicos relacionados con la distribución de los datos de entrada. En lugar de esto, como en el caso del método Self-training, asumen que sus predicciones o al menos las de mayor certeza tienden a ser correctas [173]. Este supuesto se cumple, por ejemplo, cuando las clases del problema forman conglomerados que se encuentran a una distancia suficiente entre sí [29].

El método Co-training asume que existen dos vistas de los datos condicionalmente independientes dada la clase y que cada una de estas vistas por sí sola es suficiente para realizar buenas clasificaciones [173]. Este supuesto en la mayoría de los casos resulta muy difícil de cumplir por lo que se han propuesto variantes donde en lugar de tener dos vistas de los datos se entrenan dos clasificadores supervisados [67]. De forma general, el principal supuesto que realizan los modelos de tipo multclasificador es que el aprendizaje se hace efectivo cuando existe un consenso en el conjunto de las hipótesis aprendidas y dicho consenso posee un bajo riesgo empírico [93, 166].

Cuando los supuestos que realizan los modelos de aprendizaje semi-supervisado no se cumplen entonces existe un riesgo real de que la adición de ejemplos no etiquetados deteriore la exactitud de la clasificación [36, 103]. El problema de verificar estos supuestos o matemáticamente formalizarlos aún continúa como un problema abierto [104].

## 1.3 Series temporales

En la mayoría de las ramas científicas, la ingeniería y el comercio existen variables que son medidas secuencialmente a través del tiempo. Los bancos registran las tasas de interés y de cambio de monedas diariamente. Las oficinas de meteorología llevan el control de las precipitaciones y la temperatura en diferentes lugares y con diferente granularidad. Cuando una variable es medida secuencialmente en el tiempo o en un intervalo determinado, los datos tomados forman una serie de tiempo [39].

Las series temporales son susceptibles a contener información valiosa para su dominio de procedencia. Existen dos ramas dedicadas al estudio de las series de tiempo: el **análisis de series temporales** y la **minería de datos temporales**. El análisis tradicional de series temporales [30] comprende tanto métodos matemáticos como estadísticos, los cuales han sido utilizados en el pronóstico de valores futuros o con la finalidad de interpretar eventos ocurridos. El segundo enfoque para el tratamiento de las series temporales surge con la consolidación de una rama de la minería de datos orientada específicamente al estudio de datos temporales [59]. Métodos tradicionales del aprendizaje automático han sido satisfactoriamente aplicados al dominio temporal mediante la modelación de las series como un tipo especial de dato. El uso de estos métodos, en combinación con el aumento de la potencia de cómputo, ha propiciado su aplicación en dominios diversos tales como el reconocimiento del lenguaje natural [141, 78], la biométrica [89, 120], la medicina [25, 66] y la astronomía [79], entre otros.

### 1.3.1 Definición de series temporales

Según [30] una serie temporal consiste en una colección de observaciones realizadas de manera secuencial en el tiempo. Otros autores [154, 24] ofrecen una definición más rigurosa:

**Definición 1.5.** Una serie temporal  $s$  consiste en una secuencia de  $n$  pares  $((s_1, t_1), (s_2, t_2), \dots, (s_i, t_i), \dots, (s_n, t_n))$  ( $t_1 < t_2 < \dots < t_i < \dots < t_n$ ), donde cada  $s_i$  es un punto en el espacio  $\mathcal{D}$ -dimensional al que pertenecen los datos, y



cada  $t_i$  es el instante de tiempo al cual se corresponde la ocurrencia de  $s_i$ .

Una serie de tiempo en cada observación  $s_i$  puede contener valores de varias variables. Si la cantidad de variables medidas es uno, se llama **univariada**, y cuando es mayor que uno, se llama **multivariada**.

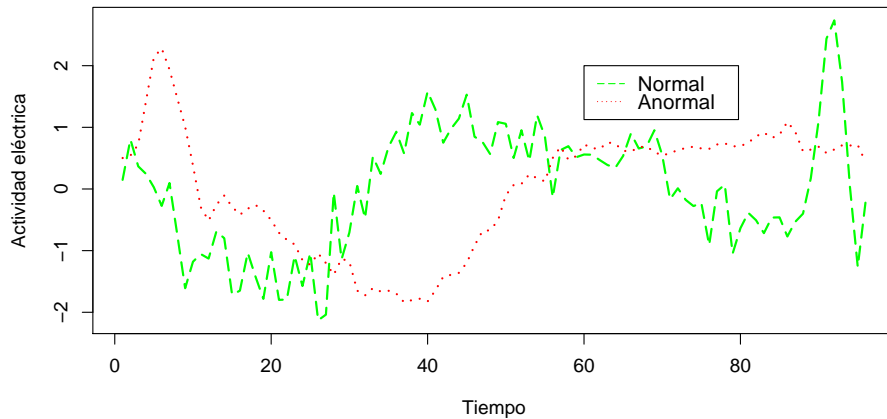


Figura 1.4: Fragmento de un electrocardiograma que describe una pulsación del corazón en una persona sana (Normal) y en una enferma (Anormal).

Una serie temporal es considerada **continua** cuando las observaciones son hechas de manera continua en el tiempo. Si las observaciones tienen lugar sólo en momentos específicos entonces es considerada como una serie temporal **discreta**. La continuidad de una serie se determina por el modo de tomar los valores en el tiempo y es independiente de la naturaleza continua o no de las variables medidas.

La Figura 1.4 muestra dos series univariadas y discretas que describen una pulsación del corazón en dos personas diferentes [26]. La variable fue medida mediante un electrodo que registró el impulso eléctrico generado por el corazón a intervalos regulares de tiempo.

### 1.3.2 Análisis de series temporales

El análisis de series temporales comprende métodos tanto matemáticos como estadísticos que ayudan a interpretar datos de este tipo teniendo en cuenta las

correlaciones temporales existentes en los mismos. Existen múltiples objetivos que motivan el análisis de series temporales y están enfocados principalmente tanto a la extracción de información representativa como a la posibilidad de extrapolar y predecir su comportamiento futuro. Estos objetivos [30] son clasificados como:

- **Descripción:** permite mediante la aplicación de técnicas descriptivas definir las principales propiedades de la serie. La más simple consiste en visualizar gráficamente la serie analizada.
- **Explicación:** en ocasiones es posible usar la variación de unas series de tiempo para explicar la variación en otras. Los modelos de regresión múltiple resultan útiles en esta tarea.
- **Predicción:** es uno de los principales objetivos y consiste en predecir los valores futuros de las series analizadas. Resulta de vital importancia en los análisis económicos e industriales.
- **Control:** este se aplica cuando se desea controlar la calidad de determinado proceso y existen múltiples tipos de procedimientos de control. El control incluye poder tomar medidas oportunas frente al proceso que se está controlando.

El análisis clásico de las series temporales comprende el estudio de cuatro componentes básicos presentes en la serie los cuales son la fuente de su variación. Varios métodos tradicionales están relacionados con la descomposición de la serie temporal en sus componentes. Los componentes básicos [30] reconocidos en la literatura son:

- **Tendencia:** es una componente de la serie que refleja la evolución a largo plazo del fenómeno observado.
- **Variación estacional:** es el movimiento periódico de corto plazo. Se trata de una componente causal debida a la influencia de ciertos fenómenos que se repiten de manera periódica en el tiempo.
- **Variación cíclica:** además de la variación estacional, debido a alguna otra causa algunas series exhiben variaciones cada cierto período de tiempo de mayor longitud.

- **Variación aleatoria:** también denominada residuo, no muestra ninguna regularidad y se obtiene una vez eliminadas la tendencia y las variaciones cíclicas de la serie.

Los métodos utilizados en el análisis de series temporales son típicamente divididos en categorías: dominio de la frecuencia [24] y dominio del tiempo [144]. El primero se basa en la función de densidad espectral y el segundo en la función de autocorrelación. Ambos enfoques resultan equivalentes aunque proveen formas alternativas de analizar los procesos que originan las series.

Las técnicas de análisis de series temporales pueden ser divididas además según sus métodos en paramétricas y no paramétricas [24]. Los enfoques paramétricos asumen que la estacionalidad fundamental del proceso estocástico tiene cierta estructura la cual puede ser descrita usando un reducido número de parámetros, por ejemplo los modelos autorregresivos de medias móviles [21, *Autoregressive Integrated Moving-Average* (ARIMA)]. En estos enfoques, el objetivo es estimar los parámetros del modelo que mejor describen el proceso estocástico. Por el contrario, los enfoques no paramétricos estiman explícitamente la covarianza o el espectro del proceso sin asumir que este tenga alguna estructura en particular. Adicionalmente otras clasificaciones han sido creadas para describir los modelos, algunas de ellas son: lineales y no lineales, univariados y multivariados.

### 1.3.3 Minería de datos para series temporales

La minería de datos tiene como objetivo revelar patrones desconocidos a partir de los datos. Su singularidad radica en los tipos de problemas que es capaz de resolver, los cuales incluyen aquellos con enormes conjuntos de datos y relaciones muy complejas entre ellos. Su extensión a problemas con contenido temporal explícito o implícito ha dado lugar a una rama de la minería que ha experimentado un vertiginoso desarrollo.

La minería de datos temporales se encuentra en la intersección de varias disciplinas incluyendo estadística, reconocimiento de patrones temporales, bases de datos temporales y optimización, entre otras. Según la revisión realizada por Lin y colaboradores [99] la minería de datos temporales constituye un paso en el proceso de descubrimiento del conocimiento en conjuntos de datos temporales

y se relaciona con el descubrimiento de patrones temporales. En el trabajo de Povinelli [127] también se hace alusión al concepto de patrones temporales como estructuras que se encuentran potencialmente ocultas en las series temporales. Un patrón temporal puede estar asociado a un evento por lo cual es necesario en la predicción de los mismos.

En el contexto de la minería de datos temporales [52] es una práctica común representar las series como una secuencia ordenada de  $n$  observaciones o puntos  $s = (s_1, s_2, \dots, s_i, \dots, s_n)$ . En series temporales discretas, donde las observaciones son hechas en intervalos regulares de tiempo, es posible omitir la variable  $t_i$ . Esta representación resulta más cercana a la definición de ejemplo de entrenamiento que se introdujo en el epígrafe 1.1. En este punto podemos hacer una analogía entre el valor que toma un caso  $x$  del aprendizaje automático en el  $i$ -ésimo atributo y el valor que toma la serie  $s$  en el  $i$ -ésimo instante de tiempo. La principal diferencia entre ambos radica en la relevancia del orden de los atributos. En los problemas de aprendizaje tradicionales el orden de los atributos es irrelevante y la relación entre ellos es independiente de sus posiciones. Por el contrario, para las series temporales este orden resulta generalmente crucial para determinar las características discriminantes.

Esta particularidad hace que el tratamiento de las series temporales constituya un reto para la minería de datos [59], pues lo hace diferente al de los problemas tradicionales del aprendizaje automático. Entre las características distintivas del dominio temporal se encuentran: alta numerosidad, gran número de dimensiones y una constante actualización de sus datos al transcurrir el tiempo. En el contexto del aprendizaje automático, es imprescindible considerar una serie de tiempo como un todo en lugar de tratarla como un conjunto de campos numéricos individuales.

Las tareas de minería de datos temporales que comúnmente se han enfrentado pueden ser clasificadas en los siguientes grupos [84, 59, 52, 143]:

- **Indexado:** tiene como objetivo, a partir de una serie de interés  $s$  y una medida de similitud, encontrar la serie más cercana a  $s$  en una base de datos temporal.
- **Descubrimiento de patrones y conglomerados:** consiste en descubrir pa-

trones interesantes que pueden aparecer con frecuencia o de forma repentina en las series temporales. En esta tarea es común emplear algoritmos de búsqueda de conglomerados.

- **Clasificación:** tiene como objetivo asignarle etiqueta a una serie a partir de un conjunto de clases previamente definido.
- **Segmentación:** puede ser considerada como un paso previo de preprocesamiento o como una técnica de análisis. Tiene como objetivo, a partir de una serie, obtener un conjunto reducido de segmentos que aproximen la serie original.

## 1.4 Aprendizaje automático en el dominio temporal

El aprendizaje automático para series temporales ha seguido fundamentalmente dos enfoques: la transformación de las series originales a un nuevo espacio de descripción y la adaptación de los clasificadores existentes al dominio temporal. El primero elimina la relación temporal entre los atributos que describen la serie. El segundo se basa principalmente en la utilización de medidas de disimilitud adaptables a las características de las series temporales.

### 1.4.1 Medidas de disimilitud

Las medidas de disimilitud constituyen el núcleo de varios métodos del aprendizaje automático. Dada la naturaleza numérica y continua de las series temporales, el cálculo de la similaridad entre dos series temporales se satisface de forma aproximada, a diferencia de otros tipos de datos donde el concepto de similitud se resuelve de forma exacta. Esto se debe a que resulta prácticamente imposible encontrar dos series exactamente iguales. Otra característica que dificulta la aplicación de las medidas de disimilitud es la presencia de distorsiones en la serie tanto en el dominio del tiempo como en el de los valores. Una práctica extendida consiste en realizar un proceso de normalización de la serie antes de aplicar cualquier medida de disimilitud [134].

Una de las distancias más usadas por su simplicidad y eficiencia es la tradicional distancia euclidiana, que se emplea fundamentalmente en las series temporales después de alguna transformación [84]. Esta medida calcula la diferencia entre cada punto de datos de la serie objetivo respecto a su similar en la serie de referencia (Figura 1.5). Entre los beneficios de utilizar esta medida se encuentra una complejidad computacional de orden lineal. Como consecuencia de las características particulares que poseen las series temporales, varios estudios revelan que no siempre es la distancia indicada para dominios más específicos, pues resulta sensible a los desplazamientos y distorsiones de las series [154]. Una limitación de esta métrica es el requisito de que las series tengan la misma longitud.

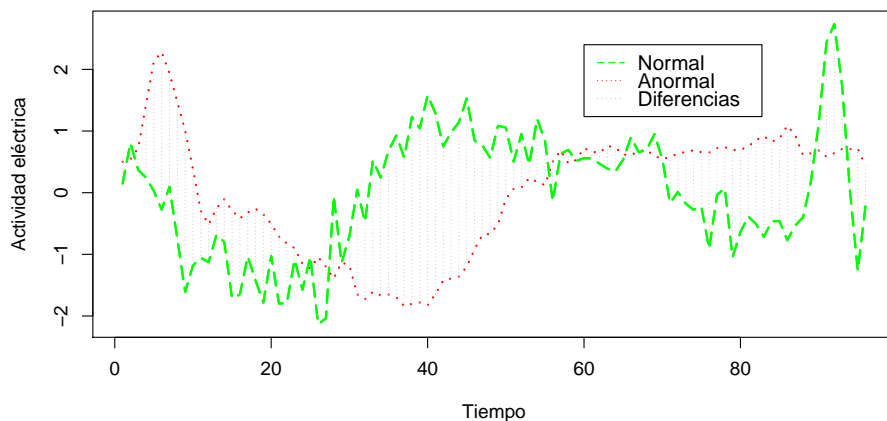


Figura 1.5: Diferencias entre cada punto de datos que se obtienen durante el cálculo de la distancia euclidiana entre las dos series formadas por fragmentos de electrocardiogramas. El valor total de distancia obtenido es de 13.9.

Existen múltiples medidas que se han aplicado para evaluar la disimilitud entre series temporales, las cuales se pueden categorizar en los siguientes grupos:

- **Basadas en rasgos:** este tipo de funciones determina la disimilitud entre dos series utilizando rasgos discriminantes pertenecientes al dominio de la frecuencia. La Transformada Discreta de Fourier es comúnmente utilizada con este fin [123].

- **Basadas en modelos:** este tipo de medidas primeramente ajustan un modelo auto-regresivo, por ejemplo el ARIMA, en las series a comparar. Los parámetros obtenidos, a partir del ajuste de los modelos, son posteriormente utilizados como rasgos discriminatorios [28, 37, 5].
- **Elásticas:** estas medidas basan su funcionamiento en determinar los rasgos discriminantes en el dominio del tiempo. Específicamente, se espera que series de la misma clase compartan determinadas curvas que pueden encontrarse ocultas mediante el ruido o desplazamientos en el tiempo. La característica distintiva de este tipo de medidas es que posibilitan el alineamiento de puntos desfasados en el tiempo durante las comparaciones [141, 32, 106].

Una gran parte de las investigaciones en clasificación de series temporales están basadas en la utilización de medidas elásticas. Una de las más utilizadas se denomina Distorsión Dinámica del Tiempo [141, *Dynamic Time Warping* (DTW)]. Esta medida ha sido utilizada en un gran número de aplicaciones con fines diversos [138, 7, 149, 70]. Con esta técnica no solo se consigue el valor de la disimilitud entre dos series sino que además se obtiene el alineamiento óptimo entre ellas, emparejándolas de forma no lineal mediante contracciones y dilataciones de las series en el eje temporal. Por consiguiente, este emparejamiento permite encontrar regiones equivalentes entre las series que facilitan el cálculo de la disimilitud.

Otra familia de medidas elásticas, conocidas como distancias de edición, también han sido aplicadas para calcular la disimilitud en el dominio temporal. La distancia [32, *Edit Distance on Real sequences* (EDR)] es considerada una extensión, para series temporales, de la distancia original de Levenshtein [95]. La idea es calcular la disimilitud entre series como el costo mínimo de la cantidad de operaciones de edición necesarias para transformar una serie en la otra. La métrica [31, *Edit distance with Real Penalty* (ERP)] es una variante de la distancia anterior, donde se utiliza una constante real para penalizar aquellos valores de las series donde es necesaria una transformación de inserción o eliminación. También son penalizados, de acuerdo a la distancia existente entre ellos, aquellos valores donde se aplica una transformación de reemplazo. La métrica [106, *Time-Warped Edit Distance* (TWED)] es una extensión sumamente interesante de las distancias de edición y de DTW. En esencia puede ser considerada como una combinación

de ambos tipos de medidas.

### Definición y cálculo de DTW

Suponemos que se desea comparar dos series temporales: una serie de prueba  $q = (q_1, \dots, q_m)$  y una series de referencia  $s = (s_1, \dots, s_n)$ . Además, se asume la existencia de una función  $f$  no negativa que expresa el efecto de alinear los puntos  $q_i$  y  $s_i$  de las series:  $d(i, j) = f(q_i, s_j) \geq 0$ . La distancia euclidiana se asume generalmente para este fin. El núcleo de esta técnica<sup>1</sup> consiste en encontrar un camino  $\phi(t) = (\phi_q(t), \phi_s(t))$  de longitud  $T$ , que defina una correspondencia entre los elementos de  $q$  y  $s$ , donde  $\phi_q(t) \in \{1, \dots, m\}$  y  $\phi_s(t) \in \{1, \dots, n\}$ . Dado un camino  $\phi$ , la distancia acumulada de los alineamientos propuestos entre las series  $q$  y  $s$  se calcula según la Expresión 1.1.

$$d_\phi(q, s) = \sum_{t=1}^T d(\phi_q(t), \phi_s(t))m_\phi(t)/M_\phi \quad (1.1)$$

Aquí,  $m_\phi(t)$  es un coeficiente de peso y  $M_\phi$  su constante de normalización correspondiente, asegurando que las distorsiones acumuladas sean comparables entre diferentes caminos. Con el objetivo de encontrar alineamientos razonables se imponen ciertas restricciones como las condiciones de frontera, continuidad y monotonía. A continuación se muestra esta última.

$$\phi_q(t+1) \geq \phi_q(t)$$

$$\phi_s(t+1) \geq \phi_s(t)$$

El cálculo de DTW se basa en encontrar un alineamiento óptimo que garantice una distancia acumulada mínima entre las dos series (Expresión 1.2). En otras palabras, la distorsión que se pretende con el alineamiento es aquella que permita acercar las series  $q$  y  $s$  tanto como sea posible. Atendiendo a su definición, DTW es considerada una pseudo-distancia debido a que no cumple la desigualdad triangular. La Figura 1.6 muestra un ejemplo del cálculo de DTW,

<sup>1</sup>La notación y las expresiones utilizadas en esta sección se tomaron del trabajo de Giorgino [65].



así como los alineamientos que dan como resultado el acumulado de distancia mínima.

$$DTW(q, s) = \min_{\phi} d_{\phi}(q, s) \quad (1.2)$$

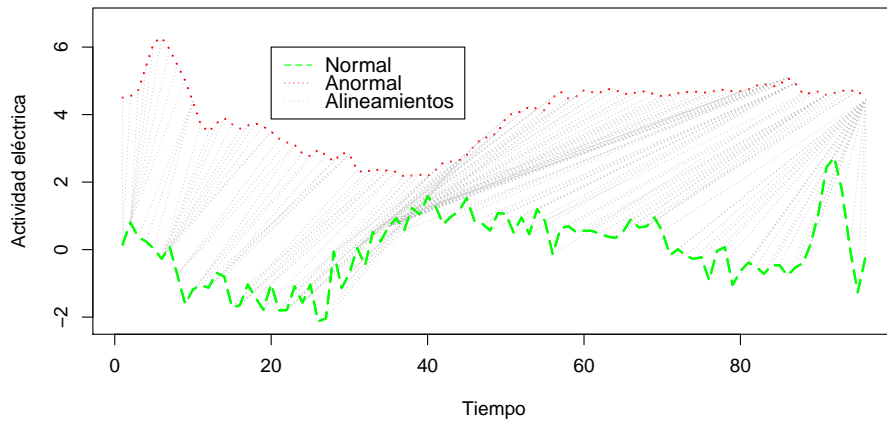


Figura 1.6: Alineamientos obtenidos al aplicar DTW a las dos series formadas por fragmentos de electrocardiogramas que se mostraron anteriormente en la Figura 1.4. El valor de distancia acumulada es de 106.8.

El cálculo de DTW, utilizando programación dinámica, tiene una complejidad máxima temporal y espacial de  $O(n \cdot m)$ . En comparación con la distancia euclidiana, DTW tiene un costo computacional mayor pues depende de forma cuadrática de las longitudes de las series. Debido a que una de las características del dominio temporal es la alta dimensionalidad, varias investigaciones han estado encaminadas a reducir el costo computacional de esta medida. Una de estas soluciones es aplicar restricciones globales. Este tipo de restricciones controla la elasticidad permitida en los alineamientos y por consiguiente la cantidad de celdas a visitar durante la construcción del camino mínimo. La banda de Sakoe-Chiba [141] y el paralelogramo de Itakura [78] son las restricciones globales más utilizadas.

En el caso de la banda de Sakoe-Chiba, se pasa como parámetro un tamaño de ventana  $w$  el cual limita los alineamientos, añadiendo una nueva restricción a

la búsqueda del camino óptimo:  $|\phi_q(t) - \phi_s(t)| \leq w$ . En la Figura 1.7 se muestra el efecto de aplicar esta banda con un tamaño de ventana  $w = 4$ . Aplicar esta restricción afecta la distancia total obtenida entre las dos series. Esta diferencia, lejos de ser perjudicial, aumenta la exactitud de las clasificaciones pues evita alineamientos patológicos que pueden obtenerse si no se limita el ancho de la ventana. El estudio desarrollado por Kurbalija [90] sobre este tema, para varias medidas elásticas incluyendo DTW, muestra que los mejores valores de exactitud se obtienen utilizando tamaños de ventana pequeños. Aplicar esta banda con un tamaño de ventana de cero equivaldría a calcular una distancia similar a la euclidiana.

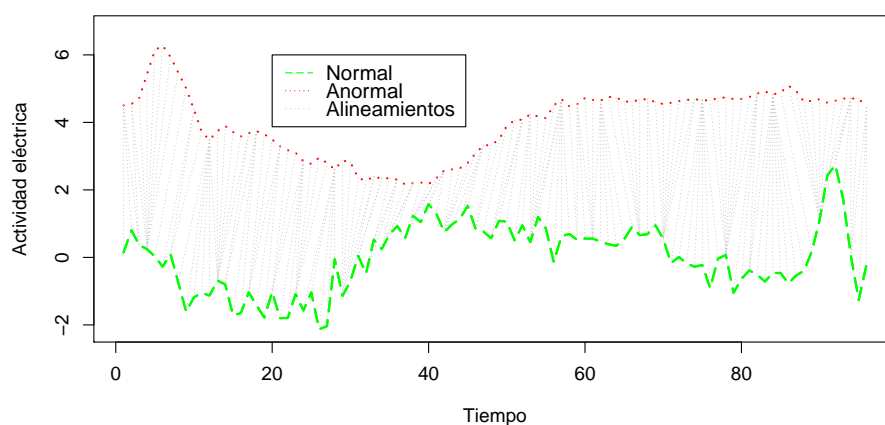


Figura 1.7: Alineamientos obtenidos al aplicar DTW combinada con la banda de Sakoe-Chiba con tamaño de ventana  $w = 4$ . El valor de distancia acumulado es de 154.3.

## 1.4.2 Métodos de clasificación

La clasificación de series temporales es uno de los problemas más abordados de la minería de datos temporales. Al igual que en el aprendizaje supervisado tradicional se parte de un conjunto de entrenamiento formado por ejemplos, que en este caso particular provienen del dominio temporal. A cada serie se encuentra asociada una clase y el objetivo es entrenar una función capaz de predecir la clase correcta para una serie futura.

Considerando las características que presentan las series temporales, la tarea de clasificarlas requiere un tratamiento especial. Un primer grupo de propuestas desarrolladas en este sentido, denotado como enfoque basado en rasgos [28, 12, 158, 41, 61], transforma la serie original a un nuevo espacio de descripción donde los clasificadores convencionales pueden ser aplicados. Para extraer los rasgos de la serie original se utilizan comúnmente herramientas del procesamiento de señales o estadísticas. Este enfoque está estrechamente relacionado con las medidas basadas en rasgos y las medidas basadas en modelos que se describieron en la sección 1.4.1. Un segundo grupo de propuestas [139, 126, 49, 138, 162, 83] se enfoca en adaptar o desarrollar clasificadores especialmente diseñados para tratar series temporales. Esta categoría se basa principalmente en la selección de una representación apropiada de las series, acompañada de una medida adecuada para calcular la disimilitud entre las mismas, por ejemplo las medidas elásticas. Esta categoría incluye además el enfoque basado en casos, el cual ha recibido gran atención en la literatura especializada.

### ***k*NN**

El clasificador *k*NN es la opción estándar que se utiliza comúnmente para evaluar la efectividad de las medidas de disimilaridad propuestas para comparar series temporales. Específicamente, se utiliza el 1NN por las múltiples ventajas que aporta [142, 154]. En primer lugar, la exactitud de este clasificador depende directamente de la medida de disimilaridad que se utilice. Además, resulta sencillo de implementar y no necesita ajustar parámetros adicionales. De manera general, en el dominio temporal este clasificador ha obtenido muy buenos resultados [162, 125].

### **DT**

La construcción de DT es otro de los enfoques que se han aplicado para la clasificación de series temporales. En la propuesta desarrollada por Yamada y colaboradores [164] se proponen dos test binarios denominados *standard-example* y *cluster-example*. El primero selecciona, mediante una búsqueda exhaustiva, una serie de referencia en el conjunto de entrenamiento que maximice la ganancia de

pureza de la división en cada nodo. Luego los ejemplos restantes se distribuyen en cada uno de los hijos del nodo en dependencia de su distancia respecto al ejemplo seleccionado. El hijo derecho se compone de las series que se encuentran a una distancia de la serie de referencia menor o igual que un umbral dado. El hijo izquierdo se conforma de los ejemplos restantes. El segundo test busca, en lugar de una, dos series de referencia de manera exhaustiva. Dichas series de referencia se asocian a cada nodo, determinando la distribución que siguen los ejemplos de entrenamiento que llegan a dicho nodo. Estos se dividen a la izquierda o la derecha en dependencia de la serie de referencia que resulte menos distante a cada ejemplo. Igualmente, la ganancia en pureza es usada para escoger el par de series de referencia. Esta segunda variante evita la necesidad de especificar un umbral. La medida empleada para obtener la cercanía entre las series es DTW.

Una idea similar se emplea en la propuesta de Balakrishnan y Madigan [6], donde se utiliza el algoritmo de descubrimiento de conglomerados *k-means* para seleccionar las dos series de referencia. Este algoritmo garantiza una partición que optimice el agrupamiento pero no así la pureza del nodo. Para aliviar esta situación el algoritmo se ejecuta varias veces y se selecciona la ejecución de mayor pureza siguiendo el criterio del índice de Gini [23]. En esta propuesta se mantiene el uso de DTW como medida de disimilitud.

La última propuesta analizada, que también utiliza series de referencia en la construcción del árbol de decisión, es desarrollada por Douzal-Chouakria y Amblard [49]. El criterio de división que se utiliza en cada nodo hace uso de una métrica adaptativa que permite cubrir tanto el comportamiento como los valores de proximidad entre las series. Esto se logra mediante la combinación de una distancia basada en valor y un costo basado en comportamiento. Para medir la proximidad respecto a los valores se utiliza la distancia euclidiana y DTW. El costo basado en comportamiento se estimó mediante la Correlación de Pearson.

## SVMs

Las SVMs es otra técnica que ha sido ampliamente aplicada a la clasificación de series temporales. La propuesta de Lei y Sun [92] consiste en una variante del núcleo *Gaussian Radial Basis Function* (GRBF) lograda mediante la sustitución de la distancia euclidiana por DTW (GDTW). Los resultados experimentales

indican que GDTW no supera al clasificador 1NN con DTW. La causa de este comportamiento se ha analizado en varios trabajos [169, 92] y está relacionada con las características del núcleo GDTW. Las funciones núcleo deben satisfacer el Teorema de Mercer [110] que establece como núcleos admisibles aquellas funciones que sean semi-definidas positivas (*Positive Definite Symmetric*, PDS). El uso de núcleos PDS garantiza que la matriz del núcleo sea convexa y la solución única. Se ha demostrado que el núcleo GDTW no cumple la propiedad PDS [92].

El trabajo de Zhang y colaboradores [169] asume como causa de la violación de la propiedad PDS el hecho de que DTW no es una métrica. Por tanto este enfoque propone núcleos basados en las métricas elásticas ERP y TWED. Aunque en los experimentos realizados no se observó ninguna violación de la propiedad PDS, posteriormente se han publicado ejemplos donde estas métricas incumplen dicha propiedad [105]. Como solución a este problema, Marteau y Gibet [105] proponen un método para crear un tipo de núcleo a partir de las medidas elásticas. Dicho método permite construir núcleos con la propiedad PDS si se cumplen ciertas condiciones. Las condiciones impuestas son más débiles que las propuestas en otros trabajos aunque necesitan la introducción de un término de regularización para obtener la prueba de la propiedad PDS.

Una idea diferente para mejorar la clasificación de las SVMs se basa en la transformación de los límites de decisión entre las clases [68]. La idea se soporta en que las múltiples variaciones que puede experimentar el patrón de una clase no están cubiertas por los ejemplo de entrenamiento. Con el objetivo de suplir la falta de ejemplos, para mejorar la generalización de las SVMs, se crean ejemplos nuevos que son insertados en el conjunto de entrenamiento.

El estudio desarrollado por Pree y colaboradores [128] compara múltiples medidas de similitud utilizadas como funciones de núcleo para las SVMs, incluyendo la distancia euclidiana, DTW y TWED. En este estudio no se demuestra el cumplimiento de la propiedad PDS. La distancia TWED obtiene los mejores resultados y es de destacar que estos no superan significativamente a los obtenidos con la distancia euclidiana. Los resultados con DTW sí son superados por TWED aunque no así por la euclidiana.

Para la clasificación de series temporales se ha propuesto también la construcción de un núcleo [80] basado en una variante pesada de DTW [81, *Weighted*

*Dynamic Time Warping*]. Esta variante penaliza los alineamientos obtenidos en dependencia del desfase existente entre los puntos alineados.

## Otros métodos de clasificación supervisada

Rodríguez y colaboradores [139] proponen un método de clasificación de series de tiempo multivariadas mediante la programación lógica inductiva. En este trabajo se introducen dos tipos de predicados: basados en intervalo y basados en distancia. Las redes neuronales artificiales, específicamente el *perceptron* multicapa, también han sido empleadas en esta tarea [116]. Por otra parte, el trabajo desarrollado por Povinelli y colaboradores [126] presenta una investigación sobre la clasificación de señales basándose en el modelado de un sistema dinámico que captura los datos para la serie usando modelos de mixtura Gaussianos. Incluso métodos bioinspirados, como los basados en colonias de hormigas [17] y los enjambres de partículas [152], se han utilizado en la optimización de modelos de clasificación para datos temporales.

### 1.4.3 Métodos de agrupamiento

Los métodos de agrupamiento han tenido una amplia aplicación en el dominio temporal [52]. Estos se han utilizado en conjuntos de series completas con el objetivo de obtener conglomerados donde las series sean similares entre sí. Este enfoque incluye el empleo de una medida apropiada de distancia o la transformación de la serie original mediante la extracción de características [122, 108, 145, 37].

Los métodos de agrupamiento también se han utilizado en la extracción de subsecuencias de una o varias series. Esta aplicación tiene como objetivo el agrupamiento de subsecuencias con patrones comunes que se encuentren dispersas a lo largo de la serie [121, 114, 115, 137, 117].

### 1.4.4 Métodos de clasificación semi-supervisada

En la clasificación de series temporales resulta habitual asumir la existencia de grandes volúmenes de ejemplos etiquetados. El creciente uso de sensores de todo tipo ha facilitado la generación de grandes cantidades de datos temporales.

Por el contrario, la obtención de etiquetas asociadas a ese tipo de datos resulta en ocasiones un proceso costoso que requiere del tiempo y el conocimiento de expertos en determinados dominios, por ejemplo de cardiólogos, economistas, controladores de procesos fabriles y astrónomos, entre otros. La disponibilidad del repositorio UCR [86], donde se encuentran disponibles decenas de conjuntos de datos temporales, ha mantenido a la comunidad científica de esta área al margen de la dificultad que, en la práctica, representa la obtención de series temporales etiquetadas.

La aplicación de métodos de SSC en el dominio temporal permiten mejorar la exactitud de la clasificación al tener en cuenta durante el entrenamiento tanto los ejemplos etiquetados como los no etiquetados. Esta área del aprendizaje automático ha recibido una gran atención durante la pasada década, lo cual se evidencia en las variadas aplicaciones que ha encontrado como: la clasificación de insectos voladores [9], extracción de información en la web [54] y la predicción de precipitaciones [2] entre otras.

### **Clasificación mediante métodos de auto-etiquetado**

El método Self-training, debido a su simplicidad, es uno de los más utilizados en el dominio temporal. El trabajo propuesto por Zhong [170] aplica una variante de este método para clasificar series formadas por las mediciones secuenciales de la actividad eléctrica del cerebro, las cuales se conocen como electroencefalogramas (EEG). Como esquema supervisado se utiliza los modelos ocultos de Markov [133, *Hidden Markov Models* (HMMs)]. El Self-training también ha sido extensamente utilizado en un caso específico de SSC donde sólo se encuentran disponibles, durante el aprendizaje, ejemplos pertenecientes a una única clase [11, 33, 136, 157]. En este tipo de problemas resulta común utilizar como esquema de aprendizaje el algoritmo  $k$ NN debido a su probada efectividad en la clasificación de series temporales.

El método Co-training, como se describió en la sección 1.2.4, requiere para su funcionamiento de dos vistas condicionalmente independientes sobre los datos del problema. Este requisito es normalmente muy restrictivo y difícil de satisfacer para las series temporales donde en la mayoría de los casos las observaciones cercanas entre sí se encuentran correlacionadas. El trabajo de Meng y colabora-

dores [109] aplica una variante de Co-training [67] que utiliza dos clasificadores diferentes en lugar de las dos vistas independientes clásicas. En este caso los dos clasificadores utilizados son los HMMs y el 1NN. Por otra parte, en el trabajo de Liu y colaboradores [102] se entrenan dos clasificadores de tipo *AdaBoost* [57] utilizando un esquema Co-training para el problema de la clasificación de series formadas por métricas de calidad, las cuales son estimadas en secuencias de imágenes captadas por cámaras en movimiento.

### Otros métodos de clasificación

Múltiples enfoques de SSC han sido aplicados a la clasificación de series temporales. La factibilidad de aplicar métodos basados en grafos ha sido abordada en varios trabajos [43, 44]. Por otra parte, el método S3VM también ha sido extendido al manejo con datos temporales [87].

Otro enfoque muy utilizado en el dominio temporal, denominado *cluster-then-label*, combina las tareas de obtención de conglomerados y clasificación. Este método está estrechamente relacionado con los modelos generativos. La diferencia consiste en la utilización de un paso previo de descubrimiento de conglomerados en lugar de utilizar un modelo probabilístico. Posteriormente a cada conglomerado se le asigna una etiqueta con la ayuda de los ejemplos inicialmente etiquetados. El trabajo de Marussy y Buza [107] utiliza este enfoque mediante un método de obtención de conglomerados jerárquico con restricciones. Un enfoque similar se aplica en el trabajo desarrollado por Frank y colaboradores [55] donde además se propone una nueva medida de similitud para series de tiempo.

## 1.5 Sumario

En este capítulo se han presentado las bases teóricas que hacen posible comprender la investigación realizada. A partir del estado de la literatura presentado se pueden resaltar los siguientes aspectos:

- Las series de tiempo representan de manera natural fenómenos presentes en prácticamente todas las áreas del conocimiento. La existencia de un orden



temporal entre los puntos de datos que forman una serie hace que su análisis sea diferente al de otros problemas que no presentan esta característica.

- La consolidación de la minería de datos temporales ha abierto nuevas posibilidades en el análisis de las series de tiempo, propiciando la aplicación de métodos del aprendizaje automático en tareas clásicas como la obtención de conglomerados y la clasificación.
- Las características especiales que presentan las series temporales, como la dependencia temporal entre los puntos de datos, la alta dimensionalidad y numerosidad entre otras, diferencian su tratamiento en comparación con el de los problemas tradicionales del aprendizaje automático.
- La clasificación de series temporales ha atraído una gran atención de la comunidad científica. Pero la obtención de una cantidad suficiente de ejemplos etiquetados constituye, en ocasiones, una limitante para efectuar el entrenamiento automático de los modelos de clasificación. Aunque el uso generalizado de una amplia gama de sensores ha propiciado la obtención de grandes volúmenes de datos temporales, obtener las etiquetas resulta, en muchos casos, un proceso sumamente costoso.
- El aprendizaje de un modelo de clasificación, en presencia de grandes volúmenes de datos sin etiquetar de conjunto con un reducido número de ejemplos etiquetados, hace de la SSC una opción factible en el dominio temporal.
- Entre las múltiples técnicas de SSC existentes, las de auto-etiquetado se destacan por características atractivas como: simplicidad, versatilidad (la mayoría siguen una metodología de envoltorio que permite combinarlas con prácticamente cualquier esquema de aprendizaje supervisado) y que, a diferencia de otras técnicas, no asumen supuestos específicos relacionados con la distribución de los datos de entrada.
- En el dominio temporal sólo se han empleado para la clasificación de series temporales los métodos de auto-etiquetado más populares: Self-training y Co-training. Considerando los beneficios que brindan este tipo de métodos resulta necesario profundizar su estudio en el área específica de las series temporales.



## Capítulo 2

# Estudio sobre métodos de auto-etiquetado aplicados al dominio temporal

Los métodos de auto-etiquetado conforman una familia cuya característica común consiste en extender el conjunto original de ejemplos a partir de las predicciones más confiables que se realizan para clasificar los ejemplos no etiquetados. A diferencia de otras técnicas de SSC, las de auto-etiquetado no asumen supuestos específicos sobre la distribución de los datos de entrada. Esto, entre otras características, hace de esta familia una opción prometedora a la problemática de la clasificación de series temporales.

A pesar de los beneficios que presentan este tipo de técnicas, su aplicación en el dominio temporal se ha reducido fundamentalmente a dos de las más populares: Self-training y Co-training. Por ello, este capítulo tiene como objetivo hacer extensivo el estudio a otros métodos, de manera que se puedan alcanzar conclusiones más detalladas sobre este tópico. Además, se pretende explorar el uso de otros esquemas de aprendizaje, en adición al clásico 1NN, para la clasificación de series temporales.

El capítulo se organiza en los siguientes apartados. En la sección [2.1](#) se detallan los métodos semi-supervisados involucrados en el estudio así como los

supervisados que serán utilizados como clasificadores base. En la sección 2.2 se describen los detalles de la experimentación incluyendo: los conjuntos de datos seleccionados, los parámetros prefijados para los distintos algoritmos utilizados y las medidas empleadas para la evaluación de los resultados. En la sección 2.3 se realiza un análisis comparativo de los resultados, tanto transductivos como inductivos, que se obtuvieron a partir de la experimentación. Por último, en la sección 2.4 se presenta un breve resumen con las principales conclusiones del capítulo.

## 2.1 Métodos seleccionados para el estudio experimental

En la descripción ofrecida en el Capítulo 1 se destacan los principales aspectos que conforman los métodos de auto-etiquetado. Muchos de los métodos propuestos en la literatura constituyen variantes de los métodos Self-training y Co-training estándar. De este último, considerando que la dependencia temporal existente entre los puntos de datos que conforman una serie viola el supuesto de independencia entre las vistas que asume el Co-training, se ha decidido no incluirlo en su forma estándar. Teniendo en cuenta que cada método presenta sus propias ventajas y desventajas, para este estudio se ha seleccionado una muestra representativa de los mismos. La selección que presentamos está basada en los resultados obtenidos en el estudio precedente efectuado por Triguero y colaboradores [150], el cual se desarrolló sobre problemas clásicos del aprendizaje automático. Los métodos de auto-etiquetado incluidos en el presente estudio, los cuales fueron descritos con anterioridad en la sección 1.2.4, son:

- Self-training [165]
- SETRED [96]
- SNNRCE [155]
- Tri-training [172]
- Democratic [171]

En la Tabla 2.1 se muestra un resumen con las principales características de los métodos de auto-etiquetado seleccionados, las cuales incluyen: nombre abreviado del método, mecanismo de adición, modelo de clasificación respecto a la utilización de uno o más clasificadores, esquema de aprendizaje respecto a la utilización de uno o varios esquemas diferentes y respecto a cuál elemento se estableció el criterio de parada.

Nombre abreviado	Mecanismo de adición	Modelo de Clasificación	Esquema de aprendizaje	Criterio de parada
SelfT	incremental	Monoclasificador	Monoaprendizaje	Num. iteraciones
SETRED	modificable	Monoclasificador	Monoaprendizaje	Num. iteraciones
SNNRCE	modificable	Monoclasificador	Monoaprendizaje	Tamaño de $U$
TriT	incremental	Multiclasificador	Monoaprendizaje	Camb. hipótesis
Democratic	incremental	Multiclasificador	Multiaprendizaje	Camb. hipótesis

Tabla 2.1: Principales características de los métodos de auto-etiquetado seleccionados.

La mayoría de los métodos de auto-etiquetado siguen una metodología de envoltorio donde resulta fundamental la selección del esquema de aprendizaje que será utilizado. A partir de la revisión realizada en el Capítulo 1, sobre los métodos de clasificación propuestos en el dominio temporal (sección 1.4.2), se ha efectuado una selección representativa de los enfoques más utilizados del aprendizaje automático. Atendiendo a la misma, los métodos seleccionados como clasificadores base son los siguientes:

- $k$ NN
- DT
- SVM

Los clasificadores seleccionados han sido ampliamente estudiados en el dominio temporal. Su uso específico como clasificadores base en métodos de auto-etiquetado ha sido poco abordado en la literatura, siendo el  $k$ NN el más utilizado. Una de las ventajas que presentan los métodos seleccionados es que siguen un

enfoque basado en el cálculo de la disimilitud<sup>1</sup> entre los ejemplos. Esto hace factible ampliar el estudio mediante la utilización de varias medidas de disimilitud.

## 2.2 Marco experimental

Esta sección presenta toda la información relacionada con el diseño de los experimentos. Inicialmente, se presenta la información básica de los conjuntos de datos involucrados en este estudio. Luego se ofrece una breve descripción de las medidas de desempeño utilizadas para evaluar el aprendizaje. Finalmente se describe la configuración los parámetros de de todos los algoritmos involucrados en el estudio.

### 2.2.1 Conjuntos de datos experimentales

La experimentación se basa en 35 conjuntos de datos temporales provenientes del repositorio público de la Universidad de California Riverside (UCR) [34] con excepción del conjunto de datos ECG que proviene de otro repositorio [156]. La Tabla 2.2 muestra un resumen con las principales características de los conjuntos de datos seleccionados. Estos contienen entre 56 y 9236 ejemplos, la longitud de las series varía en el rango de 24 a 1882 puntos de datos y el número de clases se encuentra entre 2 y 14.

En cada conjunto de datos las series fueron *z-normalizadas* (*z-normalized*), siguiendo la recomendación ofrecida en el trabajo de Rakthanmanon y colaboradores [134]. Esta transformación se realiza aplicando la Expresión 2.1 a cada punto  $s_i$  de la serie  $s$  presente en el conjunto de datos, siendo  $n$  la longitud de la misma.

$$s'_i = \frac{s_i - \mu(s)}{\sigma(s)}, \forall i = 1, \dots, n \quad (2.1)$$

Las expresiones  $\mu$  y  $\sigma$  hacen referencia a la media y la desviación estándar de  $s$ , respectivamente. Como resultado se obtiene una serie  $s'$  con media cercana

---

<sup>1</sup>Las propuestas realizadas en el dominio temporal, a diferencia de los métodos clásicos de construcción de DT que se basan en los atributos, realizan la construcción de los árboles en base a la disimilitud calculada entre los ejemplos de entrenamiento

a cero y desviación estándar próxima a uno, lo cual permite una comparación más acertada entre series que presentan amplitudes diferentes.

Conjunto de datos	Tamaño del conjunto	Longitud	Cantidad de clases
CBF	930	128	3
Chlorine	4307	166	3
CinCECGt	1420	1639	4
Coffee	56	286	2
CricketX	780	300	12
CricketY	780	300	12
CricketZ	780	300	12
ECG	2026	85	2
ECGFiveD	884	136	2
FaceAll	2250	131	14
FacesUCR	2250	131	14
Fish	350	463	7
GunPoint	200	150	2
Haptics	463	1092	5
InlineS	650	1882	7
Italy	1096	24	2
Lighting2	121	637	2
Lighting7	143	319	7
MALLAT	2400	1024	8
MedicalI	1141	99	10
MoteStrain	1272	84	2
Olive	60	570	4
OSULeaf	442	427	6
Sony	621	70	2
SonyII	980	65	2
StarLightC	9236	1024	3
Synthetic	600	60	6
Trace	200	275	4
Two	5000	128	4
TwoLeadECG	1162	82	2
uWaveGLX	4478	315	8
uWaveGLY	4478	315	8
uWaveGLZ	4477	315	8
Wafer	7164	152	2
Yoga	3300	426	2

Tabla 2.2: Descripción detallada de los conjuntos de datos temporales.

Los conjuntos de datos fueron divididos de forma aleatoria en cinco particiones con el objetivo de aplicar validación cruzada. Posteriormente, cada partición de entrenamiento (4/5 del conjunto total de ejemplos) fue dividida de manera aleatoria en dos conjuntos:  $L$  y  $U$ , de ejemplos etiquetados y no etiquetados. En

el caso del conjunto  $U$  las etiquetas son removidas de manera que resulten inaccesibles a los algoritmos durante el entrenamiento. No se ha intentado mantener en los conjuntos resultantes  $L$  y  $U$  la misma proporción por clases observada en el conjunto de entrenamiento total, siguiendo el enfoque sugerido por varios autores [150, 155]. No obstante, sí se ha garantizado que todas las clases del problema se encuentren representadas en el conjunto  $L$ .

Para subdividir el conjunto de entrenamiento se tomaron diferentes proporciones de ejemplos etiquetados iniciales con el propósito de estudiar la influencia que ejerce dicha cantidad. Específicamente, para formar el conjunto  $L$  se utilizaron tres proporciones diferentes respecto a la cantidad disponible de ejemplos: 10 %, 20 % y 30 %. Por ejemplo, si se cuenta con una partición de entrenamiento que contiene 100 ejemplos, cuando la proporción de datos etiquetados es de 10 %, sólo 10 ejemplos con sus respectivas etiquetas forman el conjunto  $L$  mientras los restantes 90 son colocados en  $U$  sin sus etiquetas. La partición de prueba, conformada por 25 ejemplos es mantenida aparte con el objetivo de evaluar la capacidad inductiva de la hipótesis aprendida.

## 2.2.2 Medidas de evaluación

Para evaluar la efectividad de la clasificación realizada por las técnicas de auto-etiquetado se utilizan dos estadísticos clásicos del aprendizaje automático: exactitud (*accuracy*) [160] y la kappa de Cohen [13]. Ambas medidas se describen brevemente a continuación:

- Exactitud: Esta medida refleja el consenso entre las clases observadas y las predichas. Esta simple medida es una de las más utilizadas para evaluar el comportamiento de los clasificadores.

$$\text{Exactitud} = \frac{\text{Cantidad de ejemplos correctamente clasificados}}{\text{Cantidad de ejemplos evaluados}} \quad (2.2)$$

- Kappa de Cohen: Esta medida tiene en cuenta los aciertos que el clasificador obtiene por pura aleatoriedad. Toma valores en el rango de  $[-1, 1]$ , donde el valor cero significa que no existe correspondencia entre las clases predichas y las observadas, uno significa concordancia total, mientras que los valores negativos indican que las clases predichas son opuestas a las observadas.



### 2.2.3 Consideraciones sobre los algoritmos

En esta sección se especifica la configuración de los parámetros presentes en los métodos involucrados en este estudio. Los valores seleccionados se utilizan de manera uniforme para todos los conjuntos de datos y han sido seleccionados en su mayoría teniendo en cuenta las recomendaciones ofrecidas en trabajos previos. Los parámetros de los métodos de auto-etiquetado no se optimizan para cada conjunto de datos en específico ya que el principal propósito de este estudio experimental es comparar el comportamiento general de este tipo de métodos. La configuración de los parámetros se muestra en la Tabla 2.3. En adición, el Apéndice A incluye información detallada sobre el significado de cada parámetro presente en las técnicas de auto-etiquetado.

Algoritmo	Parámetros
$k$ NN	$k = 1$ , distancia = euclidiana y DTW
DT	Número mínimo de ejemplos por hoja = 3, nivel máximo de impureza permitido por hoja = 0.1 influencia del comportamiento en la medida de disimilitud= 2.0, distancia = euclidiana y DTW
SVM	Tipo de núcleo = <i>GRBF</i> , $C = \{2^{-5}, 2^{-4}, \dots, 1, \dots, 2^5\}$ , $\sigma = \{2^{-5}, 2^{-4}, \dots, 1, \dots, 2^5\}$ , selección mediante validación cruzada de 3 particiones, distancia = euclidiana y DTW
SelfT	Máximo de iteraciones = $\min\{50, \lceil 0.7 \cdot  U  / InstPerIter \rceil\}$
SETRED	Máximo de iteraciones = $\min\{50, \lceil 0.7 \cdot  U  / InstPerIter \rceil\}$ , umbral de significación $\theta = 0.1$
SNNRCE	Umbral de significación $\alpha = 0.1$
TriT	No se especifican parámetros
Democratic	Clasificadores = 1NN, DT y SVM

Tabla 2.3: Especificación de los parámetros utilizados por los clasificadores base y los métodos de auto-etiquetado involucrados en la experimentación.

Para los métodos de auto-etiquetado que tienen como criterio de parada una cantidad fija de iteraciones, esta se definió a partir de la combinación de dos criterios de parada:

- El 70 % de los ejemplos en el conjunto inicial  $U$  ha sido removido e insertado en  $L$ .
- El algoritmo ha alcanzado un número máximo de 50 iteraciones.

De esta forma el algoritmo se detiene cuando se cumple el primero de estos criterios, siendo  $InstPerIter$  el número de ejemplos que son removidos de  $U$  en cada iteración. Estas cantidades son calculadas para cada conjunto de datos en dependencia de la distribución por clase observada en el conjunto  $L$  inicial. El criterio de parada así definido facilita la explotación del conjunto  $U$  y a su vez evita las salidas extremas del clasificador base, causadas por la adición en  $L$  de todos los ejemplos no etiquetados disponibles en  $U$ .

Muchos de los métodos de auto-etiquetado incluyen uno o varios clasificadores base. Para los métodos que soportan clasificadores base con diferentes paradigmas de aprendizaje (SelfT y TriT), se han explorado todas las combinaciones posibles. A partir de los tres paradigmas de clasificación seleccionados ( $k$ NN, DT y SVM) se definen los tres algoritmos representativos de cada enfoque que serán utilizados como clasificadores base en el dominio temporal:

- $k$ NN: de este enfoque se seleccionó el 1NN como clasificador base considerando los múltiples estudios en el dominio temporal [163, 90, 142, 64] que están basados en este tipo de clasificador.
- DT: el método propuesto por Douzal-Chouakria y Amblard [49] es el seleccionado para construir árboles específicamente diseñados para clasificar series temporales. Este método obtiene resultados competitivos y el criterio de división utilizado para ramificar los nodos es lo suficientemente flexible como para cubrir tanto el comportamiento como los valores de distancia presentes en las series a comparar.
- SVM: la función de núcleo seleccionada es la GRBF que se define según la Expresión 2.3. La mayoría de los estudios precedentes [169, 128, 105] lo utilizan en combinación con una medida de distancia  $d$  utilizada en el dominio temporal. Siguiendo la metodología presentada en el trabajo de Marteau y Gibet [105] se ha normalizado la matriz de distancia (contiene las distancias entre todos los pares de ejemplos) durante la etapa de entrenamiento

para limitar el espacio de búsqueda de los parámetros involucrados. Específicamente, se ha utilizado un conjunto predefinido de valores para los parámetros  $C$  y  $\sigma$  (Tabla 2.3), de forma que se seleccionen los más apropiados mediante el proceso de validación cruzada.

$$K_d(x_i, x_j) = \exp(-d(x_i, x_j)^2 / (2\sigma^2)) \quad (2.3)$$

Durante la experimentación se evalúan dos medidas de disimilitud diferentes para comparar las series: euclidiana y DTW. La banda de Sakoe-Chiba es utilizada como restricción global para el cálculo de DTW. Los resultados experimentales de Kurbalija y colaboradores [90] muestran que un tamaño de ventana, que representa el cuatro por ciento de la longitud de la serie, resulta como promedio el mejor valor respecto a exactitud de la clasificación. Por lo tanto, este es el valor seleccionado para evaluar la medida DTW en todos los conjuntos de datos.

## 2.3 Resultados experimentales

Esta sección presenta los resultados experimentales obtenidos y una detallada discusión de los mismos. El comportamiento de los métodos de auto-etiquetado se evalúa a partir de sus habilidades transductivas e inductivas, utilizando tres porcentajes diferentes de ejemplos inicialmente etiquetados. También se realiza un análisis de los tiempos de cómputo consumidos por los métodos durante su entrenamiento y evaluación. Al finalizar la sección se presenta una comparación entre el paradigma supervisado y el semi-supervisado. Adicionalmente, el Apéndice B incluye los resultados completos de las evaluaciones realizadas con cada una de las técnicas evaluadas.

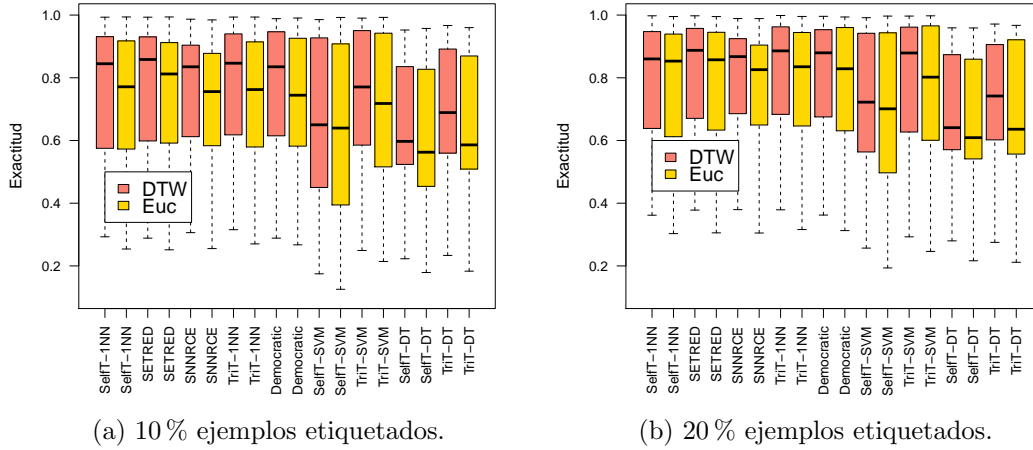
Para contrastar los resultados se utilizan pruebas estadísticas no paramétricas, siguiendo la metodología descrita por García y colaboradores [62]. Concretamente se utiliza la prueba alineada de Friedman [75] para múltiples comparaciones con el objetivo de detectar diferencias significativas en los métodos evaluados. Posteriormente se aplica la prueba pareada de Hochberg [74] para caracterizar las diferencias en caso de que existan.

### 2.3.1 Resultados transductivos

Según se especificó en el Capítulo 1 el objetivo principal de la clasificación transductiva consiste en predecir la clase de los ejemplos no etiquetados usados durante el entrenamiento. La Tabla 2.4 presenta los resultados de exactitud promedio (**Exact.**) obtenidos por los métodos de auto-etiquetado sobre los 35 conjuntos de datos utilizados con 10 %, 20 % y 30 % de ejemplos etiquetados. Los métodos se muestran en orden descendente de exactitud. Para los métodos que soportan diferentes clasificadores base se han explorado todas las combinaciones, especificando el nombre del clasificador utilizado a continuación del nombre del método semi-supervisado. Los resultados completos se muestran en la sección B.1.

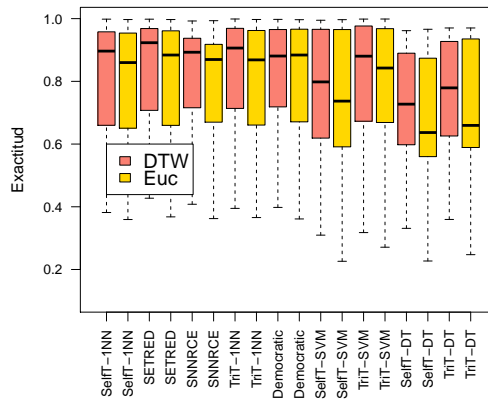
La diferencia entre el ordenamiento utilizando los resultados de exactitud y el obtenido con el estadístico kappa, se denota como  $\Delta K$ . Para esta diferencia, los valores positivos indican que el método obtiene una mejor posición cuando se ordena utilizando la kappa, los valores negativos indican el caso contrario y cero significa que no existen diferencias entre los dos ordenamientos.  $\Delta K$  permite identificar aquellos métodos que se benefician de los aciertos que ocurren al azar. Según la información mostrada en la Tabla 2.4 no existen diferencias significativas entre los ordenamientos ya que solamente algunos métodos exhiben valores de  $\Delta K$  diferentes a cero.

La Figura 2.1 muestra mediante un diagrama de cajas (*box-plot*) la exactitud obtenida por los métodos tanto para la distancia euclidiana como para DTW. En este tipo de diagrama la parte inferior y superior del rectángulo representan el primer y tercer cuartil, respectivamente. La banda dentro del rectángulo representa la mediana. Esta figura permite visualizar de una manera más detallada el comportamiento particular de cada método. Es posible apreciar la ganancia de exactitud que se obtiene al utilizar DTW en comparación con la distancia euclidiana. Por otra parte los métodos que se combinan con el 1NN como clasificador base exhiben los mejores resultados. Por el contrario, los valores de exactitud más bajos se obtienen utilizando el clasificador DT. Adicionalmente, las SVMs como clasificador base causan un comportamiento disperso de los valores de exactitud. La Figura 2.2 muestra el efecto que provoca el aumento de los ejemplos etiquetados en el promedio total de exactitud por cada método.



(a) 10 % ejemplos etiquetados.

(b) 20 % ejemplos etiquetados.



(c) 30 % ejemplos etiquetados.

Figura 2.1: Diagrama de cajas que muestra los resultados de exactitud para diferentes proporciones de ejemplos etiquetados y las medidas euclidiana y DTW.

A partir de estos resultados se ha efectuado una comparación entre los métodos de tipo monoaprendizaje, agrupados por cada esquema de aprendizaje diferente. Esta primera comparación permite establecer los métodos de auto-etiquetado más exitosos para cada clasificador base. La prueba alineada de Friedman, aplicada a la exactitud en el conjunto de métodos que emplean el 1NN como esquema de aprendizaje base, detecta diferencias significativas para un nivel de significación de  $\alpha = 0.05$  en todas las comparaciones efectuadas.

10 %			20 %			30 %		
Euclidiana	Exact.	$\Delta K$	Euclidiana	Exact.	$\Delta K$	Euclidiana	Exact.	$\Delta K$
SETRED	0.719	-1	SETRED	0.762	0	SETRED	0.793	0
TriT-1NN	0.718	+1	Democratic	0.760	0	Democratic	0.793	0
Democratic	0.715	0	TriT-1NN	0.759	0	TriT-1NN	0.790	0
SelfT-1NN	0.708	0	SelfT-1NN	0.750	0	SNNRCE	0.778	-2
SNNRCE	0.701	0	SNNRCE	0.742	0	TriT-SVM	0.777	+1
TriT-SVM	0.694	0	TriT-SVM	0.734	0	SelfT-1NN	0.776	+1
TriT-DT	0.635	0	SelfT-SVM	0.686	0	SelfT-SVM	0.721	0
SelfT-SVM	0.618	0	TriT-DT	0.671	0	TriT-DT	0.694	0
SelfT-DT	0.598	0	SelfT-DT	0.639	0	SelfT-DT	0.664	0
DTW	Exact.	$\Delta K$	DTW	Exact.	$\Delta K$	DTW	Exact.	$\Delta K$
TriT-1NN	0.771	0	TriT-1NN	0.815	0	TriT-1NN	0.840	0
SETRED	0.768	0	SETRED	0.813	0	SETRED	0.839	0
Democratic	0.758	0	Democratic	0.802	0	Democratic	0.829	0
SNNRCE	0.752	-1	SNNRCE	0.794	-1	SNNRCE	0.822	0
SelfT-1NN	0.745	+1	SelfT-1NN	0.787	+1	SelfT-1NN	0.812	0
TriT-SVM	0.732	0	TriT-SVM	0.782	0	TriT-SVM	0.806	0
TriT-DT	0.679	0	TriT-DT	0.718	-1	SelfT-SVM	0.750	0
SelfT-DT	0.638	-1	SelfT-SVM	0.715	+1	TriT-DT	0.748	0
SelfT-SVM	0.638	+1	SelfT-DT	0.681	0	SelfT-DT	0.710	0

Tabla 2.4: Métodos de auto-etiquetado ordenados mediante los promedios de exactitud obtenidos al evaluar la clasificación transductiva.

La Tabla 2.5 muestra el ranking promedio obtenido para dichos valores de exactitud. El método más exacto es seleccionado como control para la aplicación de la prueba pareada de Hochberg. El SETRED es el método seleccionado en todas las comparaciones que involucran la distancia euclidiana. Para la medida DTW, el método seleccionado es el Tri-training. Por otro lado, los métodos SNNRCE y SelfT muestran los valores más bajos de exactitud. Ambos son significativamente superados por el método de control según muestra la prueba de Hochberg.

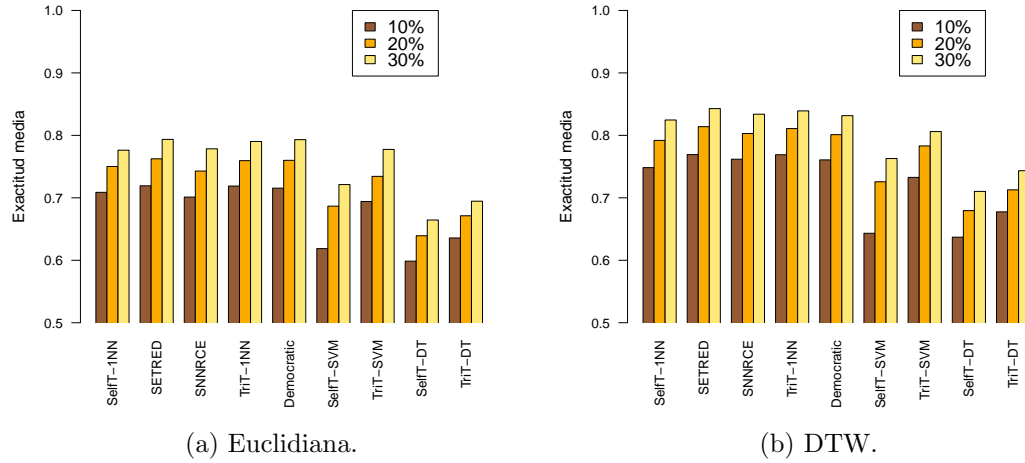


Figura 2.2: Gráfico de barras que muestra la comparación entre la exactitud promedio obtenida por cada método para cada proporción de ejemplos etiquetados.

1NN	Algoritmo	10 %		20 %		30 %	
		Rank.	$p_{Hochberg}$	Rank.	$p_{Hochberg}$	Rank.	$p_{Hochberg}$
Euc	SETRED	<b>56.6</b>	—	<b>52.2</b>	—	<b>49.9</b>	—
	TriT	57.1	0.95	59.4	0.46	60.4	0.28
	SelfT	73.4	0.16	74.4	<b>0.04</b>	85.0	<b><math>5.9 \cdot 10^{-4}</math></b>
	SNNRCE	94.6	<b><math>2.6 \cdot 10^{-4}</math></b>	95.9	<b><math>2.0 \cdot 10^{-5}</math></b>	86.6	<b><math>4.6 \cdot 10^{-4}</math></b>
DTW	TriT	<b>47.7</b>	—	<b>46.7</b>	—	<b>49.1</b>	—
	SETRED	59.0	0.24	51.3	0.63	51.6	0.79
	SNNRCE	87.3	<b><math>8.8 \cdot 10^{-5}</math></b>	92.9	<b><math>6.0 \cdot 10^{-6}</math></b>	85.1	<b><math>4.1 \cdot 10^{-4}</math></b>
	SelfT	87.9	<b><math>8.8 \cdot 10^{-5}</math></b>	91.0	<b><math>1.0 \cdot 10^{-5}</math></b>	96.0	<b><math>4.0 \cdot 10^{-6}</math></b>

Tabla 2.5: Ranking de la exactitud obtenido mediante la prueba alineada de Friedman para ambas medidas de distancia. Adicionalmente se muestran los  $p$ -valores obtenidos al aplicar la prueba pareada de Hochberg.

DT	Algoritmos	10 %			20 %			30 %		
		Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$
Euc	TriT - SelfT	4	<b>31</b>	$4.0 \cdot 10^{-6}$	4	<b>31</b>	$1.0 \cdot 10^{-5}$	3	<b>32</b>	$1.0 \cdot 10^{-6}$
DTW	TriT - SelfT	3	<b>32</b>	$1.0 \cdot 10^{-6}$	5	<b>30</b>	$2.0 \cdot 10^{-6}$	1	<b>34</b>	<b>0.0</b>

SVM	Algoritmos	10 %			20 %			30 %		
		Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$
Euc	TriT - SelfT	5	<b>30</b>	$7.0 \cdot 10^{-6}$	12	<b>23</b>	$4.0 \cdot 10^{-3}$	8	<b>27</b>	$1.0 \cdot 10^{-4}$
DTW	TriT - SelfT	5	<b>30</b>	$3.0 \cdot 10^{-6}$	8	<b>27</b>	$3.2 \cdot 10^{-5}$	7	<b>28</b>	$2.1 \cdot 10^{-5}$

Tabla 2.6: Prueba de signos de Wilcoxon aplicados a la exactitud de los métodos que tienen DT o SVM como clasificadores base. De cada comparación se muestra la cantidad de rangos positivos y negativos junto al  $p$ -valor obtenido.

La Tabla 2.6 muestra el resultado de la prueba de signos de Wilcoxon aplicada a los valores de exactitud de los métodos que utilizan DT y SVM como clasificadores base. El método TriT supera el SelfT en ambos clasificadores y para todas las proporciones de etiquetados, incluyendo ambas medidas de distancia. Las diferencias obtenidas resultan significativas para un nivel de significación de  $\alpha = 0.05$ .

Para finalizar, en la Tabla 2.7 se ofrece una comparación entre los métodos más competitivos de cada esquema de monoaprendizaje y el enfoque de multiaprendizaje (Democratic). Se considera como “competitivo” aquel método que no ha sido superado más de una vez de manera significativa por otro método en las seis comparaciones (dos medidas de distancia  $\times$  tres proporciones de ejemplos etiquetados) mostradas en las Tablas 2.5 y 2.6. Siguiendo este criterio los métodos que se destacan por su exactitud son: SETRED y TriT. La prueba alineada de Friedman, aplicada a la comparación entre los métodos destacados en unión con el Democratic, detecta diferencias significativas para un nivel de significación de  $\alpha = 0.05$  en todas las comparaciones realizadas. Para el 10 % de ejemplos etiquetados, el método de control seleccionado es el TriT-1NN tanto para la distancia euclidiana como para DTW. Para el resto de las proporciones, los métodos de control seleccionados son el Democratic y el TriT-1NN para las medidas euclidiana y DTW, respectivamente. SETRED exhibe un comportamiento competitivo a lo



largo de las comparaciones. Por el contrario, TriT-SVM y TriT-DT son significativamente superados por el método de control en la mayoría de las comparaciones efectuadas.

Distancia	Algoritmo	10 %		20 %		30 %	
		Rank	$p_{Hochberg}$	Rank	$p_{Hochberg}$	Rank	$p_{Hochberg}$
Euc	TriT-1NN	<b>66.0</b>	–	70.0	0.48	72.4	0.34
	Democratic	66.9	0.93	<b>61.5</b>	–	<b>61.1</b>	–
	SETRED	73.1	0.93	71.2	0.48	73.9	0.34
	TriT-SVM	92.3	<b>0.08</b>	91.9	<b>0.03</b>	85.0	0.14
	TriT-DT	141.4	<b>0.0</b>	145.1	<b>0.0</b>	147.4	<b>0.0</b>
DTW	TriT-1NN	<b>56.4</b>	–	<b>57.7</b>	–	<b>57.5</b>	–
	SETRED	66.7	0.39	63.6	0.62	64.9	0.54
	Democratic	71.7	0.39	72.2	0.45	71.4	0.49
	TriT-SVM	103.8	$2.7 \cdot 10^{-4}$	100.0	$1.4 \cdot 10^{-3}$	100.8	$1.0 \cdot 10^{-3}$
	TriT-DT	141.2	<b>0.0</b>	146.3	<b>0.0</b>	145.2	<b>0.0</b>

Tabla 2.7: Ranking obtenido mediante la prueba alineada de Friedman aplicada a la comparación de la exactitud de los métodos más competitivos. Adicionalmente se muestran los  $p$ -valores obtenidos al aplicar la prueba pareada de Hochberg.

### 2.3.2 Resultados inductivos

El objetivo principal del aprendizaje inductivo consiste en clasificar instancias que no se encuentran incluidas en la fase de entrenamiento. Este análisis resulta sumamente útil para probar la hipótesis aprendida en la fase de entrenamiento y sus habilidades de generalización. La Tabla 2.8 muestra los promedios de exactitud obtenidos tanto para la distancia euclidiana como para DTW. Los resultados completos se muestran en la sección B.1. Por su parte, la Figura 2.3 muestra un diagrama de caja con estos resultados agrupados por las diferentes proporciones de ejemplos etiquetados. La Figura 2.4 refleja en un gráfico de barras la exactitud promedio de cada método y las mejoras que se obtienen con el aumento en el conjunto de entrenamiento de los ejemplos etiquetados.

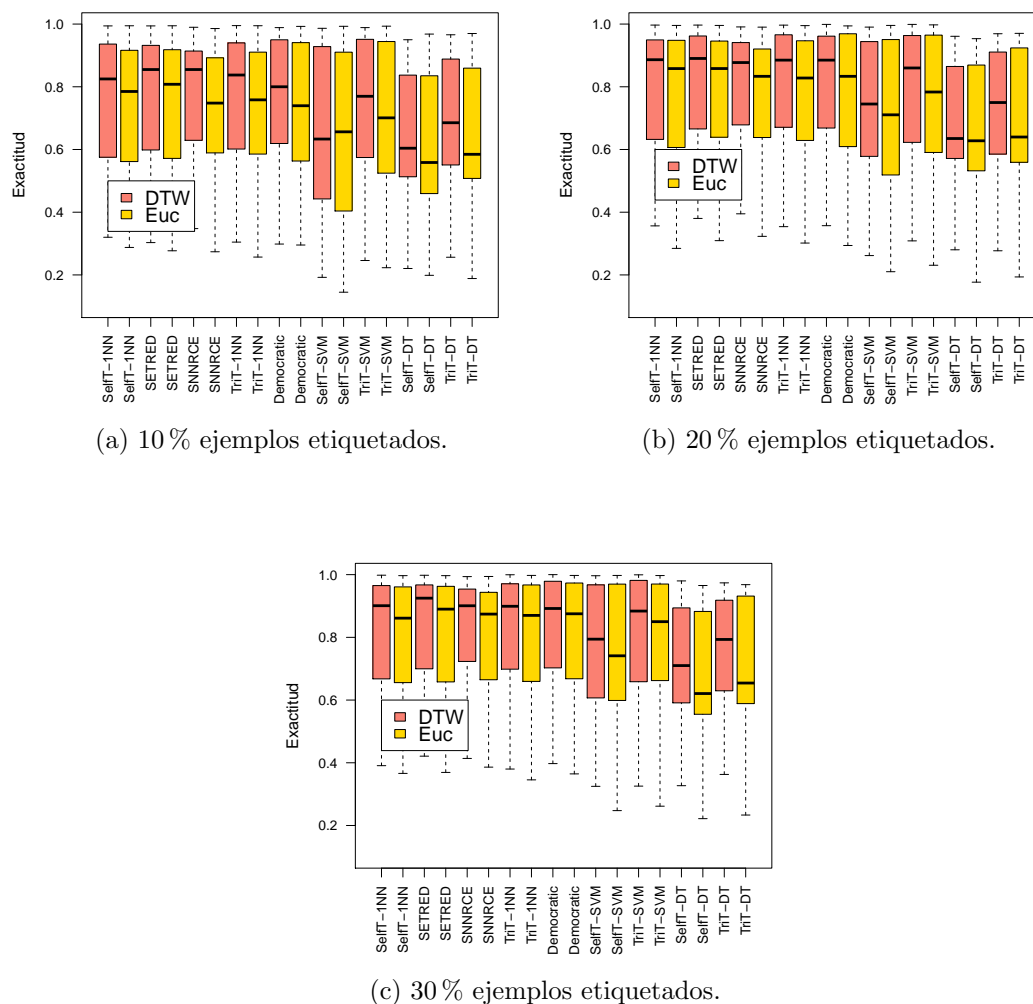


Figura 2.3: Diagrama de cajas que muestra los resultados de exactitud para diferentes proporciones de ejemplos etiquetados y las medidas euclidiana y DTW.

Se realiza nuevamente una comparación entre la exactitud de los métodos monoaprendizaje agrupados por su esquema de aprendizaje. La prueba alineada de Friedman, aplicada a la exactitud de todos aquellos métodos que utilizan el 1NN como clasificador base, detecta diferencias significativas en todas las comparaciones realizadas. La Tabla 2.9 muestra los ranking obtenidos. El método de control seleccionado es el SETRED para todas las proporciones de etiquetados y ambas medidas, con la excepción del 10 % con la medida DTW donde es seleccionado el TriT. Los métodos SNNRCE y SelfT muestran los valores más bajos

de exactitud y ambos son superados significativamente por el método de control en la mayoría de las comparaciones según los resultados de la prueba pareada de Hochberg.

10 %			20 %			30 %		
Euclidiana	Exact.	$\Delta K$	Euclidiana	Exact.	$\Delta K$	Euclidiana	Exact.	$\Delta K$
SETRED	0.719	-1	SETRED	0.764	0	Democratic	0.794	0
Democratic	0.719	+1	Democratic	0.757	0	SETRED	0.793	0
TriT-1NN	0.715	0	TriT-1NN	0.756	0	TriT-1NN	0.788	0
SelfT-1NN	0.711	0	SelfT-1NN	0.753	0	SNNRCE	0.786	-1
SNNRCE	0.702	0	SNNRCE	0.749	0	SelfT-1NN	0.784	+1
TriT-SVM	0.693	0	TriT-SVM	0.734	0	TriT-SVM	0.778	0
TriT-DT	0.633	-1	SelfT-SVM	0.694	0	SelfT-SVM	0.739	0
SelfT-SVM	0.627	+1	TriT-DT	0.667	0	TriT-DT	0.693	0
SelfT-DT	0.602	0	SelfT-DT	0.641	0	SelfT-DT	0.665	0
DTW	Exact.	$\Delta K$	DTW	Exact.	$\Delta K$	DTW	Exact.	$\Delta K$
SETRED	0.769	0	SETRED	0.813	0	SETRED	0.842	0
TriT-1NN	0.768	0	TriT-1NN	0.810	0	TriT-1NN	0.839	0
SNNRCE	0.761	0	SNNRCE	0.802	0	SNNRCE	0.833	-1
Democratic	0.760	0	Democratic	0.801	0	Democratic	0.831	+1
SelfT-1NN	0.748	0	SelfT-1NN	0.791	0	SelfT-1NN	0.824	0
TriT-SVM	0.732	0	TriT-SVM	0.783	0	TriT-SVM	0.806	0
TriT-DT	0.677	0	SelfT-SVM	0.725	0	SelfT-SVM	0.762	0
SelfT-SVM	0.643	0	TriT-DT	0.712	0	TriT-DT	0.743	0
SelfT-DT	0.636	0	SelfT-DT	0.679	0	SelfT-DT	0.710	0

Tabla 2.8: Métodos de auto-etiquetado ordenados mediante los promedios de exactitud obtenidos al evaluar la clasificación inductiva.

La Tabla 2.10 muestra la aplicación de la prueba de signos de Wilcoxon a la exactitud de aquellos métodos que utilizan DT y SVM como clasificadores base. En todas las comparaciones, el método TriT supera de manera significativa al SelfT para ambos clasificadores base.

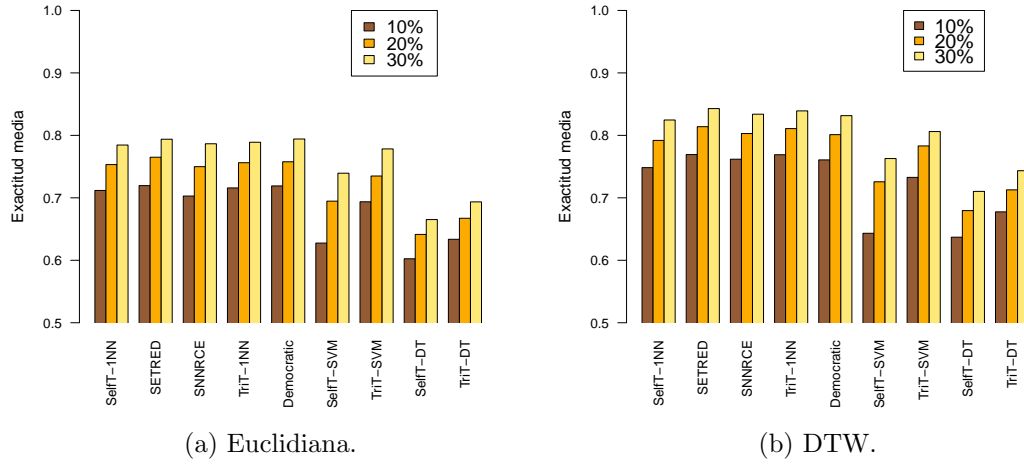


Figura 2.4: Gráfico de barras que muestra la comparación entre la exactitud promedio obtenida por cada método para cada proporción de ejemplos etiquetados.

1NN	Algoritmo	10 %		20 %		30 %	
		Rank.	$p_{Hochberg}$	Rank.	$p_{Hochberg}$	Rank.	$p_{Hochberg}$
Euc	SETRED	<b>54.7</b>	—	<b>50.7</b>	—	<b>54.6</b>	—
	TriT	63.7	0.35	72.2	<b>0.02</b>	66.5	0.21
	SelfT	68.1	0.33	74.6	<b>0.02</b>	82.5	<b>0.01</b>
	SNNRCE	95.4	<b><math>8.2 \cdot 10^{-5}</math></b>	84.4	<b><math>1.4 \cdot 10^{-3}</math></b>	78.3	<b>0.02</b>
DTW	TriT	<b>54.6</b>	—	60.6	0.52	62.2	0.28
	SETRED	59.3	0.62	<b>54.5</b>	—	<b>51.9</b>	—
	SNNRCE	76.2	0.05	79.7	<b>0.01</b>	77.9	<b>0.01</b>
	SelfT	91.8	<b><math>3.6 \cdot 10^{-4}</math></b>	87.0	<b><math>2.4 \cdot 10^{-3}</math></b>	89.8	<b><math>2.7 \cdot 10^{-4}</math></b>

Tabla 2.9: Ranking de la exactitud obtenido mediante la prueba alineada de Friedman para ambas medidas de distancia. Adicionalmente se muestran los  $p$ -valores obtenidos al aplicar la prueba pareada de Hochberg.

DT	Algoritmos	10 %			20 %			30 %		
		Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$
Euc	TriT - SelfT	6	<b>28</b>	$1.3 \cdot 10^{-5}$	6	<b>29</b>	$7.1 \cdot 10^{-5}$	3	<b>32</b>	$5.0 \cdot 10^{-6}$
DTW	TriT - SelfT	3	<b>32</b>	$2.0 \cdot 10^{-6}$	6	<b>29</b>	$1.0 \cdot 10^{-5}$	3	<b>31</b>	$5.3 \cdot 10^{-5}$

SVM	Algoritmos	10 %			20 %			30 %		
		Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$
Euc	TriT - SelfT	6	<b>29</b>	$2.6 \cdot 10^{-5}$	12	<b>23</b>	$8.0 \cdot 10^{-3}$	9	<b>25</b>	$1.8 \cdot 10^{-3}$
DTW	TriT - SelfT	3	<b>32</b>	$4.0 \cdot 10^{-6}$	7	<b>27</b>	$6.1 \cdot 10^{-5}$	9	<b>26</b>	$2.2 \cdot 10^{-4}$

Tabla 2.10: Prueba de signos de Wilcoxon aplicados a la exactitud de los métodos que tienen DT o SVM como clasificadores base. De cada comparación se muestra la cantidad de rangos positivos y negativos junto al  $p$ -valor obtenido.

Distancia	Algoritmo	10 %		20 %		30 %	
		Rank.	$p_{Hochberg}$	Rank.	$p_{Hochberg}$	Rank.	$p_{Hochberg}$
Euc	Democratic	<b>61.6</b>	—	<b>64.3</b>	—	<b>62.5</b>	—
	SETRED	71.5	0.41	65.2	0.93	72.8	0.39
	TriT-1NN	71.8	0.41	72.8	0.93	75.1	0.39
	TriT-SVM	93.0	<b>0.02</b>	92.0	0.06	83.3	0.26
	TriT-DT	141.9	<b>0.0</b>	145.5	<b>0.0</b>	146.0	<b>0.0</b>
DTW	TriT-1NN	<b>60.5</b>	—	62.5	0.96	62.8	0.76
	SETRED	67.0	0.59	<b>61.9</b>	—	<b>59.2</b>	—
	Democratic	70.9	0.59	70.2	0.96	70.8	0.68
	TriT-SVM	101.9	$1.8 \cdot 10^{-3}$	97.4	<b>0.01</b>	99.3	$2.7 \cdot 10^{-3}$
	TriT-DT	139.6	<b>0.0</b>	147.9	<b>0.0</b>	147.8	<b>0.0</b>

Tabla 2.11: Ranking obtenido mediante la prueba alineada de Friedman aplicada a la comparación de la exactitud de los métodos más competitivos. Adicionalmente se muestran los  $p$ -valores obtenidos al aplicar la prueba pareada de Hochberg.

Finalmente, en la Tabla 2.11 se ofrece una comparación entre los métodos más competitivos pertenecientes al enfoque de monoaprendizaje, incluyendo además el enfoque de multiaprendizaje. Una vez más, los métodos de monoaprendizaje que sobresalen son: SETRED y TriT. La prueba alineada de Friedman,

aplicada a los valores de exactitud, detecta diferencias significativas en todas las comparaciones realizadas. El Democratic es seleccionado como método de control en todas las comparaciones que utilizan la distancia euclidiana. En el caso de la medida DTW, para el 10 % de etiquetados el método de control seleccionado es TriT-1NN y para el resto de las comparaciones el método de control seleccionado es el SETRED. Los métodos TriT-SVM y TriT-DT son superados de manera significativa por el método de control en la mayoría de las comparaciones, excepto para el 20 % y 30 % de etiquetados donde el método TriT-SVM exhibe un mejor comportamiento cuando se utiliza la distancia euclidiana.

### 2.3.3 Discusión sobre los resultados de clasificación

A continuación se brinda una discusión general que abarca los aspectos sobresalientes que se observaron de manera común tanto para el aprendizaje transductivo como para el inductivo. Adicionalmente, se destacan aquellos métodos que de manera general sobresalieron de acuerdo a los resultados obtenidos.

- La exactitud de la mayoría de los métodos experimenta un incremento con el aumento de los ejemplos etiquetados que se suministran durante el entrenamiento. Este incremento resulta moderado para la mayoría de los casos con la excepción de los métodos que utilizan las SVMs como clasificador base, donde el incremento de la exactitud resulta más notable.
- El método SelfT estándar es superado por otros métodos de auto-etiquetado independientemente del esquema de aprendizaje utilizado.
- Usualmente no existen diferencias en el orden obtenido entre los métodos usando los estadísticos exactitud y kappa. Esto significa que no existen diferencias apreciables en la forma en que los clasificadores se benefician de los aciertos ocurridos de manera aleatoria.
- De forma general, el 1NN ofrece los mejores resultados como clasificador base para los métodos de auto-etiquetado tanto en el aprendizaje inductivo como el transductivo.
- Aunque SVM y DT no ofrecen resultados competitivos como clasificadores

base, cuando se combinan con el 1NN, siguiendo un esquema multiaprendizaje (Democratic), se obtienen resultados prometedores.

- El uso de DTW implica un aumento de la exactitud en comparación con la distancia euclidiana, pero esta ventaja se reduce en presencia del SVM como clasificador base. Este comportamiento es causado por el incumplimiento de la propiedad PDS de los núcleos que se construyen utilizando DTW.
- Los métodos SETRED, Democratic y TriT-1NN son los que mostraron mejor comportamiento a lo largo de este estudio. Específicamente para la distancia euclidiana, se recomienda utilizar el método Democratic, pero si en el conjunto de entrenamiento existen muy pocos ejemplos etiquetados entonces las capacidades transductivas del método TriT-1NN resultan igual de competitivas. En el caso de DTW, se recomienda utilizar TriT-1NN y SETRED para los escenarios transductivo e inductivo, respectivamente. No obstante, el método TriT-1NN también exhibe un buen comportamiento en el aprendizaje inductivo cuando existe un número extremadamente reducido de ejemplos etiquetados.

### 2.3.4 Análisis de los tiempos de ejecución

El tiempo de ejecución consumido por los métodos de auto-etiquetado depende tanto de la complejidad temporal asociada con el propio método como de la complejidad temporal del algoritmo que se utiliza como clasificador base. A continuación se presenta un estudio empírico, basado en una muestra de 12 conjuntos de datos que formaron parte de la experimentación. Todos los experimentos se realizaron en un clúster que contiene 46 nodos, cada uno de ellos equipado con un procesador Intel® Core™ i7-930 a 2.8 GHz y 24 GB de memoria RAM.

Los métodos semi-supervisados, durante el proceso de entrenamiento y evaluación, recibieron como entrada la matriz de distancia entre todos los ejemplos existentes. Por este motivo el consumo de tiempo asociado al cálculo de las distancias no se contempla en el análisis que se presenta a continuación.

La Tabla 2.12 muestra los tiempos de cómputo resultantes de la ejecución secuencial del esquema de validación cruzada de cinco particiones. Los menores tiempos de ejecución se obtuvieron con el TriT-1NN en todos los casos. De

manera general, el clasificador base 1NN aportó los mejores tiempos mientras que el SVM obtuvo los más elevados. Esto se debe al mecanismo de validación cruzada que utiliza internamente el SVM para realizar el ajuste de parámetros. El clasificador base DT mantiene resultados intermedios entre 1NN y SVM. El método Democratic presenta tiempos elevados de ejecución debido a que entrena un clasificador base de cada tipo. El comportamiento de los métodos SETRED y SNNRCE resulta competitivo también ya que el clasificador base que utilizan es el 1NN.

Datos	SETRED	SNNRCE	TrT-1NN	TrT-SVM	TrT-DT	Self-1NN	Self-SVM	Self-DT	Democratic
Coffee	$9.2 \cdot 10^{-1}$	$9.4 \cdot 10^{-1}$	<b><math>4.9 \cdot 10^{-1}</math></b>	$2.5 \cdot 10^2$	4.9	$5.6 \cdot 10^{-1}$	$3.1 \cdot 10^2$	4.8	$7.6 \cdot 10^1$
Fish	$4.5 \cdot 10^1$	$2.1 \cdot 10^1$	<b>1.7</b>	$1.7 \cdot 10^3$	$1.6 \cdot 10^2$	$2.1 \cdot 10^1$	$1.4 \cdot 10^4$	$9.6 \cdot 10^2$	$1.0 \cdot 10^4$
Gun_Point	$2.0 \cdot 10^1$	$1.4 \cdot 10^1$	<b><math>9.6 \cdot 10^{-1}</math></b>	$8.1 \cdot 10^2$	$3.9 \cdot 10^1$	$1.5 \cdot 10^1$	$1.0 \cdot 10^4$	$2.8 \cdot 10^2$	$1.0 \cdot 10^3$
InlineS	$2.0 \cdot 10^2$	$5.3 \cdot 10^1$	<b>4.0</b>	$7.6 \cdot 10^3$	$1.9 \cdot 10^2$	$1.0 \cdot 10^2$	$1.0 \cdot 10^5$	$5.2 \cdot 10^3$	$7.3 \cdot 10^4$
Italy	$3.5 \cdot 10^2$	$7.0 \cdot 10^2$	<b>6.2</b>	$1.3 \cdot 10^3$	$1.3 \cdot 10^3$	$3.3 \cdot 10^2$	$7.0 \cdot 10^3$	$1.0 \cdot 10^3$	$1.8 \cdot 10^3$
Lighting2	5.1	4.5	<b><math>7.2 \cdot 10^{-1}</math></b>	$1.1 \cdot 10^2$	$1.1 \cdot 10^1$	3.4	$1.0 \cdot 10^3$	$4.6 \cdot 10^1$	$1.9 \cdot 10^2$
Lighting7	4.9	3.7	<b><math>8.5 \cdot 10^{-1}</math></b>	$1.7 \cdot 10^3$	$2.3 \cdot 10^1$	2.6	$1.2 \cdot 10^3$	$9.6 \cdot 10^1$	$2.3 \cdot 10^3$
Olive	1.3	$9.6 \cdot 10^{-1}$	<b><math>6.4 \cdot 10^{-1}</math></b>	$5.0 \cdot 10^2$	$1.2 \cdot 10^1$	$6.5 \cdot 10^{-1}$	$1.7 \cdot 10^3$	$2.3 \cdot 10^1$	$9.1 \cdot 10^1$
OSULeaf	$7.2 \cdot 10^1$	$2.0 \cdot 10^1$	<b>2.3</b>	$1.7 \cdot 10^3$	$1.8 \cdot 10^2$	$3.7 \cdot 10^1$	$1.9 \cdot 10^4$	$1.4 \cdot 10^3$	$1.8 \cdot 10^4$
SonyII	$3.3 \cdot 10^2$	$6.7 \cdot 10^2$	<b>5.7</b>	$1.3 \cdot 10^3$	$8.1 \cdot 10^2$	$3.0 \cdot 10^2$	$4.7 \cdot 10^3$	$1.8 \cdot 10^3$	$4.2 \cdot 10^3$
Trace	$1.1 \cdot 10^1$	8.9	<b>1.1</b>	$4.2 \cdot 10^3$	$8.9 \cdot 10^1$	5.9	$1.3 \cdot 10^4$	$1.7 \cdot 10^2$	$9.3 \cdot 10^3$
Yoga	$2.3 \cdot 10^3$	$2.6 \cdot 10^4$	<b><math>3.0 \cdot 10^1</math></b>	$2.9 \cdot 10^4$	$6.9 \cdot 10^3$	$2.3 \cdot 10^3$	$1.6 \cdot 10^5$	$1.1 \cdot 10^4$	$4.2 \cdot 10^5$

Tabla 2.12: Consumo de tiempo (segundos) de cómputo durante el entrenamiento y evaluación de los métodos de auto-etiquetado para el 10 % de ejemplos etiquetados con la distancia euclidiana.

### 2.3.5 Efectividad de los métodos de auto-etiquetado

Un procedimiento recomendable [29] en presencia de ejemplos etiquetados y no etiquetados consiste en comenzar por entrenar un clasificador supervisado utilizando sólo los ejemplos etiquetados, el cual será considerado como cota inferior de referencia (CIR). Las comparaciones con dicho clasificador permiten identificar situaciones donde la adición de ejemplos no etiquetados causan una degradación en el desempeño del clasificador obtenido.

A continuación presentamos un análisis en este sentido donde se evaluará la ganancia de exactitud que se obtiene a partir de la adición de ejemplos no

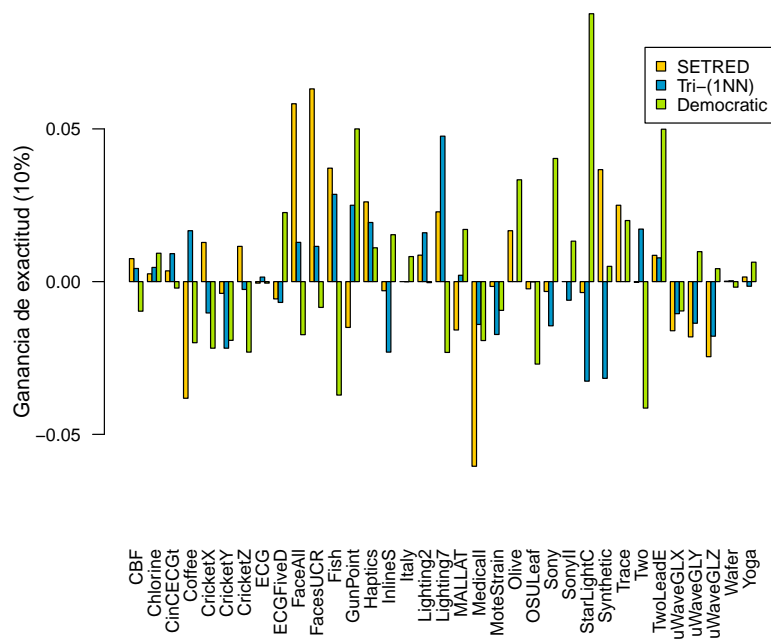


etiquetados durante el entrenamiento. La ganancia de exactitud se calcula, según la Expresión 2.4, restando la exactitud obtenida con el clasificador supervisado de la obtenida utilizando SSC. En ambos casos el desempeño de los clasificadores se evalúa en el conjunto de prueba, utilizando validación cruzada con cinco particiones. El método seleccionado para entrenar el clasificador supervisado que será utilizado como cota inferior es el 1NN, ya que este ofrece los mejores resultados de clasificación. Todos los valores de exactitud obtenidos mediante este clasificador se muestran en la sección B.2.

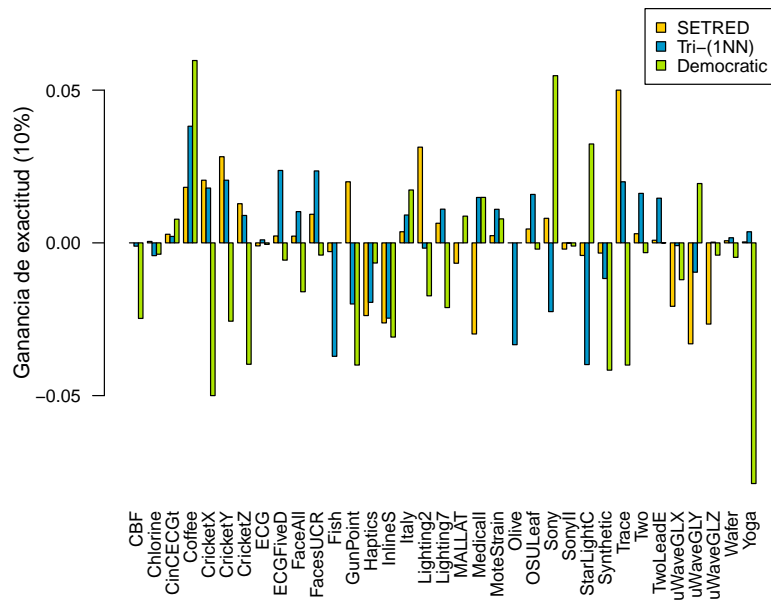
$$Ganancia = Exactitud_{(SSC)} - Exactitud_{(CIR)} \quad (2.4)$$

De manera general, los métodos de auto-etiquetado con mayores ganancias de exactitud se corresponden con los que obtuvieron un desempeño destacado en las comparaciones realizadas anteriormente. Por este motivo, el análisis estará enfocado en los siguientes métodos: SETRED, TriT-1NN y Democratic. Las Figuras 2.5, 2.6 y 2.7 muestran de manera detallada la ganancia de exactitud que experimenta cada conjunto de datos mediante los tres métodos de auto-etiquetado y para cada proporción de ejemplos etiquetados. Una ganancia negativa implica que ocurrió una degradación en el desempeño del clasificador al aplicar SSC. Es posible observar un comportamiento extremadamente diverso de las ganancias para las distintas proporciones de ejemplos etiquetados y los métodos aplicados.

Existen conjuntos de datos que no se benefician de la SSC, por ejemplo ECG [156] y Wafer. El tamaño de estos conjuntos de datos provocan que la hipótesis aprendida por el clasificador supervisado sea perfectamente capaz de obtener buenos resultados de exactitud durante la clasificación inductiva. En otros conjuntos de datos, como MedicalI o InlineS, el desempeño de la clasificación se deteriora con la adición de ejemplos no etiquetados para la mayoría de las ejecuciones. Estos constituyen casos típicos donde el conjunto inicial de ejemplos etiquetados resultan insuficientes para entrenar un modelo correcto de forma tal que los ejemplos no etiquetados resulten realmente beneficiosos.

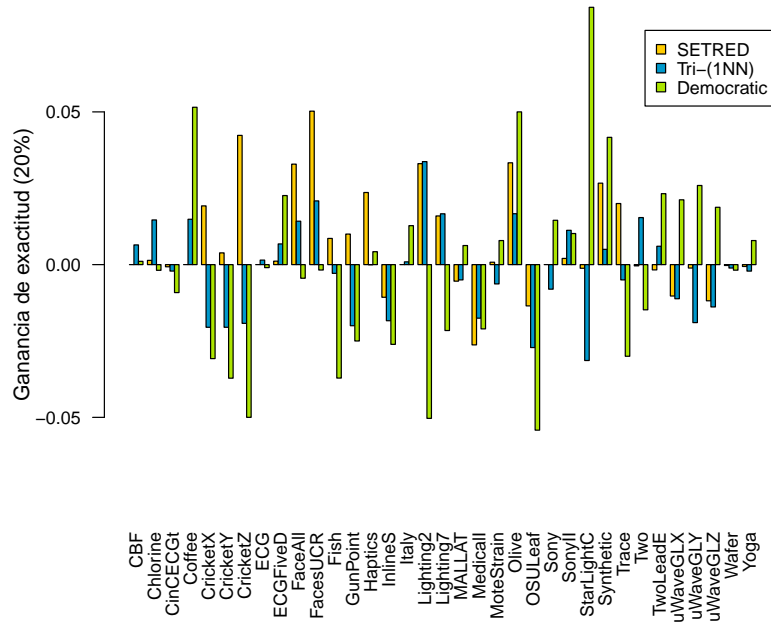


(a) Euclidiana.

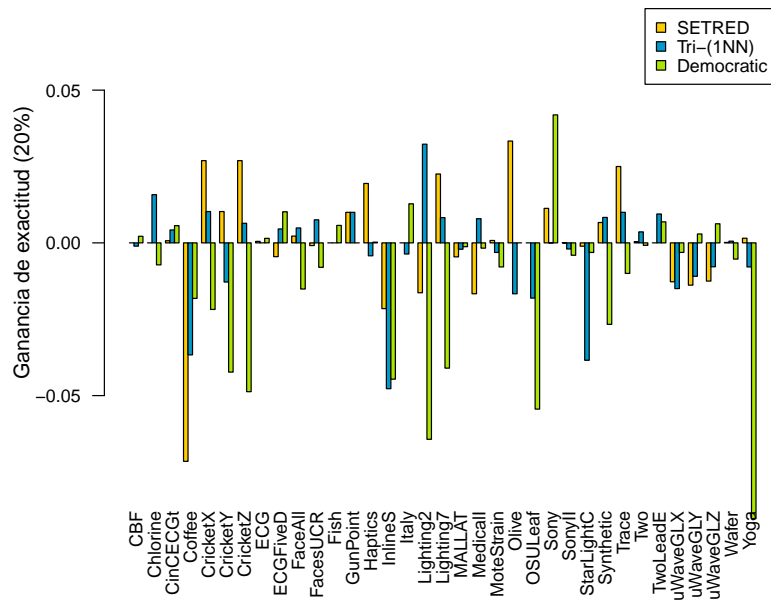


(b) DTW.

Figura 2.5: Ganancia de exactitud obtenida mediante los tres mejores métodos para el 10% de ejemplos etiquetados.

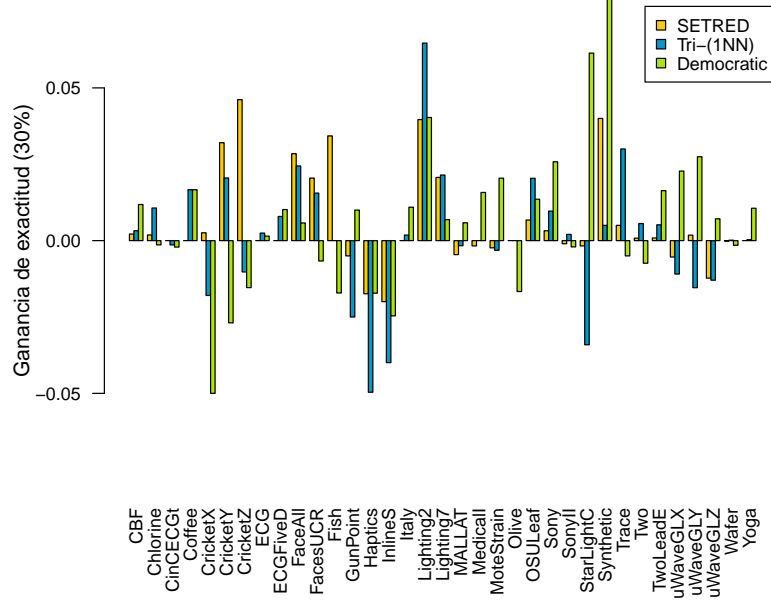


(a) Euclidiana.

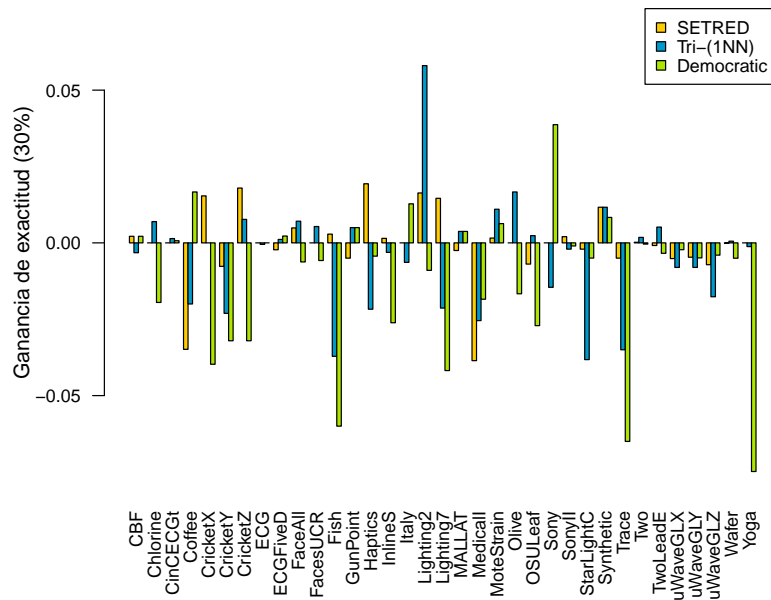


(b) DTW.

Figura 2.6: Ganancia de exactitud obtenida mediante los tres mejores métodos para el 20% de ejemplos etiquetados.



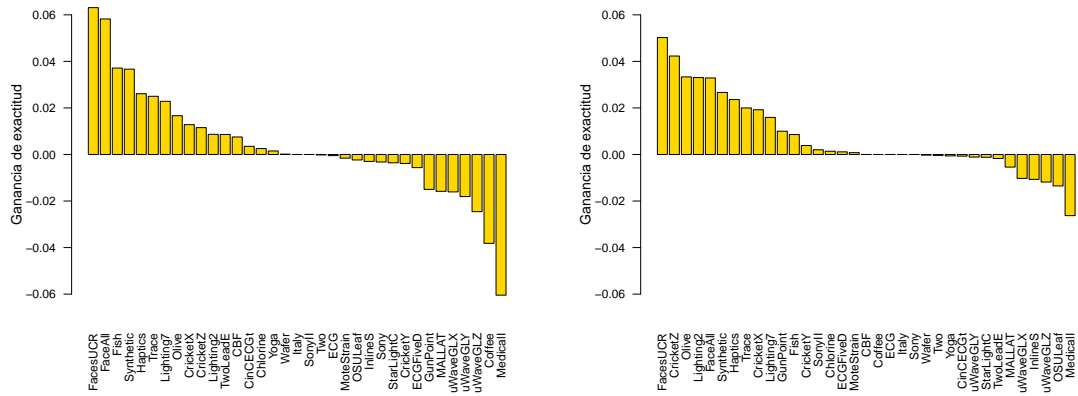
(a) Euclidiana.



(b) DTW.

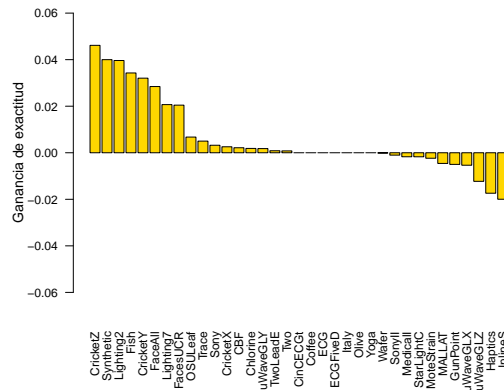
Figura 2.7: Ganancia de exactitud obtenida mediante los tres mejores métodos para el 30% de ejemplos etiquetados.

Es necesario hacer notar que ambos conjuntos de datos son problemas con múltiples clases lo cual dificulta el entrenamiento (MedicalI tiene diez clases e InlineS siete). En el Capítulo 3 se retoma el análisis sobre la complejidad que presentan los conjuntos de datos mencionados.



(a) Euclidiana (10 %).

(b) Euclidiana (20 %).



(c) Euclidiana (30 %).

Figura 2.8: Ganancia de exactitud obtenida con el método SETRED para la distancia euclidiana.

Aunque existen situaciones desfavorables para la SSC sí es posible obtener beneficios de este paradigma. Las Figuras 2.8 y 2.9 muestran las ganancias obtenidas con el método SETRED, ordenadas de forma decreciente. De manera general, para el 20% de ejemplos etiquetados puede observarse una ganancia

positiva apreciable con la distancia euclidiana. Para DTW las ganancias resultan más apreciables para el 10%. También se puede apreciar que las ganancias positivas así como su magnitud dependen estrechamente de la proporción de ejemplos etiquetados. En resumen, las circunstancias que hacen del SSL un enfoque adecuado, para un conjunto de datos temporales en particular, dependen de los supuestos adoptados por el clasificador además de las características particulares del problema.

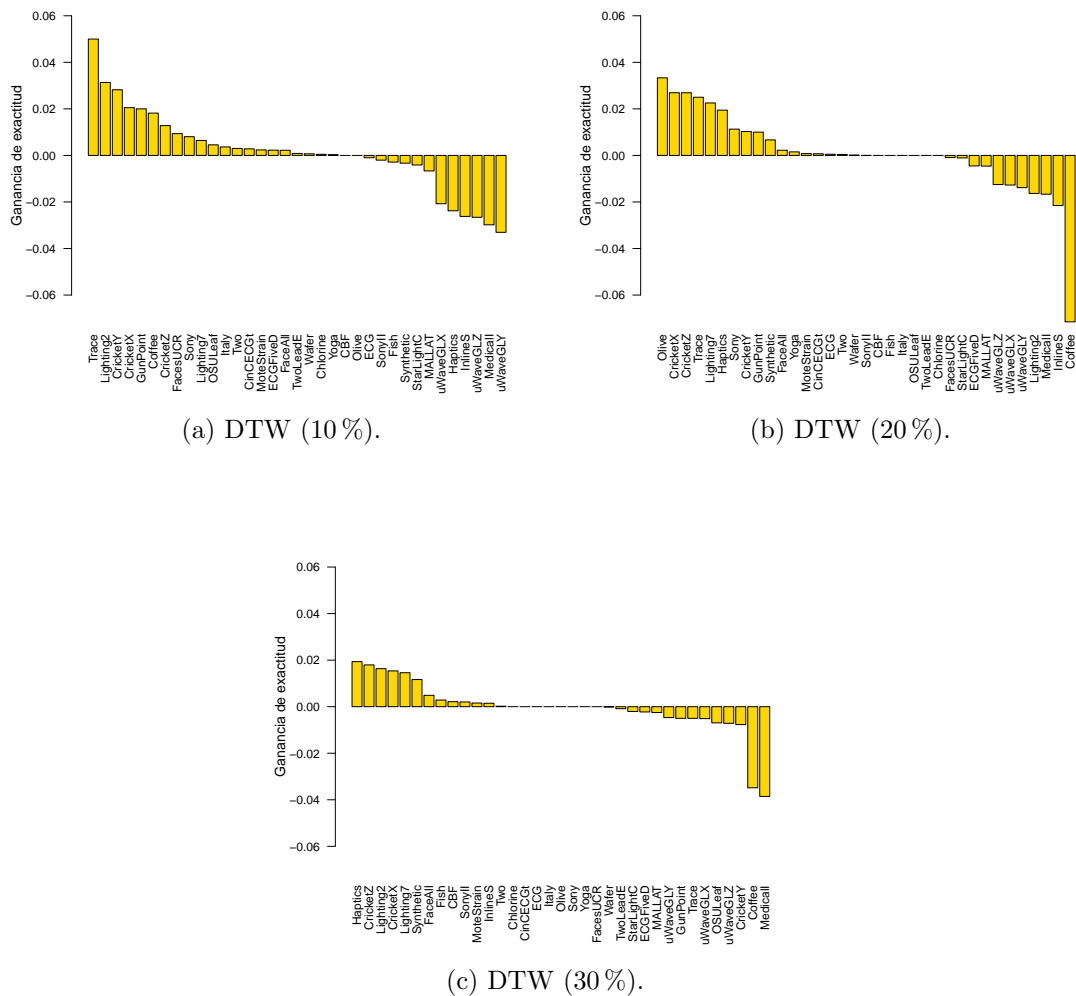
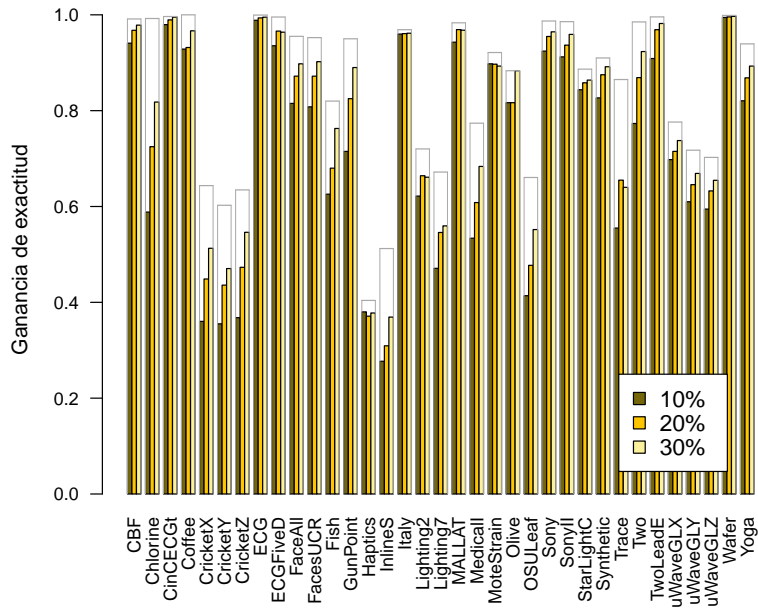
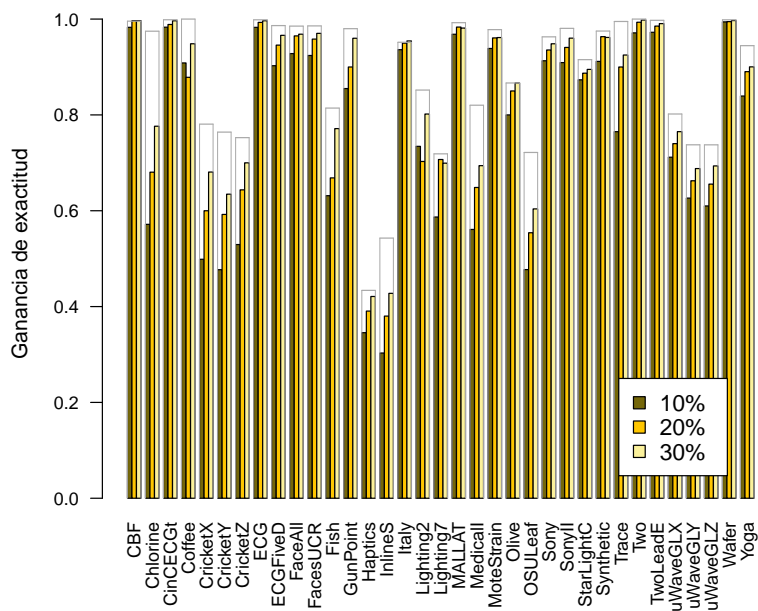


Figura 2.9: Ganancia de exactitud obtenida con el método SETRED para la medida DTW.

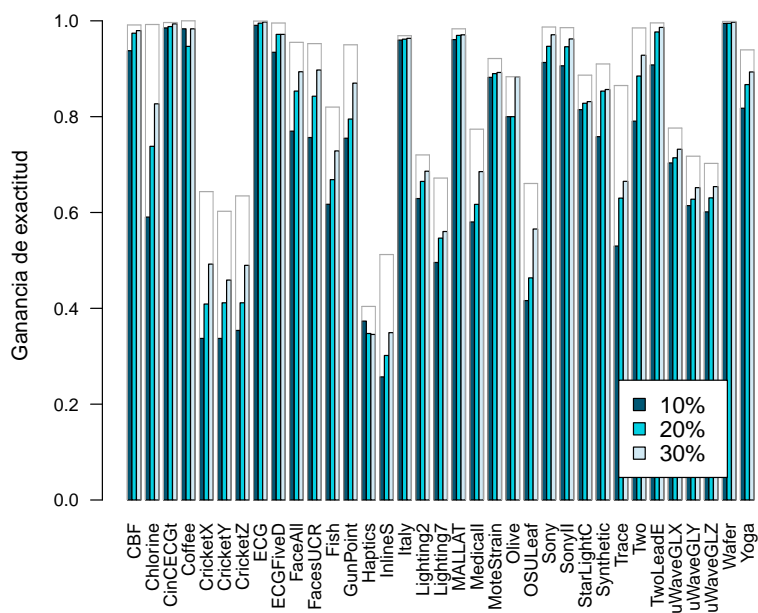


(a) Euclidiana.

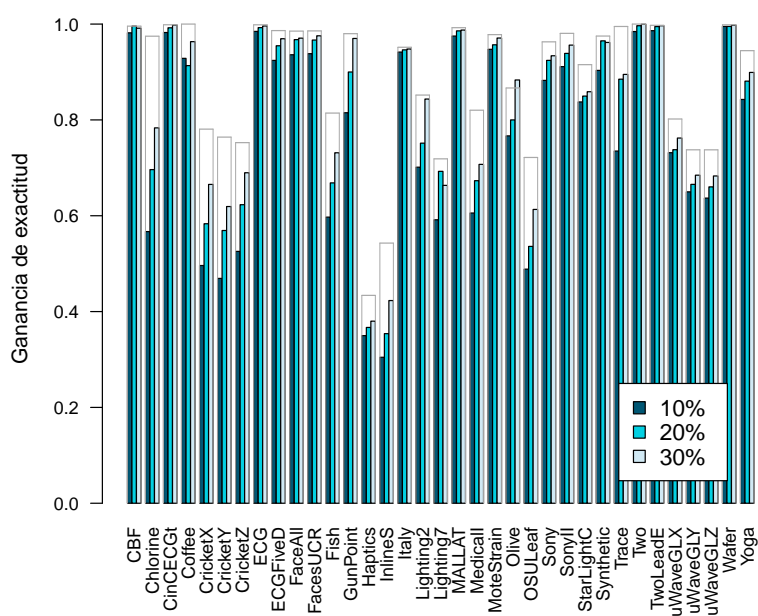


(b) DTW.

Figura 2.10: Comparación del método SETRED con la cota superior de exactitud.



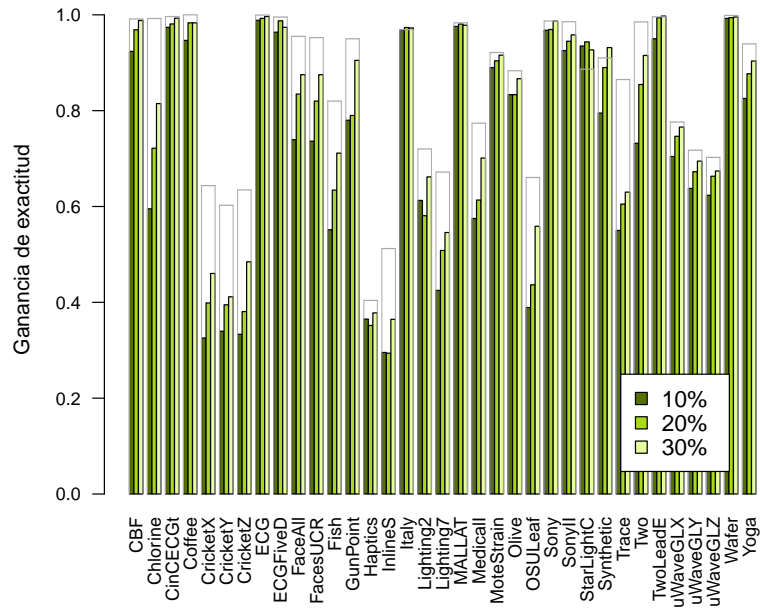
(a) Euclidiana.



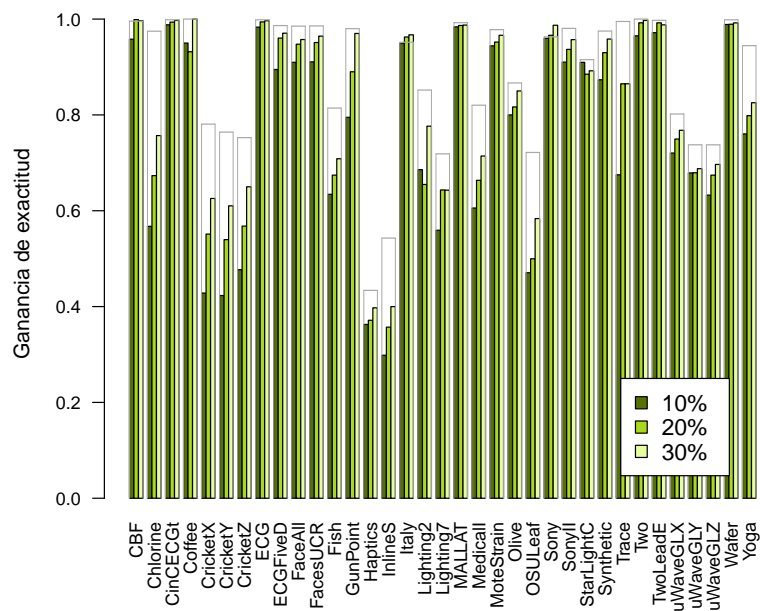
(b) DTW.

Figura 2.11: Comparación del método TriT-1NN con la cota superior de exactitud.





(a) Euclidiana.



(b) DTW.

Figura 2.12: Comparación del método Democratic con la cota superior de exactitud.

En adición al análisis realizado, se consideró además el entrenamiento del clasificador supervisado 1NN a partir de todos los ejemplos disponibles durante el entrenamiento (incluidos los no etiquetados con sus respectivas etiquetas). Este clasificador se evaluó en el conjunto de prueba y este resultado se consideró como una cota superior de referencia (CSR) para los métodos de auto-etiquetado. La diferencia obtenida entre ambos enfoques (Expresión 2.5) permite evaluar cuánto distan los resultados semi-supervisados del paradigma supervisado.

$$\text{Diferencia} = \text{Exactitud}_{(CSR)} - \text{Exactitud}_{(SSC)} \quad (2.5)$$

Las Figuras 2.10, 2.11 y 2.12 muestran los resultados de exactitud obtenidos mediante SSC acotados superiormente por el clasificador supervisado. La Tabla 2.13 muestran un resumen del comportamiento general de cada método. El SETRED muestra las menores diferencias<sup>2</sup> aunque estos valores se encuentran muy cercanos a los obtenidos para el resto de los métodos. Los resultados obtenidos muestran la competitividad del paradigma semi-supervisado para la mayoría de los conjuntos de datos en comparación con la cota superior considerada. Resulta notable cómo para algunos conjuntos de datos, como StarLightC para la distancia euclidiana y Sony para DTW, la hipótesis de multiaprendizaje aprendida por el Democratic supera la cota superior obtenida.

Algoritmo	Euclidiana						DTW					
	10 %		20 %		30 %		10 %		20 %		30 %	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
SETRED	<b>0.13</b>	<b>0.10</b>	<b>0.08</b>	<b>0.07</b>	<b>0.05</b>	<b>0.05</b>	<b>0.11</b>	<b>0.09</b>	<b>0.07</b>	0.07	<b>0.04</b>	<b>0.04</b>
TriT-1NN	0.13	0.10	0.09	0.07	0.06	0.05	0.11	0.10	0.07	<b>0.07</b>	0.04	0.04
Democratic	0.13	0.11	0.09	0.09	0.05	0.06	0.12	0.11	0.08	0.08	0.05	0.05

Tabla 2.13: Estadísticos media ( $\mu$ ) y desviación estándar ( $\sigma$ ) de las diferencias obtenidas para cada método respecto a la CSR.

<sup>2</sup>Los menores valores por columna se han resaltado en negrita teniendo en cuenta una mayor cantidad de cifras significativas que las mostradas.

## 2.4 Sumario

Este capítulo estuvo dedicado a evaluar la aplicabilidad de diferentes métodos de auto-etiquetado existentes al problema de la SSC en el dominio temporal. Además del popular SelfT con el 1NN como clasificador base, se han explorado otras combinaciones de métodos de auto-etiquetado y esquemas de aprendizaje, sobre las cuales no se han hallado referencias de anteriores aplicaciones en el contexto temporal.

Se realizaron experimentos a partir de un total de cinco métodos de auto-etiquetado y tres esquemas de aprendizaje supervisado diferentes. Para aquellos métodos que siguen una metodología de envoltorio se exploraron todas las combinaciones posibles sumando un total de nueve métodos evaluados sobre 35 conjuntos de datos temporales. De cada conjunto de datos se establecieron tres proporciones distintas de ejemplos etiquetados (10%, 20% y 30%) respecto al total de ejemplos disponibles para entrenamiento. Para comparar los métodos, se utilizó fundamentalmente como medida de desempeño el promedio de exactitud de la clasificación. Estos valores se contrastaron utilizando pruebas estadísticas no paramétricas para comprobar las diferencias significativas entre los distintos métodos.

Después de analizar los resultados de la experimentación, podemos extraer las siguientes conclusiones:

- De forma general el 1NN es considerado una opción robusta de clasificador base para el contexto semi-supervisado ya que este ofrece los resultados más exactos de clasificación y no necesita realizar ajuste de parámetros. Este clasificador además aporta los tiempos de cómputo más competitivos.
- El método SelfT estándar no presenta un resultado competitivo y generalmente es superado por otros métodos de auto-etiquetado como el TriT y el SETRED.
- A partir del estudio empírico realizado se destacan tres métodos de auto-etiquetado en particular: SETRED, TriT-1NN y Democratic. Estos métodos presentaron resultados significativamente superiores al resto en términos de sus capacidades tanto inductivas como transductivas.

- La utilización de modelos de tipo multclasificador (TriT-1NN y Democratic) es una solución prometedora al problema del aprendizaje semi-supervisado en el contexto de las series temporales. Este resultado se encuentra a tono con estudios recientes [4, 100] desarrollados en el área de la clasificación supervisada de datos temporales. El mayor inconveniente de este enfoque es el aumento del costo computacional que implica el entrenamiento de varios clasificadores base.
- Teniendo en consideración los riesgos que conlleva la adición de ejemplos no etiquetados en el desempeño de la clasificación, se hace recomendable una comparación entre los resultados obtenidos con la SSC y un clasificador base con alto grado de exactitud como el 1NN. Esta comparación permite identificar los beneficios reales que representa el aprendizaje con ejemplos no etiquetados.

En términos generales, los resultados mostraron que los métodos de auto-etiquetado constituyen una opción factible para enfrentar problemas de SSC en el dominio temporal.

## Capítulo 3

# Influencia de la complejidad de los conjuntos de datos en el desempeño de los métodos de auto-etiquetado

Diversos estudios han puesto de manifiesto que los problemas reales de clasificación poseen estructuras, las cuales se encuentran estrechamente relacionadas con el desempeño alcanzable por los clasificadores que son entrenados para resolver dichos problemas. Para caracterizar la dificultad de un problema de clasificación supervisada, existe un enfoque que se basa en cuantificar la complejidad geométrica presente en la frontera entre las clases. Diversas de medidas de complejidad se han propuesto con este fin, las cuales permiten caracterizar diversos aspectos que determinan la complejidad final que presenta un problema de clasificación.

Aunque la mayoría de estos estudios se han enfocado en el paradigma supervisado, resulta interesante utilizar medidas de complejidad para evaluar la dificultad que presentan los ejemplos etiquetados que se utilizan para iniciar el entrenamiento semi-supervisado. Considerando que en este tipo de aprendizaje se encuentra limitado el número de ejemplos etiquetados, resulta de vital importancia evaluar la complejidad subyacente en dicho conjunto de entrenamiento desde

el punto de vista de la tarea de clasificación. Este capítulo tiene como objetivo realizar un estudio sobre la dificultad que enfrentan los métodos semi-supervisados de auto-etiquetado durante el proceso de entrenamiento en el contexto temporal, utilizando el enfoque de las medidas de complejidad. Además, se pretende estudiar la relación existente entre la complejidad inicial estimada y el desempeño obtenido por dichos métodos.

El capítulo se organiza en las siguientes secciones. En la sección 3.1 se mencionan los principales aspectos relacionados con la complejidad de la clasificación y se describen las cinco medidas de complejidad seleccionadas para desarrollar el presente estudio. En la sección 3.2 se presentan los resultados obtenidos de la evaluación de dichas medidas para diferentes proporciones de ejemplos etiquetados. En la sección 3.3 se efectúa el análisis sobre la influencia de la complejidad en el desempeño de los métodos de auto-etiquetado. Finalmente, en la sección 3.4 se presenta un resumen sobre las conclusiones principales derivadas del capítulo.

## 3.1 Complejidad de los datos

Existen varios factores que determinan la complejidad existente en un problema de clasificación [73]. Entre ellos se encuentra la ambigüedad existente entre las clases. Otra fuente usual de dificultad para el proceso de aprendizaje se relaciona con la complejidad de la frontera de decisión entre las clases, la cual incluye la forma en que las clases se separan o entremezclan [8]. Además, los conjuntos reducidos de ejemplos etiquetados y la alta dimensionalidad introducen otra capa adicional de complejidad. En problemas reales la complejidad está dada por la combinación de estos factores.

La caracterización de la complejidad geométrica de la frontera entre clases, estimada a partir de un conjunto de ejemplos etiquetados, resulta una opción viable para cuantificar la dificultad que presupone dicho problema de clasificación [71, 72]. En la literatura [73], se han propuesto varias medidas que permiten caracterizar este tipo de complejidad, las cuales se dividen en tres categorías:

- Medidas de solapamiento a nivel de atributos
- Medidas de solapamiento entre las clases

- Medidas de geometría, topología y densidad de las clases

El presente análisis se centra en el segundo y tercer grupo de medidas, teniendo en cuenta que en el contexto de la minería de datos temporales los puntos de datos que forman la serie no se consideran atributos independientes. Varias de estas medidas de complejidad basan su funcionamiento en la estimación de la disimilitud entre los ejemplos, lo cual las hace aplicables en dominios donde existan medidas apropiadas para calcular la diferencia entre los objetos. Teniendo en consideración este aspecto, se seleccionaron cinco medidas de complejidad a partir del estudio realizado por Basu y Ho en el año 2006 [8]. La notación asociada a cada medida se ha mantenido según el trabajo original. Las medidas seleccionadas se describen a continuación:

**Complejidad de la frontera (N1):** para calcular esta medida es necesario construir un árbol de expansión mínimo que abarque todo el conjunto de entrenamiento y contabilizar todos los arcos que conectan ejemplos de clases diferentes. Esta cantidad es normalizada por el número total de arcos presentes en el árbol. Esta medida toma valores en el intervalo  $(0, 1]$ , donde los valores más cercanos a uno implican una mayor proporción de ejemplos en la frontera y por tanto una mayor complejidad.

**Ratio entre las distancias internas y externas a la clase (N2):** para evaluar esta medida se calcula para cada ejemplo  $x_i$  del conjunto de entrenamiento compuesto por  $N$  ejemplos, la distancia ( $d$ ) con su vecino más cercano  $x_j$  dentro y fuera de la clase  $y_i$  a la que pertenece el ejemplo  $x_i$ . Las distancias tanto internas como externas de todos los ejemplos de entrenamiento son promediadas de forma independiente. El ratio entre ambos promedios es considerado como valor final de la medida, siendo este un número estrictamente mayor que cero. Los valores cercanos a cero implican que existe una diferencia considerable entre las distancias internas y externas, lo cual se interpreta como una complejidad baja.

$$N2 = \frac{\sum_{i=1}^N \min_{\{j \in \mathcal{X} | y_i = y_j, i \neq j\}} d(x_i, x_j)}{\sum_{i=1}^N \min_{\{j \in \mathcal{X} | y_i \neq y_j, i \neq j\}} d(x_i, x_j)} \quad (3.1)$$

**Error mediante la validación LOO (*Leave-One-Out Error*) (N3):** se considera este error como una medida de complejidad ya que resulta útil

para detectar ejemplos que se encuentran irregularmente alejados del resto (*outliers*). Para el cálculo de este error se crean tantas particiones como ejemplos existentes ( $N$ ), y se aplica el esquema clásico de validación cruzada. Los errores obtenidos en cada partición son promediados tomando un valor final en el rango  $[0, 1]$ .

$$N3 = \frac{1}{N} \sum_{i=1}^N Error_i \quad (3.2)$$

**Descripción interna de clases (T1):** esta medida, originada en el trabajo de Frank y Hubert [56], utiliza la noción de subconjuntos de adherencia en una pretopología (*pretopology*). La idea consiste en evaluar la forma en que se distribuyen los ejemplos dentro de cada clase. Para esto se calcula la cantidad de  $\epsilon$ -vecindades necesarias para cubrir todas las clases del problema. Cada  $\epsilon$ -vecindad se encuentra centrada en un determinado ejemplo y crece mediante el factor  $\epsilon$  hasta un tamaño máximo donde la vecindad no toque ningún ejemplo perteneciente a otra clase. La cantidad final de  $\epsilon$ -vecindades se normaliza por el número total de ejemplos. Toma valores en el rango  $(0, 1]$  y los valores cercanos a uno implican que las clases no presentan una descripción compacta y por tanto la complejidad es elevada.

**Cantidad de ejemplos por atributos (T2):** esta medida calcula el ratio entre la cantidad de ejemplos en el conjunto de entrenamiento ( $N$ ) y la dimensión del problema ( $D$ ). Toma valores estrictamente mayores que cero y mientras más elevados son estos valores se considera que la complejidad es menor.

$$T2 = \frac{N}{D} \quad (3.3)$$

Las medidas N1, N2 y N3 se consideran basadas en la regla del vecino más cercano. En el caso particular de la medida T2, la cantidad de atributos se evalúa como la longitud de las series temporales. Con la excepción de esta última medida, las restantes se calculan utilizando un enfoque basado en el cálculo de la disimilitud entre los ejemplos.

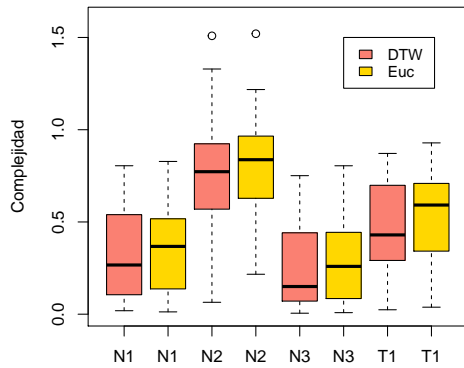


## 3.2 Evaluación de las medidas de complejidad seleccionadas

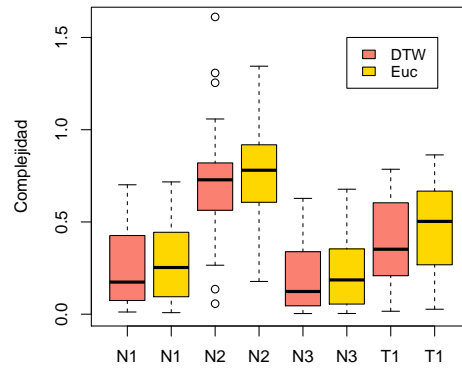
Para efectuar la evaluación de las medidas de complejidad se utilizaron los 35 conjuntos de datos que fueron descritos con anterioridad en la sección 2.2.1. Las particiones de entrenamiento involucradas en el presente estudio coinciden con las utilizadas en el Capítulo 2. Como se describió anteriormente, dichos conjuntos de datos fueron divididos en cinco particiones con el objetivo de aplicar validación cruzada. Cada partición de entrenamiento está dividida en los conjuntos:  $L$  y  $U$ . Debido a que las medidas de complejidad seleccionadas fueron propuestas en el contexto del aprendizaje supervisado (tienen en cuenta el valor del atributo clase durante su cálculo), estas son evaluadas solamente en el conjunto  $L$ . Los resultados obtenidos en cada partición son promediados entre sí. Se evaluaron diferentes proporciones de ejemplos etiquetados (10 %, 20 % y 30 %) respecto a la cantidad disponible, con el propósito de estudiar la influencia que ejerce dicho valor en los resultados de complejidad estimados. Para el cálculo de la disimilitud entre los ejemplos se utilizaron las medidas euclidiana y DTW. El Apéndice C incluye los resultados completos de todas las evaluaciones realizadas.

La Figura 3.1 muestra los resultados de las medidas de complejidad mediante gráficos de caja. De esta figura se excluyó la medida T2 por presentar un rango de valores mucho más amplio que el resto de las medidas. En este gráfico puede observarse cómo la mediana de las medidas de complejidad son menores para DTW que para la distancia euclidiana. Otro aspecto interesante es que la complejidad tiende a ser menor en la medida que aumentan los ejemplos etiquetados.

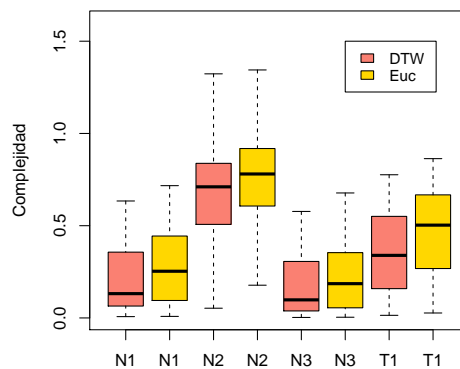
Para evaluar si la diferencia de complejidad que se percibe a través de la medida DTW es menor que con la distancia euclidiana, se aplicó la prueba de signos de Wilcoxon entre los valores de complejidad obtenidos para cada medida de disimilitud y proporciones de ejemplos etiquetados. Los resultados se muestran en la Tabla 3.1, donde en la mayoría de las comparaciones el uso de DTW implica que la complejidad observada en los conjuntos de datos sea significativamente menor que la observada con la distancia euclidiana. La medida T2 se excluye de las comparaciones ya que no es una medida basada en distancias.



(a) 10 % ejemplos etiquetados.



(b) 20 % ejemplos etiquetados.



(c) 30 % ejemplos etiquetados.

Figura 3.1: Diagrama de cajas que muestra los resultados de cuatro medidas de complejidad para diferentes proporciones de ejemplos etiquetados y las medidas de disimilitud euclidiana y DTW.

A continuación se muestra de forma particular el comportamiento de los valores de complejidad para seis conjuntos de datos específicos. El objetivo de este análisis es mostrar los diversos grados de complejidad que pueden observarse entre los conjuntos de datos experimentales. Las Figuras 3.2 y 3.3 muestran la complejidad que refleja cada conjunto de datos a través de las medidas de complejidad evaluadas para cada proporción de ejemplos etiquetados.

DTW–Euc	10 %			20 %			30 %		
	Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$	Neg	Pos	$p_{valor}$
N1	<b>26</b>	9	$6.1 \cdot 10^{-4}$	<b>26</b>	9	$2.0 \cdot 10^{-4}$	<b>30</b>	5	$5.0 \cdot 10^{-7}$
N2	20	15	0.41	21	14	0.18	<b>27</b>	8	$7.3 \cdot 10^{-4}$
N3	<b>24</b>	11	$1.0 \cdot 10^{-3}$	<b>25</b>	9	$8.0 \cdot 10^{-4}$	<b>31</b>	4	$4.0 \cdot 10^{-6}$
T1	<b>27</b>	7	$1.5 \cdot 10^{-4}$	<b>25</b>	9	$3.4 \cdot 10^{-5}$	<b>31</b>	4	$2.6 \cdot 10^{-8}$

Tabla 3.1: Prueba de signos de Wilcoxon aplicada a cuatro medidas de complejidad donde se comparan los resultados entre euclidiana y DTW. De cada comparación se muestra la cantidad de rangos positivos y negativos junto al  $p$ -valor.

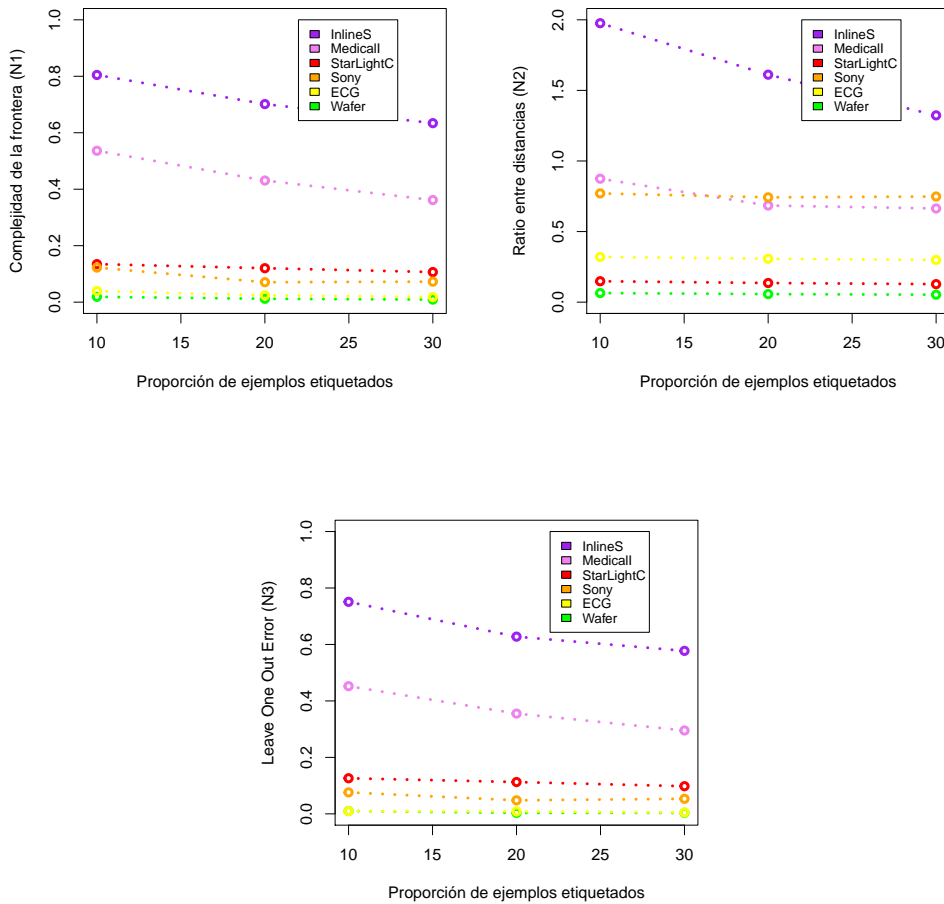


Figura 3.2: Evaluación de las medidas N1, N2 y N3 en seis conjuntos. Estas se evalúan para cada proporción de ejemplos etiquetados usando DTW.

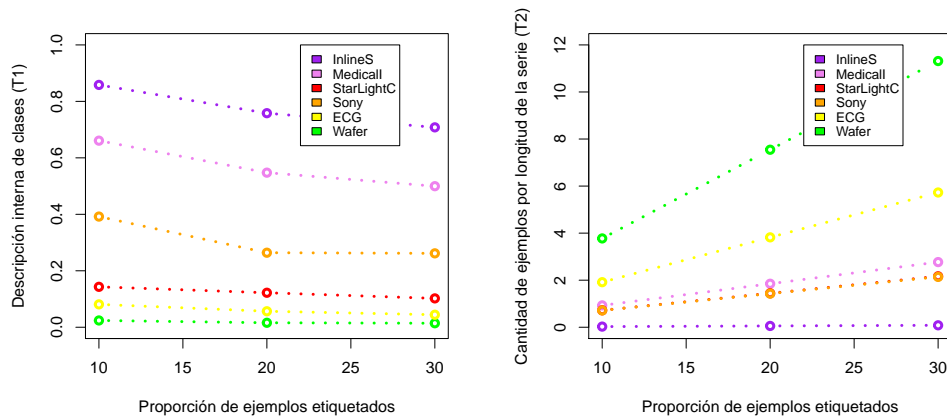


Figura 3.3: Evaluación de las medidas T1 y T2 en seis conjuntos de datos particulares, usando DTW. Se muestran los resultados para cada proporción de ejemplos etiquetados.

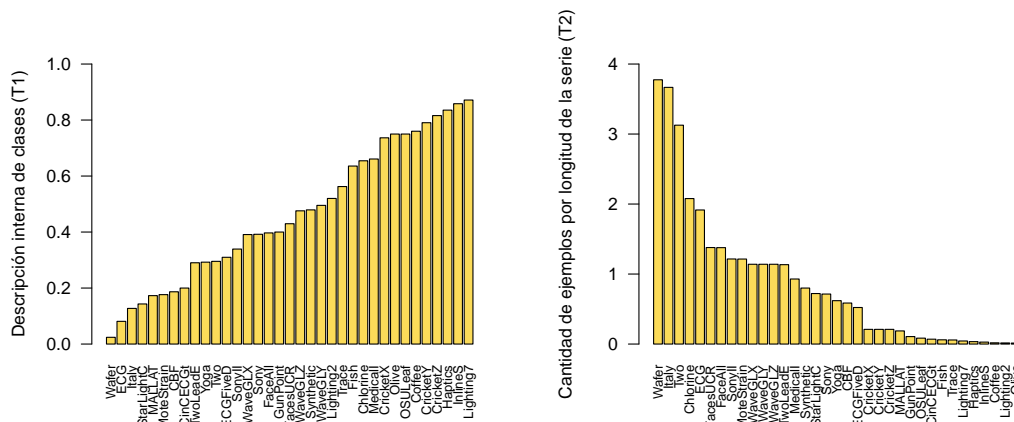


Figura 3.4: Ordenamiento ascendente en complejidad de todos los conjuntos de datos en base a las medidas T1 y T2, usando DTW y el 10% de ejemplos etiquetados.

El conjunto de datos InlineS es el de mayor complejidad según el criterio común de las cinco medidas evaluadas. Este problema tiene siete clases y un conjunto reducido de ejemplos (650) en comparación con su alta dimensión (1882 de longitud en las series), aspectos que incrementan la complejidad de la clasificación. Por el contrario el conjunto de datos Wafer es un problema de sólo dos



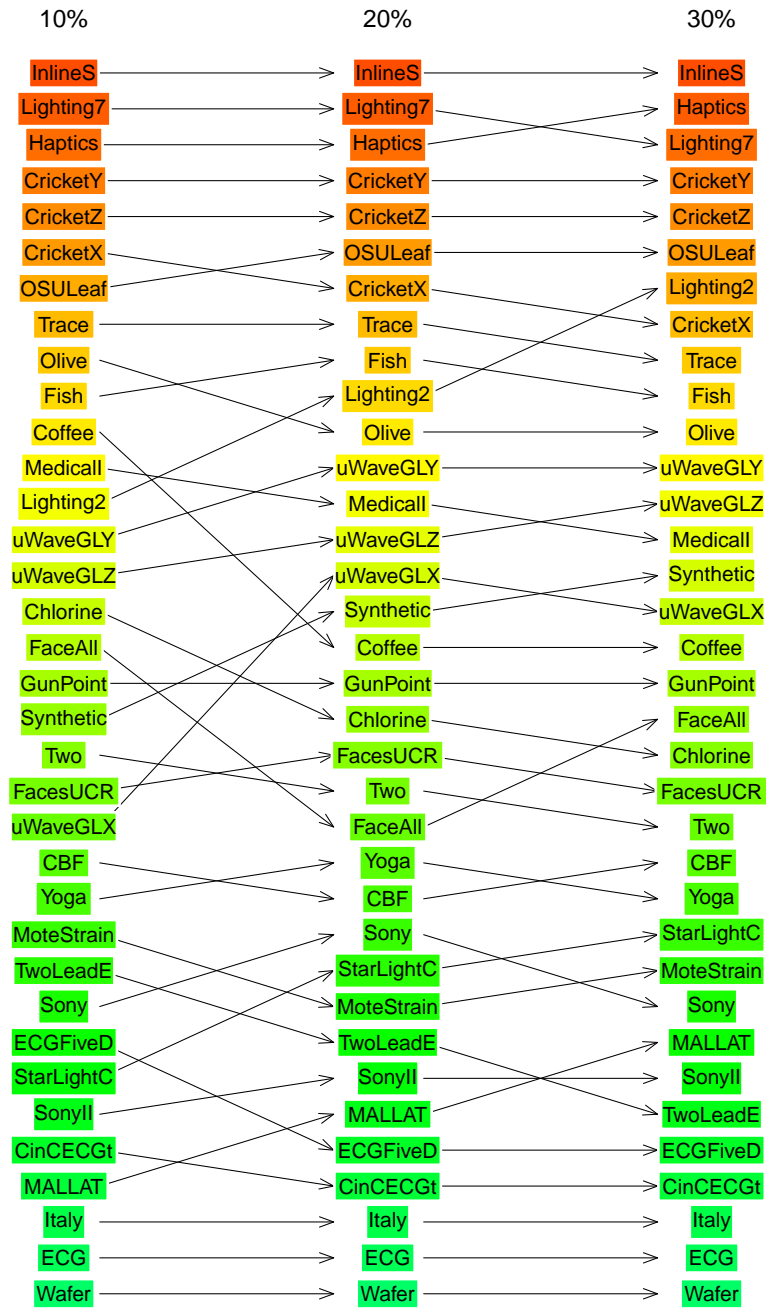


Figura 3.6: Ordenamiento ascendente en complejidad según el Conteo de Borda aplicado a los rankings individuales por cada medida de complejidad usando la distancia euclidiana.

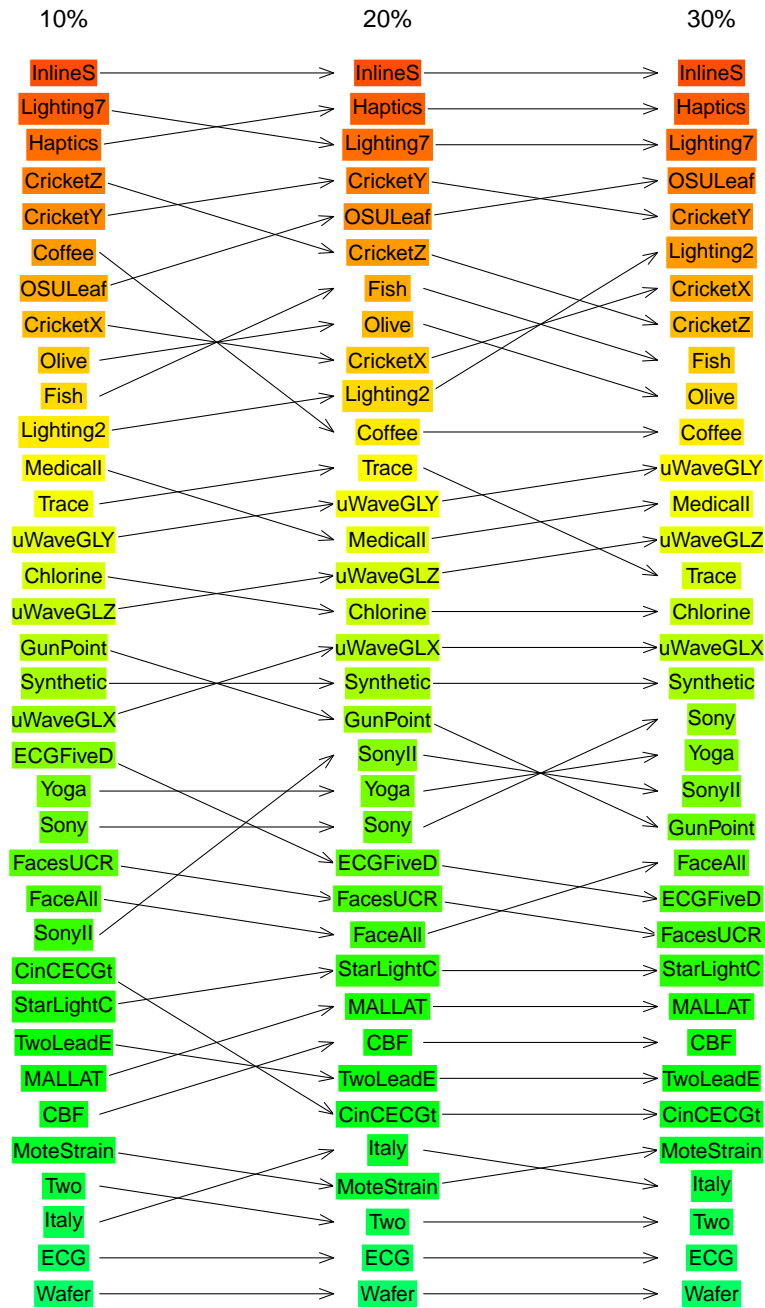


Figura 3.7: Ordenamiento ascendente en complejidad según el Conteo de Borda aplicado a los rankings individuales por cada medida de complejidad usando la medida DTW.

Con el objetivo de mostrar un consenso entre las cinco medidas, se aplicó el Conteo de Borda [51] para obtener un ordenamiento ascendente en complejidad de los 35 conjuntos de datos involucrados en el estudio. El Conteo de Borda es un método simple de selección donde a cada candidato se le asigna un número de puntos en correspondencia con el lugar que ocupa en cada ranking. Por ocupar el último lugar en el ranking recibe un punto, por ocupar el penúltimo recibe dos puntos y así sucesivamente. Por ejemplo, el conjunto de datos InlineS para el 10 % utilizando DTW (Figuras 3.4 y 3.5) aparece tres veces en el último puesto, una en el penúltimo y una en el puesto número 32. Esto suma un total de nueve puntos para este conjunto de datos. Para obtener un ranking final de los candidatos se suman todos los puntos obtenidos y se ordenan de forma ascendente.

Las Figuras 3.6 y 3.7 muestran el orden final que ocupa cada conjunto de datos para ambas medidas de disimilitud y las diferentes proporciones de ejemplos etiquetados evaluadas. La posición inferior se relaciona con el conjunto de datos menos complejo y la superior con el más complejo. Las flechas muestran las diferencias entre los ordenamientos de cada proporción. Se puede apreciar que para la mayoría de los conjuntos de datos no existen grandes diferencias en el orden obtenido entre las proporciones. Específicamente, las diferencias son menores entre las proporciones 20 % y 30 %. De forma general, el comportamiento de los ordenamientos obtenidos apunta a que las medidas de complejidad estudiadas pueden resultar informativas aún cuando se aplican a conjuntos de datos reducidos.

### 3.2.1 Análisis de los tiempos de ejecución

El tiempo de ejecución necesario para la evaluación de las medidas de complejidad, depende directamente de la cantidad de ejemplos que componen el conjunto de datos. Las medidas de complejidad recibieron como entrada, para el proceso de evaluación, la matriz de distancia entre todos los ejemplos disponibles. Por este motivo el consumo de tiempo asociado al cálculo de las distancias no se contempla en el análisis que se presenta a continuación. Teniendo en cuenta sólo la complejidad temporal de la evaluación de las medidas, estas presentan una complejidad teórica de orden polinomial en todos los casos.



A continuación se presenta un análisis empírico sobre el tiempo de cómputo consumido, basado en una muestra de 12 conjuntos de datos que formaron parte de los experimentos. Al igual que los experimentos del Capítulo 2, las ejecuciones se realizaron en un clúster que contiene 46 nodos, cada uno de ellos equipado con un procesador Intel® Core™ i7-930 a 2.8 GHz y 24 GB de memoria RAM. La Tabla 3.2 muestra los tiempos de cómputo obtenidos durante la evaluación de las medidas de complejidad de forma secuencial en las cinco particiones existentes. Los menores tiempos de ejecución se obtuvieron con la medida T2, ya que esta no depende de la cantidad de ejemplos involucrados. De manera general, las medidas presentan tiempos de cómputo competitivos lo cual facilita su aplicación.

Datos	N1	N2	N3	T1	T2
Coffee	0.014	<b>0.012</b>	<b>0.012</b>	0.014	<b>0.012</b>
Fish	0.033	0.018	0.014	0.031	<b>0.011</b>
Gun_Point	0.021	0.016	0.013	0.024	<b>0.012</b>
InlineS	0.179	0.023	0.012	0.080	<b>0.009</b>
Italy	0.237	0.039	0.016	0.126	<b>0.012</b>
Lighting2	0.015	0.014	0.012	0.019	<b>0.011</b>
Lighting7	0.019	0.015	0.012	0.020	<b>0.011</b>
Olive	0.015	0.014	0.013	0.016	<b>0.011</b>
OSULeaf	0.049	0.020	0.014	0.053	<b>0.012</b>
SonyII	0.209	0.038	0.016	0.169	<b>0.0109</b>
Trace	0.020	0.015	0.013	0.024	<b>0.012</b>
Yoga	1.454	0.137	0.042	1.71	<b>0.013</b>

Tabla 3.2: Consumo de tiempo (segundos) de cómputo durante la evaluación de las medidas de complejidad para el 10% de ejemplos etiquetados con la distancia euclidiana.

### 3.3 Influencia de la complejidad en los resultados de clasificación

Las características geométricas con que se distribuyen las clases en los conjuntos de datos influyen en el rendimiento de los métodos de SSC. Este paradig-

ma resulta especialmente sensible a los efectos de la complejidad en los conjuntos de datos, pues la incorporación de ejemplos mal clasificados en el conjunto de entrenamiento durante la etapa inicial provoca el problema conocido como reforzamiento del error. Este aspecto ha sido abordado en la literatura, por ejemplo en el trabajo de Wang y colaboradores [155] se investiga el solapamiento de clases en aras de ofrecer una explicación sobre el pobre desempeño que experimentan algunos métodos de SSC en conjuntos de datos donde se manifiesta este problema.

La presente sección tiene como objetivo estudiar la influencia que ejerce la complejidad de los problemas temporales (evaluada a través de las medidas seleccionadas) en el desempeño de los métodos de auto-etiquetado. Con este fin, se han incluido los resultados inductivos obtenidos con dichos métodos, los cuales fueron discutidos previamente en el Capítulo 2. Como se mencionó anteriormente, las particiones de entrenamiento utilizadas en dicho capítulo son las mismas que se emplearon para evaluar la complejidad de los conjuntos etiquetados. El análisis de los resultados obtenidos en la clasificación abarca tanto la exactitud como la ganancia que representa la inclusión de ejemplos no etiquetados en el aprendizaje.

### **3.3.1 Análisis de la exactitud**

Para dar inicio al análisis sobre la relación existente entre la complejidad y los valores de exactitud obtenidos, se muestran a modo de ejemplo los gráficos de dispersión obtenidos entre la exactitud alcanzada con el método SETRED y los valores de complejidad estimados con las cinco medidas estudiadas (Figura 3.8). Según el estudio efectuado en la sección 2.3 este método resultó ser uno de los más competitivos. Los gráficos muestran además la recta de regresión lineal obtenida en cada caso.

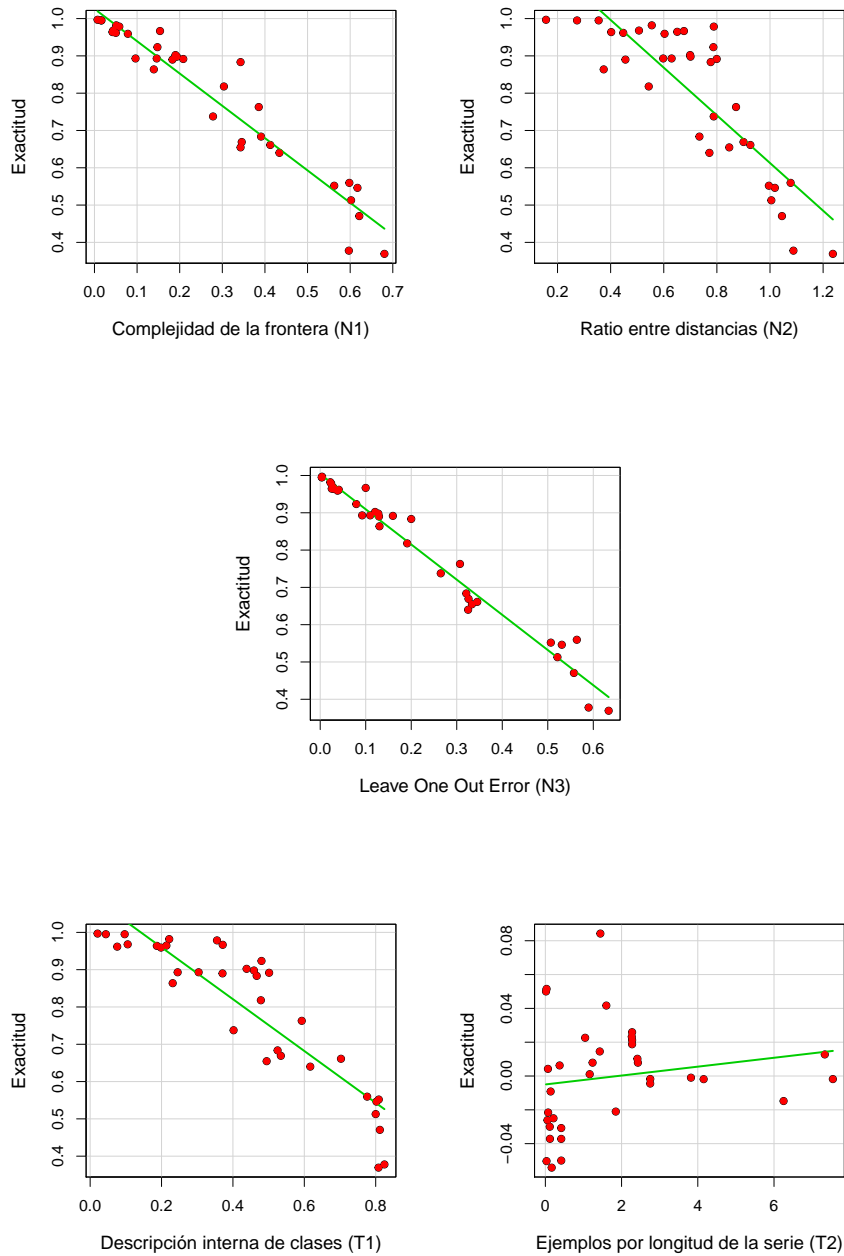


Figura 3.8: Gráfico de dispersión entre la exactitud obtenida con el SETRED y la complejidad estimada, con el 30% de ejemplos etiquetados y la distancia euclidiana.

De manera general, es posible apreciar la correlación inversa existente entre la exactitud y los valores de las medidas N1, N2, N3 y T1. Esto indica que

la alta complejidad de los conjuntos de datos se relaciona con bajos valores de exactitud alcanzada mediante los métodos de auto-etiquetado. Respecto a la medida T2, la correlación resulta positiva pues los valores altos de esta media se relacionan con un aumento de la exactitud. Esta última medida muestra un nivel de correlación más débil dado el alto grado de dispersión que se observa en el gráfico correspondiente.

Con el objetivo de cuantificar la relación existente entre la exactitud de los métodos de auto-etiquetado y los valores estimados de complejidad, se aplicó la prueba no paramétrica conocida como coeficiente de correlación de Spearman. Las Tablas 3.3, 3.4, 3.5, 3.6 y 3.7 muestran los coeficientes de correlación obtenidos. La correlación se estimó para cada medida de distancia y cada proporción de ejemplos etiquetados.

El  $p_{valor}$  resulta significativo en todas las correlaciones que involucran las medidas N1, N2, N3 y T1, para un nivel de significación de  $\alpha = 0.05$ . En el caso de la medida T2, el  $p_{valor}$  no es significativo en algunas de las correlaciones, por ejemplo en aquellas donde están involucrados los árboles de decisión como clasificadores base.

Algoritmo	Euclidiana			DTW		
	10 %	20 %	30 %	10 %	20 %	30 %
SelfT-1NN	-0.90	-0.96	-0.95	-0.89	-0.97	-0.93
SETRED	-0.92	-0.96	-0.95	-0.92	-0.95	-0.95
SNNRCE	-0.90	-0.97	-0.94	-0.91	-0.96	-0.94
TriT-1NN	-0.90	-0.96	-0.94	-0.91	-0.95	-0.95
Democratic	-0.92	-0.95	-0.96	-0.90	<b>-0.96</b>	<b>-0.92</b>
SelfT-SVM	-0.90	-0.91	<b>-0.95</b>	<b>-0.90</b>	<b>-0.91</b>	<b>-0.92</b>
TriT-SVM	-0.93	-0.95	<b>-0.97</b>	-0.93	<b>-0.96</b>	<b>-0.94</b>
SelfT-DT	-0.83	-0.85	-0.89	<b>-0.78</b>	-0.83	-0.78
TriT-DT	-0.84	-0.90	-0.91	<b>-0.81</b>	-0.84	-0.83
media	-0.89	-0.93	-0.94	-0.86	-0.92	<b>-0.91</b>

Tabla 3.3: Coeficiente de correlación ( $\rho$ ) de Spearman entre la exactitud obtenida por cada método y la complejidad estimada mediante la medida N1.

En cada tabla se ha incluido además una fila con las medias por cada proporción de ejemplos etiquetados con el fin de resaltar<sup>1</sup> las medidas donde se obtiene una correlación más fuerte. La más destacada en este aspecto resulta la medida N3 seguida por la N1.

Las medidas N3 y N1 resultan particularmente efectivas para evaluar la complejidad en aquellos métodos que tienen el algoritmo 1NN como clasificador base. Esto se debe a que dichas medidas están enfocadas en los aspectos que le añaden complejidad a la clasificación mediante el 1NN y en efecto la correlación inversa que se obtiene es muy fuerte. Entre ellas la más destacada para este tipo de métodos resulta la N3. Por su parte, la medida N1 parece captar bien las características que dificultan la clasificación mediante SVMs cuando se utiliza DTW. Adicionalmente, la medida T1 parece ser la más adecuada para evaluar la complejidad en los métodos que tienen el algoritmo DT como clasificador base.

Algoritmo	Euclidiana			DTW		
	10 %	20 %	30 %	10 %	20 %	30 %
SelfT-1NN	-0.87	-0.87	-0.84	-0.83	-0.78	-0.73
SETRED	-0.87	-0.86	-0.81	-0.83	-0.75	-0.71
SNNRCE	-0.86	-0.87	-0.83	-0.83	-0.77	-0.77
TriT-1NN	-0.85	-0.86	-0.82	-0.83	-0.74	-0.74
Democratic	-0.89	-0.88	-0.82	-0.82	-0.75	-0.69
SelfT-SVM	-0.90	-0.85	-0.85	-0.87	-0.76	-0.74
TriT-SVM	-0.89	-0.86	-0.83	-0.87	-0.73	-0.70
SelfT-DT	-0.83	-0.87	-0.86	-0.77	-0.76	-0.71
TriT-DT	-0.84	-0.89	-0.86	-0.79	-0.79	-0.76
media	-0.87	-0.87	-0.84	-0.82	-0.76	-0.73

Tabla 3.4: Coeficiente de correlación ( $\rho$ ) de Spearman entre la exactitud obtenida por cada método y la complejidad estimada mediante la medida N2.

<sup>1</sup>Se han resaltado en negrita los valores más fuertes de correlación entre todas las medidas de complejidad, para cada método y proporción de ejemplos etiquetados.

Algoritmo	Euclidiana			DTW		
	10 %	20 %	30 %	10 %	20 %	30 %
SelfT-1NN	<b>-0.92</b>	<b>-0.98</b>	<b>-0.98</b>	<b>-0.91</b>	<b>-0.97</b>	<b>-0.93</b>
SETRED	<b>-0.94</b>	<b>-0.98</b>	<b>-0.98</b>	<b>-0.95</b>	<b>-0.97</b>	<b>-0.97</b>
SNNRCE	<b>-0.91</b>	<b>-0.98</b>	<b>-0.96</b>	<b>-0.93</b>	<b>-0.97</b>	<b>-0.96</b>
TriT-1NN	<b>-0.92</b>	<b>-0.98</b>	<b>-0.98</b>	<b>-0.93</b>	<b>-0.97</b>	<b>-0.97</b>
Democratic	<b>-0.94</b>	<b>-0.96</b>	<b>-0.97</b>	<b>-0.92</b>	-0.96	-0.91
SelfT-SVM	<b>-0.92</b>	<b>-0.92</b>	-0.93	-0.88	-0.90	-0.89
TriT-SVM	<b>-0.95</b>	<b>-0.95</b>	-0.96	<b>-0.95</b>	-0.96	-0.93
SelfT-DT	-0.82	-0.84	-0.83	-0.75	-0.82	-0.74
TriT-DT	-0.84	-0.89	-0.87	-0.78	-0.83	-0.80
media	<b>-0.91</b>	<b>-0.94</b>	<b>-0.94</b>	<b>-0.89</b>	<b>-0.93</b>	-0.90

Tabla 3.5: Coeficiente de correlación ( $\rho$ ) de Spearman entre la exactitud obtenida por cada método y la complejidad estimada mediante la medida N3.

Algoritmo	Euclidiana			DTW		
	10 %	20 %	30 %	10 %	20 %	30 %
SelfT-1NN	-0.86	-0.91	-0.92	-0.83	-0.90	-0.85
SETRED	-0.87	-0.91	-0.91	-0.84	-0.86	-0.85
SNNRCE	-0.86	-0.93	-0.93	-0.84	-0.89	-0.88
TriT-1NN	-0.85	-0.91	-0.91	-0.84	-0.84	-0.86
Democratic	-0.89	-0.93	-0.92	-0.82	-0.89	-0.85
SelfT-SVM	-0.89	-0.89	-0.93	-0.87	-0.87	-0.88
TriT-SVM	-0.89	-0.92	-0.94	-0.88	-0.87	-0.87
SelfT-DT	<b>-0.85</b>	<b>-0.89</b>	<b>-0.92</b>	-0.78	<b>-0.87</b>	<b>-0.82</b>
TriT-DT	<b>-0.86</b>	<b>-0.92</b>	<b>-0.93</b>	-0.81	<b>-0.88</b>	<b>-0.87</b>
media	-0.87	-0.91	-0.92	-0.83	-0.87	-0.86

Tabla 3.6: Coeficiente de correlación ( $\rho$ ) de Spearman entre la exactitud obtenida por cada método y la complejidad estimada mediante la medida T1.

Algoritmo	10 %	20 %	30 %
SelfT-1NN	0.40	0.42	0.39
SETRED	0.36	0.42	0.42
SNNRCE	0.43	0.47	0.38
TriT-1NN	0.33 •	0.42	0.39
Democratic	0.37	0.42	0.40
SelfT-SVM	0.43	0.37	0.39
TriT-SVM	0.41	0.40	0.40
SelfT-DT	0.26 •	0.26 •	0.29 •
TriT-DT	0.28 •	0.28 •	0.30 •
media	0.36	0.38	0.37

Tabla 3.7: Coeficiente de correlación ( $\rho$ ) de Spearman entre la exactitud por cada método y la complejidad estimada con la medida T2. Las correlaciones que no son significativas aparecen marcadas con (•).

La medida N2, aunque competitiva, no presenta una correlación destacada con ningún método en particular. La medida T2 resulta la de menor correlación con la exactitud de los métodos por lo que podemos concluir que la alta dimensionalidad que caracteriza los problemas temporales no es el aspecto con mayor efecto en el comportamiento de los métodos de auto-etiquetado.

De manera general las medidas de complejidad han mostrado una correlación significativa con la exactitud obtenida mediante los métodos de auto-etiquetado estudiados. El clasificador base utilizado parece ser un factor clave en la determinación de cuál medida ofrece mayor información sobre complejidad de los problemas de clasificación. En resumen, las medidas más destacadas resultaron ser: N1, N3 y T1, las cuales en conjunto obtuvieron los más altos grados de correlación con los métodos de auto-etiquetado estudiados. Esto implica que la extensión de la frontera, los *outliers* y la dispersión que presentan los ejemplos dentro de cada clase, son los elementos que afectan de forma más significativa la exactitud de la clasificación mediante métodos de auto-etiquetado.

### 3.3.2 Análisis de la ganancia de exactitud

La ganancia de exactitud obtenida al utilizar métodos de auto-etiquetados es otro factor a tener en cuenta al analizar el efecto que tiene la complejidad en esta clase de métodos. La ganancia, tal y como se definió en la Expresión 2.4, se calcula al sustraer la exactitud obtenida con el clasificador supervisado 1NN (entrenado en el conjunto  $L$  de ejemplos etiquetados) de la exactitud alcanzada mediante clasificación semi-supervisada, utilizando todo el conjunto de entrenamiento.

La Tabla 3.8 resume los máximos<sup>2</sup> valores de correlación obtenidos mediante la prueba de Spearman entre la ganancia de exactitud y las cinco medidas de complejidad evaluadas para cada método de auto-etiquetado. También se incluye en cada caso la medida de complejidad correspondiente con la correlación mostrada. A diferencia del análisis realizado con los valores de exactitud, la correlación obtenida entre la ganancia y la complejidad no es fuerte, alcanzando  $p$ valores no significativos en varios de los métodos estudiados. Esto ocurre principalmente para la proporción del 10 % de ejemplos etiquetados. También existe diversidad en cuanto a la medida de complejidad que favorece la correlación más fuerte obtenida y el signo de la misma. De manera general, la medida T1 es la que con mayor frecuencia obtiene los valores absolutos más elevados respecto al resto de las medidas evaluadas.

---

<sup>2</sup>Los valores máximos de correlación se calcularon respecto al valor absoluto de correlación obtenido, por tanto los valores mostrados pueden ser tanto positivos como negativos.



Algoritmo	10 %			20 %			30 %			
	$\rho$		M.	$\rho$		M.	$\rho$		M.	
<b>Euc</b>	SelfT-1NN	0.09	•	T2	-0.11	•	N2	-0.27	•	N3
	SETRED	0.24	•	T1	0.33		T1	0.35		T1
	SNNRCE	0.38		N3	0.51		N1	0.64		N3
	TriT-1NN	-0.20	•	T2	-0.34		N3	-0.13	•	N3
	Democratic	-0.43		T1	-0.53		T1	-0.34		N1
	SelfT-SVM	-0.63		N2	-0.57		T1	-0.58		N1
	TriT-SVM	-0.60		N3	-0.58		N1	-0.47		N1
	SelfT-DT	-0.25	•	T2	-0.35		N2	-0.48		N2
	TriT-DT	-0.29	•	T1	-0.48		N2	-0.46		N2
Algoritmo	10 %			20 %			30 %			
	$\rho$		M.	$\rho$		M.	$\rho$		M.	
<b>DTW</b>	SelfT-1NN	-0.34		N3	-0.39		N2	-0.36		N1
	SETRED	-0.21	•	T2	0.16	•	T1	0.32	•	N2
	SNNRCE	0.45		N3	0.57		N3	0.46		N3
	TriT-1NN	0.14	•	T2	-0.18	•	N3	-0.31	•	N3
	Democratic	-0.32	•	N1	-0.43		N1	-0.63		N1
	SelfT-SVM	-0.60		T1	-0.50		T1	-0.63		T1
	TriT-SVM	-0.60		T1	-0.57		N1	-0.57		T1
	SelfT-DT	-0.23	•	N2	-0.35		T1	-0.47		T1
	TriT-DT	-0.27	•	T1	-0.41		T1	-0.51		T1

Tabla 3.8: Coeficiente de correlación ( $\rho$ ) de Spearman de máximo valor absoluto entre la ganancia de exactitud por cada método y las medidas de complejidad evaluadas. Se incluye la medida que aportó dicho valor máximo. Las correlaciones que no son significativas aparecen marcadas con (•).

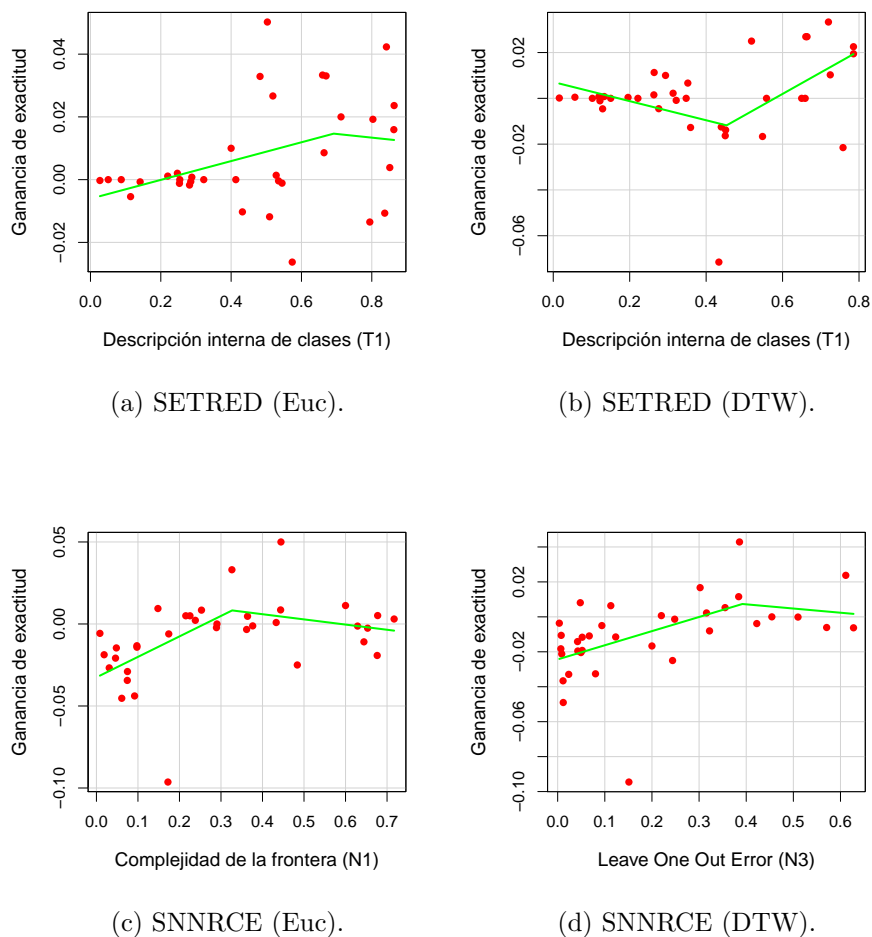


Figura 3.9: Gráfico de dispersión entre la exactitud obtenida con los métodos SETRED y SNNRCE y la complejidad estimada, para el 20% de ejemplos etiquetados y ambas medidas de distancia.

Los métodos de auto-etiquetado con mecanismos de adición de tipo modificable (SETRED y SNNRCE) presentan en la mayoría de los casos una correlación positiva. Para analizar más profundamente este comportamiento en la Figura 3.9 se muestran los gráficos de dispersión que se obtienen con estos métodos para el 20% de los ejemplos etiquetados y la distancia euclidiana. En cada gráfico se muestra además una o varias rectas de ajuste lineal por segmentos con el objetivo de describir de una forma más precisa la relación existente entre las variables analizadas. A diferencia del análisis realizado respecto a la exactitud, la relación entre la ganancia y la complejidad no mantiene un comportamiento lineal esta-

ble a lo largo de todo el rango de valores estudiados. La medida de complejidad mostrada en cada gráfico se corresponde con la que obtuvo mayor correlación (Tabla 3.8).

El método SETRED no obtiene ganancias apreciables de exactitud para los conjuntos de datos que presentan baja complejidad ( $T1 < 0.20$ ). En estos casos el conjunto inicial  $L$ , suministrado para el entrenamiento, resulta ser suficiente para que el 1NN aprenda una hipótesis adecuada para efectuar la clasificación y por tanto no se obtienen ventajas de incluir ejemplos no etiquetados en el entrenamiento. Para valores de mayor complejidad (estimados mediante la medida  $T1$ ) comienza a apreciarse en la mayoría de los casos una ganancia positiva de utilizar el método SETRED, pues el aporte de los ejemplos no etiquetados comienza a ser valioso. En una minoría de los conjuntos de datos, la hipótesis aprendida se deteriora con el aprendizaje semi-supervisado.

Respecto al método SNNRCE sucede algo similar pero para los conjuntos de datos poco complejos se obtienen ganancias en extremo negativas. Esta situación comienza a revertirse para valores de complejidad más elevados. Con este método prácticamente no se obtienen ganancias positivas apreciables.

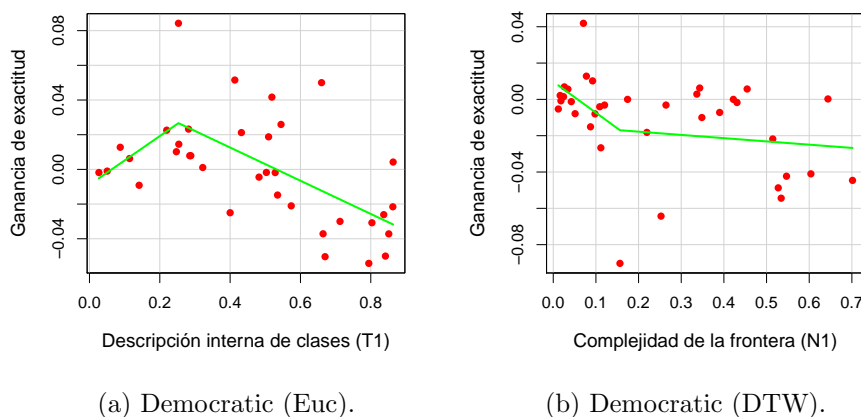
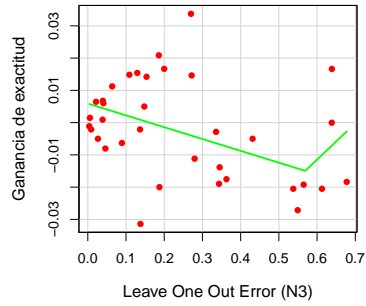
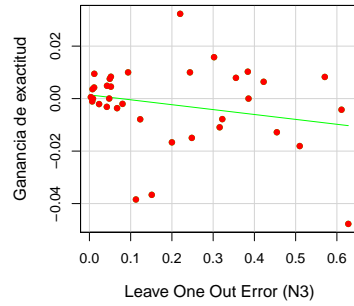


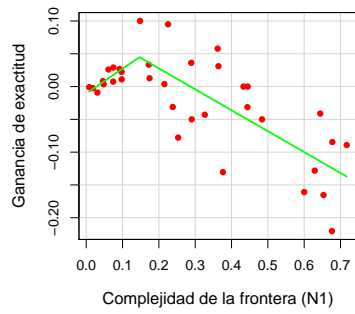
Figura 3.10: Gráfico de dispersión entre la exactitud obtenida con el método Democratic y la complejidad estimada, para el 20 % de ejemplos etiquetados y ambas medidas de distancia.



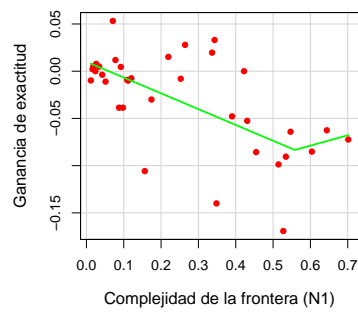
(a) Tri-training-1NN (Euc).



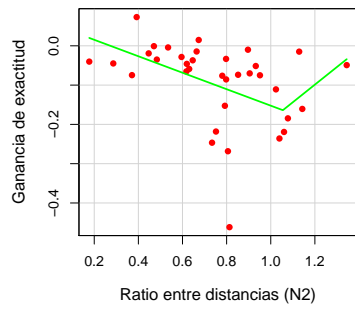
(b) Tri-training-1NN (DTW).



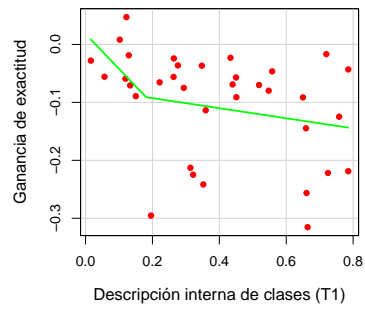
(c) Tri-training-SVM (Euc).



(d) Tri-training-SVM (DTW).



(e) Tri-training-DT (Euc).



(f) Tri-training-DT (DTW).

Figura 3.11: Gráfico de dispersión entre la exactitud obtenida con el método Tri-training y la complejidad estimada, con el 20% de ejemplos etiquetados y ambas medidas de distancia.

Algoritmo	Euclidiana			DTW		
	10 %	20 %	30 %	10 %	20 %	30 %
SelfT-1NN	0.023	0.020	0.012	0.032	0.027	0.021
SETRED	0.016	<b>0.011</b>	<b>0.011</b>	<b>0.012</b>	<b>0.011</b>	<b>0.007</b>
SNNRCE	<b>0.013</b>	0.015	0.012	0.013	0.013	0.011
TriT-1NN	<b>0.013</b>	<b>0.011</b>	0.014	0.014	<b>0.011</b>	0.012
Democratic	0.019	0.020	0.016	0.019	0.016	0.013
SelfT-SVM	0.076	0.066	0.053	0.082	0.067	0.056
TriT-SVM	0.035	0.041	0.037	0.037	0.027	0.035
SelfT-DT	0.076	0.080	0.073	0.080	0.079	0.074
TriT-DT	0.067	0.069	0.072	0.073	0.064	0.060

Tabla 3.9: Error medio absoluto de los residuos obtenidos a partir del ajuste lineal realizado entre la ganancia de exactitud y la medida de máxima correlación mostrada en la Tabla 3.8. Los errores más bajos para cada proporción de ejemplos etiquetados se han resaltado en negrita.

Para el resto de los métodos la correlación de la complejidad con las ganancias resulta generalmente negativa, pues a medida que la complejidad aumenta las ganancias positivas disminuyen. La Figura 3.10 muestra en detalle el comportamiento de las ganancias para el método de multiaprendizaje Democratic. Este método alcanza ganancias positivas apreciables para los conjuntos de datos de baja y mediana complejidad de la frontera. Cuando se utiliza la medida DTW las ganancias se concentran para valores extremadamente bajos de complejidad.

Otro de los métodos de auto-etiquetado que resulta de interés analizar es el Tri-training. La Figura 3.11 muestra el comportamiento de dicho método para cada clasificador base estudiado. En general, el comportamiento de este método es favorable para valores bajos de complejidad y se deteriora cuando esta comienza a ser elevada. Además, se destacan los altos valores de ganancia que se obtiene en la mayoría de los conjuntos de datos con Tri-training-SVM (euclidiana) para el umbral  $T1 < 0.20$ . En el caso del clasificador base DT, los valores de ganancia no resultan prometedores para ningún rango de los valores de complejidad.

Todos los gráficos de dispersión correspondientes a las correlaciones mostradas en la Tabla 3.8 se han incluido en la sección C.2. Para complementar estos

resultados se calculó además el error medio absoluto de los residuos obtenidos en cada ajuste lineal (Tabla 3.9). Los errores más bajos se obtienen generalmente para los métodos basados en el algoritmo 1NN. El método Democratic también presenta un ajuste aceptable. El resto de los métodos presentan errores más elevados causados por puntos de datos alejados considerablemente del ajuste lineal.

En general existe una correlación, en la mayoría de los casos significativa, entre las medidas de complejidad estudiadas y las ganancias de exactitud obtenidas con los métodos de auto-etiquetado. La medida T1 es seleccionada con mayor frecuencia al favorecer altos valores de correlación en comparación con el resto de las medidas. El método SETRED presenta una correlación positiva respecto a la complejidad estimada, lo cual sugiere su utilización en conjuntos de datos de mediana y alta complejidad. Por otra parte los métodos Democratic, Tri-training-1NN y Tri-training-SVM presentan una correlación negativa respecto a la complejidad, alcanzando las mayores ganancias de exactitud para conjuntos de datos caracterizados por una baja complejidad.

Resulta necesario destacar que las correlaciones observadas entre ganancia y complejidad no resultan tan fuertes como las obtenidas para los valores de exactitud, observándose una dispersión considerable entre las variables estudiadas. Este resultado confirma que son varios los factores que determinan las ganancias de exactitud al aplicar clasificación mediante métodos de auto-etiquetado y la complejidad de los conjuntos de datos es sólo uno de los factores a tener en cuenta en este tipo de análisis.

### 3.4 Sumario

Este capítulo estuvo dedicado a evaluar la dificultad, desde el punto de vista de la clasificación, que presentan los conjuntos de ejemplos etiquetados que se utilizan inicialmente durante el entrenamiento semi-supervisado. Con este fin se evaluaron cinco medidas de complejidad supervisadas, las cuales en su mayoría se basan en el cálculo de la disimilitud entre los ejemplos lo cual facilitó su aplicación en el dominio temporal.

Como parte del presente estudio se evaluaron las medidas de complejidad sobre 35 conjuntos de datos temporales. La evaluación abarcó tres proporciones

distintas de ejemplos etiquetados (10 %, 20 % y 30 %), siguiendo un esquema de cinco particiones donde se promediaron los resultados individuales de cada partición. Para estudiar la relación existente entre la complejidad inicial estimada y el desempeño obtenido por los métodos de auto-etiquetado, se aplicó la prueba no paramétrica de Spearman con el objetivo de obtener los coeficientes de correlación entre las variables estudiadas. Las medidas de desempeño utilizadas en el estudio fueron la exactitud de la clasificación y la ganancia obtenida al aplicar aprendizaje semi-supervisado.

Luego de analizar los resultados de la experimentación, podemos extraer las siguientes conclusiones:

- El uso de la medida DTW provoca una disminución significativa de la complejidad en comparación con la distancia euclidiana, atendiendo a las estimaciones obtenidas mediante cuatro de las medidas de complejidad estudiadas.
- Las medidas de complejidad involucradas en este estudio permiten realizar una evaluación acertada de la complejidad de los conjuntos de datos temporales, incluso para porciones pequeñas de ejemplos etiquetados.
- Los valores de complejidad estimados mostraron una fuerte correlación con la exactitud obtenida, mediante los métodos de auto-etiquetado, para todas las proporciones de ejemplos etiquetados utilizadas. Este resultado permitió identificar qué aspectos relacionados con la complejidad geométrica de la frontera entre clases tienen mayor influencia en la exactitud de la clasificación obtenida, los cuales son cuantificados por las medidas N1, N3 y T1.
- Las ganancias de exactitud obtenidas con los métodos de auto-etiquetado también se encuentran correlacionadas con la complejidad estimada de los conjuntos de datos temporales. Aunque los coeficientes de correlación obtenidos resultaron más débiles en comparación con la exactitud, el análisis realizado permitió identificar importantes aspectos sobre el comportamiento de las ganancias de exactitud frente a la complejidad de los conjuntos de datos. El método SETRED se mostró más competitivo al enfrentar problemas con una complejidad media o alta. Por otra parte, los métodos Democratic

y Tri-training (con los clasificadores base 1NN y SVM) presentaron mayor incidencia de ganancias positivas en conjuntos de datos caracterizados por una complejidad baja.

Los resultados obtenidos evidencian la aplicabilidad de las medidas estudiadas para estimar la dificultad, desde el punto de vista de la clasificación, que enfrentan los métodos de auto-etiquetado en el contexto temporal. Dichas medidas pueden ser valoradas como herramientas para asesorar la aplicación del paradigma semi-supervisado a problemas de clasificación.



## Capítulo 4

# CBD-GA: criterios de parada para el algoritmo 1NNP

En el presente capítulo se propone un método de SSC para series temporales basado en el Self-training y el 1NN como clasificador base. El método propuesto está enfocado en un caso especial de SSC donde se encuentran disponibles ejemplos etiquetados de una sola clase. En la sección 4.1 se formaliza este tipo de aprendizaje, el cual tiene amplia aplicación en la clasificación de series temporales. La sección 4.2 se centra en la propuesta de un método basado en técnicas de análisis gráfico a partir del cual se define una familia de criterios de parada. En la sección 4.3 se describe el diseño experimental utilizado para la evaluación de los criterios de parada propuestos y se ofrecen detalles de los conjuntos de datos utilizados durante la validación. La sección 4.4 está dedicada a la evaluación y discusión de los resultados experimentales así como la comparación con las propuestas existentes del estado del arte. Finalmente, el sumario de la sección 4.5 concluye el capítulo con los resultados más importantes derivados de la evaluación experimental realizada.

## 4.1 Aprendizaje PU y su aplicación al dominio temporal

El aprendizaje a través de ejemplos positivos y no etiquetados [46, 42, 94, *Positive Unlabeled learning* (PU)] fue formalizado por François Denis y un grupo de colaboradores alrededor del año 2000. Este tipo de aprendizaje se ha investigado como un modelo semi-supervisado para lidiar con aquellas situaciones donde no se encuentran disponibles ejemplos negativos para efectuar el entrenamiento. Esto puede ser causado por la presencia de una clase negativa extremadamente diversa donde no resulta factible obtener una representación adecuada en forma de ejemplos de entrenamiento. Este escenario se presenta con frecuencia en el dominio temporal, por ejemplo, en la clasificación de insectos voladores mediante sensores de sonido.

### 4.1.1 Formalización del aprendizaje PU

El aprendizaje de tipo PU puede considerarse un caso particular del SSL donde se tienen ejemplos etiquetados de una sola clase. También puede interpretarse como la extensión de la clasificación de una clase [146, *one-class classification*] al contexto semi-supervisado. Formalmente, podemos definir la clasificación en el marco del aprendizaje PU de la siguiente forma:

**Definición 4.1.** La clasificación de tipo PU es considerado un caso particular de la SSC, donde el conjunto de entrenamiento está formado por  $l$  ejemplos etiquetados de una única clase (denominada clase positiva)  $P = \{x^i\}_{i=1}^l$  y  $u$  ejemplos sin etiquetar  $U = \{x^j\}_{j=l+1}^{l+u}$ . El conjunto  $U$  contiene ejemplos de la clase positiva además de ejemplos que no pertenecen a dicha clase (denominada clase negativa). El objetivo del aprendizaje es entrenar un clasificador que permita discriminar, entre la clase positiva y negativa, los ejemplos en  $U$ .

Desde su surgimiento el aprendizaje PU ha encontrado múltiples aplicaciones en diversos dominios. En la clasificación de textos electrónicos ha sido ampliamente utilizado [48, 47, 98], pues resulta factible definir como positivos aquellos sitios web que resulten de interés a un determinado usuario. Por el con-

trario, definir la clase negativa no resulta igual de factible dada su diversidad. Además, en este dominio resulta común contar con abundantes ejemplos no etiquetados. Otra aplicación del aprendizaje PU ha tenido lugar en el campo de la genética, específicamente para identificar genes humanos asociados con determinadas enfermedades. En este caso se toman como ejemplos positivos, aquellos genes que se saben asociados a la enfermedad. En el dominio temporal existen problemas que son factibles de modelar utilizando este tipo de aprendizaje, por ejemplo la clasificación de insectos voladores [33]. Los sonidos que producen los insectos durante el vuelo son captados por sensores [10] y estas son las series que se utilizan durante el entrenamiento. Los ejemplos positivos son aquellas series que se corresponden con la especie de interés mientras que los ejemplos no etiquetados incluyen además toda clase de sonidos ambientales producidos por el hombre o por otros tipos de insectos.

La ausencia de ejemplos negativos para efectuar el aprendizaje impide que la clasificación de tipo PU pueda ser enfrentada directamente mediante técnicas clásicas del aprendizaje supervisado o semi-supervisado. Dos enfoques principales se han desarrollado en la literatura para lidiar con esta situación.

El primero se basa en la adaptación de técnicas tradicionales del aprendizaje supervisado a la clasificación PU [27, 97, 101, 168]. En el trabajo realizado por Zhang y Zuo [168] se emplea el  $k$ NN estándar para la obtención de un ordenamiento de los ejemplos no etiquetados, de acuerdo a su similitud con respecto a los ejemplos de la clase positiva. Luego, los  $r$  primeros ejemplos son etiquetados como positivos. Desafortunadamente, no se propone un método que aproxime el valor de  $r$ .

El segundo enfoque se concentra en adaptar las técnicas estándar de SSL al contexto PU [47, 167]. El trabajo propuesto por Denis y colaboradores en el año 2003 [47] está basado en el método Co-training de conjunto con el clasificador Naive Bayes como esquema de aprendizaje, mientras que la propuesta de Yu y Li [167] utiliza una técnica de SSL basada en grafos.

### 4.1.2 Aprendizaje PU aplicado a la tarea de clasificación de series temporales

El enfoque PU se ha utilizado ampliamente en la tarea de clasificación de series temporales. Se han utilizado fundamentalmente dos enfoques con este objetivo. El primero se basa en la utilización de técnicas para obtención de conglomerados [118, 119]. El segundo, basado en las técnicas de aprendizaje supervisado tradicionales, es el enfoque más utilizado [157, 136, 33, 11]. En su mayoría, estas soluciones realizan una adaptación al contexto PU del método Self-training estándar. La elección del esquema de aprendizaje base recae comúnmente en el método  $k$ NN dada su probada efectividad en la clasificación de series temporales.

La combinación del Self-training con el 1NN como clasificador base se conoce con el nombre de propagación del 1NN [173, *Propagating 1-Nearest Neighbor* (P1NN)] en el contexto de la SSC estándar. La idea principal de este algoritmo se basa en propagar la clase de los ejemplos etiquetados a los ejemplos más cercanos que no están etiquetados, según se muestra en el Algoritmo 4.1. En cada iteración se selecciona el ejemplo no etiquetado  $x$  más cercano a algún ejemplo  $x'$  contenido en  $L$ . La confianza de las predicciones se aproxima a través de la función de distancia  $d$  utilizada. La cantidad de comparaciones que realiza este algoritmo, considerando constante el número de elementos en  $L$ , estaría acotada superiormente por un  $O(u^3)$ , siendo  $u$  la cantidad inicial de elementos no etiquetados.

El criterio de parada que se sigue (línea 2) es continuar el proceso iterativo hasta que se agote totalmente el conjunto  $U$ . Considerando que este algoritmo está diseñado para ser utilizado en la SSC estándar, no puede ser aplicado directamente a la clasificación PU. Si aplicamos este algoritmo al problema PU todos los ejemplos en  $U$  terminarían siendo clasificados como positivos, lo cual no es cierto. La forma en que las propuestas presentadas en la literatura [157, 136] superaron esta dificultad originó una modificación de este algoritmo donde se le asigna un papel más activo al criterio de parada.

**Algoritmo 4.1:** Propagación del 1NN

```

1 Función P1NN( $L, U, d$ )
   Entrada: ejemplos etiquetados  $L$ , ejemplos no etiquetados  $U$ ,
             función de distancia  $d$ 
   Salida: conjunto extendido  $EL$ 
2 mientras  $U \neq \emptyset$  hacer
3    $(x, x') \leftarrow \operatorname{argmin}_{\{x \in U, x' \in L\}} d(x, x')$ 
4    $f(x) \leftarrow y'$  etiqueta del ejemplo  $x'$ 
5    $U \leftarrow U \setminus x$ 
6    $L \leftarrow L \cup (x, f(x))$ 
7 fin
8 retornar  $L$ 
9 fin

```

**Algoritmo P1NN adaptado al contexto PU**

La adaptación del algoritmo P1NN al contexto PU es introducida originalmente por Wei y Keogh [157] en el año 2006 para tratar el problema de la clasificación de series temporales. Posteriormente, es retomado por Ratanamahatana y Wanichsan [136] en el año 2008 con igual fin. El Algoritmo 4.2, que en lo adelante será denotado como P1NNPU<sup>1</sup>, muestra la adaptación resultante.

Como resultado de la ejecución iterativa del 1NN, se obtiene un ordenamiento de los ejemplos no etiquetados en la tupla *RankInst* y una tupla de distancias mínimas *Mindist*. El  $i$ -ésimo valor en *RankInst*, denotado como *RankInst*( $i$ ), hace referencia al ejemplo no etiquetado seleccionado en la iteración  $i$  (línea 8). De igual forma, el  $i$ -ésimo valor en *Mindist*, denotado como *Mindist*( $i$ ), hace referencia a la distancia entre el ejemplo no etiquetado seleccionado durante la iteración  $i$  y su ejemplo más cercano en  $P$  (línea 9).

Dada su naturaleza iterativa, una parte crítica del Self-training consiste en determinar cuándo debe detenerse el proceso de aprendizaje. El objetivo de este

<sup>1</sup>Las ligeras modificaciones introducidas respecto al algoritmo original propuesto por Ratanamahatana y Wanichsan sólo han tenido como objetivo facilitar una mejor comprensión del mismo.

**Algoritmo 4.2:** P1NN adaptado a la clasificación PU

```

1 Función P1NNPU( $P, U, d$ )
   Entrada: ejemplos positivos  $P$ , ejemplos no etiquetados  $U$ , función
           de distancia  $d$ 
   Salida: conjunto extendido de ejemplos positivos  $EP$ 
2    $RankInst \leftarrow \emptyset$ 
3    $i \leftarrow 1$ 
4    $Mindist \leftarrow \emptyset$ 
5    $P' \leftarrow P$ 
6   mientras ( $U \neq \emptyset$ ) hacer
7      $(x, x') \leftarrow \operatorname{argmin}_{\{x \in U, x' \in P\}} d(x, x')$ 
8      $RankInst(i) \leftarrow x$ 
9      $Mindist(i) \leftarrow d(x, x')$ 
10     $U \leftarrow U \setminus x$ 
11     $P \leftarrow P \cup x$ 
12     $i \leftarrow i + 1$ 
13  fin
14   $sp \leftarrow \mathbf{stoppingCriterion}(Mindist)$ 
15   $P \leftarrow P' \cup \{RankInst(1), \dots, RankInst(sp)\}$ 
16  retornar  $P$ 
17 fin

```

mecanismo en la SSC estándar es evitar la inclusión en  $L$  de ejemplos etiquetados con bajo nivel de confianza. En el caso de la clasificación PU, el criterio de parada se hace aún más importante pues la existencia de ejemplos etiquetados pertenecientes a una única clase facilita, en la medida que avanza el proceso iterativo, la inclusión errónea de ejemplos negativos en  $P$ . Por este motivo, el criterio de parada (**stoppingCriterion**) es el encargado de calcular la confianza de los nuevos ejemplos positivos que se han añadido, tomando como base la información que brinda la tupla  $Mindist$  (línea 14). El punto de parada indicado por la función **stoppingCriterion** es utilizado para construir el conjunto resultante  $P$  que será retornado (líneas 15–16). El punto de parada que idealmente debe devolver un criterio de parada es aquel que permita aprender suficientemente la clase positiva,

sin incluir en  $P$  ejemplos de la clase negativa.

### 4.1.3 Criterios de parada existentes para el Self-training

Múltiples criterios de parada, orientados a la clasificación de series temporales, han sido propuestos en el marco del algoritmo Self-training. A continuación se analizan cada uno de ellos.

#### Criterio WK

El criterio de parada propuesto por Wei y Keogh [157] (en lo adelante será denotado como WK) propone una heurística sencilla basada en las estadísticas obtenidas durante la ejecución del algoritmo P1NNPU. La función de distancia que utiliza el 1NN en esta propuesta es la euclidiana. Para determinar el punto de parada, el criterio WK utiliza una función que almacena en cada iteración  $i$  la distancia mínima entre todas las almacenadas en  $Mindist$  desde la primera iteración hasta la iteración  $i$ -ésima (Expresión 4.1).

$$Minsofar(i) = \min(Mindist(1, \dots, i)) \quad (4.1)$$

Usualmente, los valores de  $Minsofar(i)$  decrecen durante las primeras iteraciones, luego se estabilizan durante un número considerable de iteraciones y finalmente experimentan un nuevo descenso. Este último comportamiento se asocia con la introducción de ejemplos negativos en  $P$ . La heurística definida por el criterio WK se basa esencialmente en supervisar los cambios significativos en  $Minsofar(i)$  para detener el aprendizaje antes de comenzar la adición de ejemplos negativos en  $P$ . Con frecuencia, el criterio de parada WK causa un aprendizaje incompleto de la clase positiva conocido en la literatura como detención temprana (*early stopping*).

#### Criterio RW

La mejora propuesta por Ratanamahatana y Wanichsan [136] (en lo adelante será denotado como RW) consiste esencialmente en un nuevo criterio de

parada basado también en los cambios que experimenta la distancia mínima almacenada en cada iteración. Específicamente, el criterio RW extrae información de los cambios que ocurren en cada par de valores consecutivos almacenados en la tupla  $Mindist$ . Las Expresiones 4.2 y 4.3 especifican la forma en que se evalúa el criterio de parada RW (SCRW) en cada iteración  $i, i = 1, \dots, u$ . Las diferencias obtenidas entre las iteraciones pueden ser positivas o negativas, pero lo más importante de estas es su valor absoluto. El ratio obtenido se multiplica por un factor de peso  $LW(i)$ , siendo  $u$  el tamaño del conjunto inicial  $U$ . Este peso garantiza la adecuada convergencia del proceso de aprendizaje ya que los mayores valores son asignados a los ejemplos que se añaden en  $P$  durante las primeras iteraciones mientras que a los finales se les asignan los valores más bajos. Una vez evaluada la Expresión 4.2 en todas las iteraciones posibles, se define como punto de parada  $p - 2$ , siendo  $p$  la iteración donde el criterio  $SCRW$  alcanza el valor máximo.

$$SCRW(i) = \frac{|Mindist(i) - Mindist(i - 1)|}{Std(Mindist(1, \dots, i))} \cdot LW(i) \quad (4.2)$$

$$LW(i) = \frac{u - (i - 1)}{u} \quad (4.3)$$

Aunque los resultados obtenidos con el criterio RW superan los del anterior criterio WK, no se soluciona el problema de la detención temprana. Esto se debe a la existencia de situaciones donde dos valores consecutivos de  $Mindist$  no resultan suficientes para reconocer adecuadamente la frontera entre clases.

### Criterio BHRK

Una contribución reciente al problema del criterio de parada [11, 147] intenta aprender la estructura intrínseca de los ejemplos usando el concepto de longitud de descripción mínima [69, *Minimum Description Length* (MDL)]. Este concepto, ampliamente utilizado en otros dominios, es empleado por Begum y colaboradores [11] (en lo adelante será denotado como BHRK) para desarrollar un nuevo criterio de parada en el dominio temporal.

La longitud de descripción de una serie de tiempo  $s$  es calculada como  $DL(s) = w \cdot \log_2 a$ , donde  $w$  es la longitud de  $s$  y  $a$  es la cardinalidad (se sugiere



un valor de 16). La longitud de descripción de  $s$  puede ser reducida mediante la explotación de la información presente en la hipótesis  $h$  según se expresa en la Expresión 4.4. Aquí  $h$  representa una subsecuencia de la serie que se utiliza para codificar una o más subsecuencias. La descripción de  $s$ , teniendo en cuenta  $h$ , se calcula según la Expresión 4.5. Si la hipótesis  $h$  resulta efectiva entonces se cumple que  $DL(s, h) < DL(s)$ , alcanzándose de esta forma una reducción de la longitud de descripción.

$$DL(s, h) = DL(h) + DL(s|h) \quad (4.4)$$

$$DL(s|h) = DL(s - h) \quad (4.5)$$

El criterio de parada BHRK intenta comprimir un conjunto de entrenamiento no etiquetado  $U$ , utilizando como hipótesis  $h$  un ejemplo positivo inicial. Mientras la codificación de los ejemplos no etiquetados usando  $h$  logre una reducción de la longitud de descripción de  $U$ , estos ejemplos son considerados positivos. El punto de parada se alcanza cuando la longitud de descripción deja de disminuir. En este punto se considera que la hipótesis  $h$  es incapaz de comprimir ningún otro ejemplo y se detiene el aprendizaje de la clase positiva.

Para poder hacer uso del criterio BHRK es necesario convertir los valores reales de la serie en valores discretos. Además, este criterio no resulta efectivo en situaciones donde las series presenten distorsiones en el eje del tiempo. El principal campo de aplicación de este método es en el dominio de la Medicina, por ejemplo aplicado a electrocardiogramas, donde los ejemplos (cada latido del corazón puede considerarse un ejemplo) son suficientemente similares entre sí de forma que sea factible la compresión de estos.

Para ilustrar el funcionamiento de los criterios analizados estos serán evaluados en el conjunto de datos ECG [156]. La Figura 4.1 muestra los valores de la tupla *Mindist* calculados en cada iteración mediante el algoritmo P1NNPU. El conjunto  $P$  se inicializa con un único ejemplo positivo mientras que el conjunto  $U$  se inicializa con 809 ejemplos sin etiquetar. Es posible observar que la primera etapa del algoritmo, donde se ordenan los ejemplos no etiquetados, se ejecutó de manera satisfactoria ya que la mayoría de los ejemplos positivos son seleccionados de manera consecutiva durante las primeras 167 iteraciones. Sin embargo, cuando se evalúan los criterios de parada analizados, el criterio RW propone el punto

de parada en la iteración 19, el criterio WK en la iteración 3 y el BHRK en la iteración 2. Este resultado se considera como detención temprana pues el punto de parada correcto está en la iteración 167. Este resultado ilustra la necesidad de proponer criterios de parada que logren extraer una mayor información de la tupla *Mindist* con el propósito de distinguir de manera más acertada la frontera entre clases.

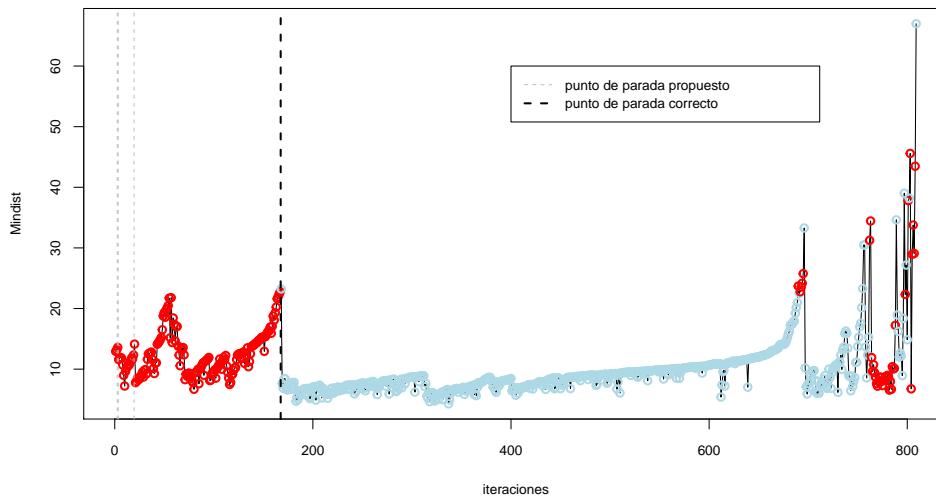


Figura 4.1: Distancias mínimas obtenidas en cada iteración mediante el algoritmo P1NNPU. El color de cada punto está dado por la clase a la que pertenece el ejemplo adicionado en dicha iteración. La clase positiva se ha representado en rojo y la negativa en azul.

## 4.2 Criterios de parada CBD-GA para el algoritmo P1NNPU

En esta sección se presenta un nuevo método para identificar la frontera entre clases que hace uso del análisis gráfico. El nuevo método es nombrado como CBD-GA por sus siglas en inglés (*Class Boundary Description using Graphical Analysis*). Algunos conceptos de las técnicas de análisis gráfico son aplicados pa-

ra identificar la frontera entre clases en la tupla *Mindist* mediante tres curvas principales. A partir de los parámetros, estimados en estas tres curvas, se define toda una familia de criterios de parada heurísticos. Además, se adaptan estadísticos clásicos del aprendizaje automático para evaluar de manera más acertada el comportamiento de los criterios de parada.

### 4.2.1 Descripción de la frontera entre clases mediante tres curvas

Como se abordó en la sección anterior, considerar sólo dos valores consecutivos de la tupla *Mindist* (criterio RW) resulta en ocasiones insuficiente para identificar correctamente el fin del aprendizaje de la clase positiva. Considerar en el análisis una mayor cantidad de puntos puede facilitar apreciablemente esta tarea. Una inspección visual a la Figura 4.1 sugiere que no solamente el número de puntos resulta informativo sino que la tendencia que siguen estos también se relaciona con la localización del punto de parada correcto, pues se experimentan varios cambios de tendencia alrededor de este.

La tupla *Mindist* puede considerarse como una serie de tiempo de distancias mínimas obtenidas durante una secuencia de iteraciones. El problema del criterio de parada puede ser formulado como la tarea de identificar ciertos patrones en esta serie temporal. El objetivo principal del análisis gráfico es la identificación de patrones y tendencias en datos secuenciales. El análisis gráfico, el cual se considera parte de la metodología del análisis técnico [88], define tres tipos básicos de tendencias: ascendente, descendente y horizontal. Aunque su principal aplicación ha estado orientada al estudio de precios en mercados de libre comercio [50], el método propuesto aplica una idea similar en la identificación de la frontera entre clases mediante el análisis de los valores secuenciales en *Mindist*.

Para lograr este objetivo se definen las tres curvas que se encuentran vinculadas estrechamente con la región frontera. Para cada curva se introduce una definición general y la interpretación de su significado para el proceso de aprendizaje. Las curvas se definen como:

- Ascendente: Una curva ascendente  $ac_{[b,p]}$ , la cual comienza en la iteración  $b$  y termina en la iteración  $p$ , debe satisfacer la restricción planteada a

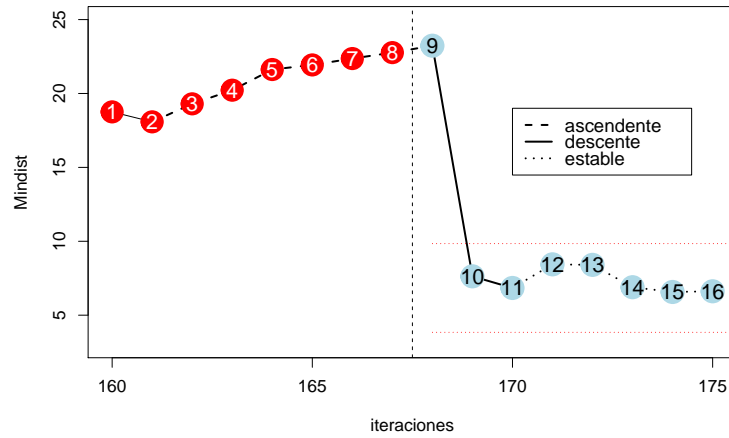


Figura 4.2: La frontera entre clases (línea discontinua vertical) rodeada de una vecindad de ocho ejemplos a cada lado. Las tres curvas principales están representadas mediante tres tipos diferentes de líneas.

continuación, la cual garantiza que la curva sea monótona ascendente.

$$\forall i, i = b, \dots, p - 1, Mindist(i) < Mindist(i + 1)$$

Un comportamiento ascendente de los valores en  $Mindist$  significa que el aprendizaje se está moviendo a través de una región poco densa del espacio formado por los ejemplos. Resulta común relacionar la terminación del aprendizaje en el espacio positivo con regiones donde los ejemplos se encuentran alejados entre sí. Este comportamiento eleva las distancias mínimas y sugiere una cercanía con la frontera de las iteraciones desde  $b$  hasta  $p$ . La Figura 4.2 muestra una imagen ampliada de la región frontera presentada en la Figura 4.1. El intervalo que contienen la frontera entre clases (entre las iteraciones 161 y 168) forma una curva ascendente  $ac_{[161,168]}$ .

De este tipo de curva resulta de interés su altura  $ha_{[b,p]}$  la cual es calculada como  $ha_{[b,p]} = Mindist(p) - Mindist(b)$ . La altura de la curva ascendente mide el crecimiento experimentado por la distancia entre las iteraciones  $b$  y  $p$ .

- Descendente: Una curva descendente  $dc_{[p,v]}$ , la cual comienza en la iteración  $p$  y termina en la iteración  $v$ , debe satisfacer la restricción planteada a

continuación, la cual garantiza que la curva sea monótona decreciente.

$$\forall i, i = p, \dots, v - 1, Mindist(i) > Mindist(i + 1)$$

Un comportamiento descendente significa que el proceso de aprendizaje se está moviendo desde una región de baja densidad hacia una región más densa, donde los ejemplos se encuentran más cercanos entre sí. Este comportamiento resulta común justo después de atravesar la frontera entre las clases. La Figura 4.2 muestra una curva descendente  $dc_{[168,170]}$  después de pasar la frontera.

Una vez más, resulta necesario calcular la altura de ese tipo de curva como  $hd_{[p,v]} = Mindist(p) - Mindist(v)$ . Adicionalmente, se propone calcular una versión de la altura normalizada en el rango  $[0, 1]$  mediante la expresión  $hdn_{[p,v]} = hd_{[p,v]} / \max(Mindist(1, \dots, u))$ .

- Estable: Una curva estable  $sc_{[v,e]}$ , la cual comienza en la iteración  $v$  y termina en la iteración  $e$ , debe satisfacer la restricción planteada a continuación. Esta restricción garantiza que los valores de dicha curva se mantengan en un rango acotado de valores, los cuales se definen mediante dos líneas horizontales.

$$\forall i, i = v, \dots, e, Mindist(v) + \beta \cdot hd_{[p,v]} > Mindist(i) > Mindist(v) - \beta \cdot hd_{[p,v]}$$

Una curva estable provee información relevante para determinar el criterio de parada y debe ser precedida por una curva descendente  $dc_{[p,v]}$ , pues su definición depende de esta. La curva estable describe un número de iteraciones consecutivas donde los valores de  $Mindist$  muestra una tendencia horizontal respecto a la altura de la curva descendente  $hd_{[p,v]}$ . El valor de  $\beta \in \mathbb{R}$  controla la estabilidad de la curva y está restringido al intervalo  $(0, 1]$ . Un valor cercano a cero se considera muy restrictivo mientras que un valor próximo a uno relajaría demasiado los límites de la curva estable. En experimentos preliminares, basados en una muestra representativa de conjuntos de datos, se ha fijado de forma empírica este parámetro a  $\beta = 0.3$ . Aunque los valores de  $Mindist$  tienen un comportamiento oscilante, sus variaciones son limitadas ya que los ejemplos se encuentran cercanos entre sí. La curva estable representa una región densa del espacio de ejemplos. Este comportamiento se observa usualmente cuando se

comienza el aprendizaje de una nueva clase. La Figura 4.2 muestra una curva estable  $sc_{[170,175]}$  delimitada por dos líneas horizontales discontinuas.

De este tipo de curva nos interesa su duración, la cual se calcula como  $ws_{[v,e]} = e - v$ . Para este tipo de curva se propone además una versión normalizada en el rango  $[0, 1]$  mediante la expresión  $wsn_{[v,e]} = ws_{[v,e]}/(u - 1)$ , donde  $u$  representa el número inicial de ejemplos no etiquetados.

Los ejemplos que son seleccionados durante una curva estable se encuentran más cercanos a los ejemplos adicionados después de la iteración  $p$  que a los ejemplos añadidos antes de dicha iteración. Según las definiciones realizadas,  $p$  representa la iteración inicial de la curva descendente  $dc_{[p,v]}$  que precede a la curva estable. Esta afirmación es demostrada en el Apéndice D.

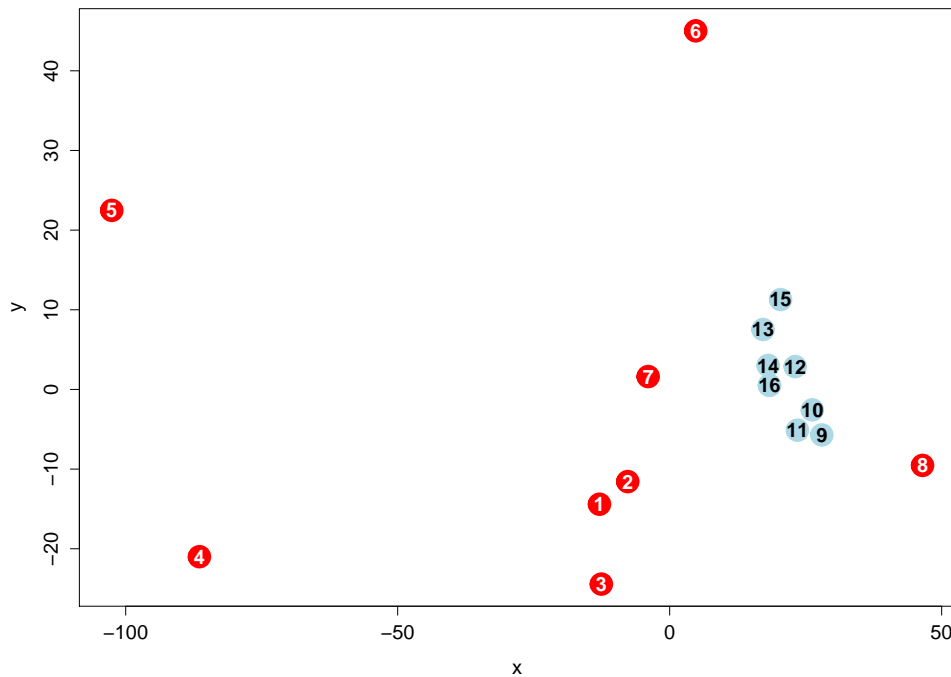


Figura 4.3: Técnica clásica de escalado multidimensional (*Classical Multidimensional Scaling*) que representa el espacio de ejemplos alrededor de la frontera entre clases. Los últimos ejemplos positivos seleccionados se encuentran diseminados alrededor de la frontera mientras que los primeros ejemplos negativos se encuentran concentrados en una región de alta densidad.

La Figura 4.3 confirma, mediante un diagrama visual de dos dimensiones, la relación existente entre las distintas fases por las cuales transcurre el aprendizaje alrededor de la frontera entre clases y las curvas definidas. En esta figura se preservaron las mismas etiquetas utilizadas en los ejemplos de aprendizaje mostrados en la Figura 4.2. Aquí se muestra cómo los últimos ejemplos que son seleccionados de la clase positiva se encuentran diseminados y sus valores de *Mindist* correspondientes forman una curva ascendente. Cuando los primeros ejemplos negativos son seleccionados, los valores en *Mindist* decrecen bruscamente formando una curva descendente. El resto de los ejemplos negativos seleccionados mantiene bajos valores de *Mindist* y como consecuencia forman una curva estable, pues el aprendizaje se ha movido a una región más densa.

En resumen, se han definido tres curvas en *Mindist* que caracterizan las diferentes fases por las que transcurre el aprendizaje alrededor del punto de parada correcto. A partir de estas curvas será definida una nueva familia de criterios de parada.

### 4.2.2 Criterios de parada basados en CBD-GA

Tal y como se mostró en el algoritmo P1NNPU, la tupla *Mindist* es el único parámetro de entrada que utiliza el criterio de parada. Como un primer paso, durante la estimación del punto de parada, los valores de *Mindist* son separados en intervalos de la forma  $I_{[b,e]}$ . En esta notación,  $b$  representa la iteración que da inicio al intervalo y  $e$  la que lo finaliza. Además, cada intervalo  $I_{[b,e]}$  está formado por las tres curvas definidas y estas deben aparecer en el siguiente orden:  $ac_{[b,p]}$ ,  $dc_{[p,v]}$  y  $sc_{[v,e]}$ . El resto de los valores en *Mindist* que no resulta posible representarlos de esta forma son descartados por el algoritmo para el cálculo del criterio de parada.

En un intervalo  $I_{[b,e]}$ , se denota como  $I_p$  el índice del valor máximo en  $Mindist(b, \dots, e)$ , así  $Mindist(p) = \max(Mindist(b, \dots, e))$ . La evaluación de un criterio de parada sobre un intervalo  $I_{[b,e]}$  retorna como resultado en qué medida la iteración  $p$  marca el aprendizaje de una nueva clase. Además, se denota como  $I_{[b,e]}^i$  al  $i$ -ésimo intervalo en *Mindist*. Para cada intervalo candidato  $I_{[b,e]}^i$  se evalúa un criterio de parada denotado como *SC*. Los detalles del procedimiento

iterativo que localiza el punto de parada se muestran en el Algoritmo 4.3. El objetivo de este algoritmo es encontrar el intervalo  $I^c$  donde el criterio de parada  $SC$  obtiene el máximo valor,  $SC(I^c) = \max(SC(I^i))$  para  $i = 1 \dots C$  (líneas 6–12). El valor  $C$  representa el número total de intervalos candidatos que se obtuvieron de  $Mindist$ . Finalmente, se propone como punto de parada el índice  $I_{p-1}^c$  del intervalo seleccionado (líneas 13–15).

**Algoritmo 4.3:** Criterio de parada basado en el método CBD-DA

```

1 Función StoppingCriterion(Mindist)
   Entrada: distancias mínimas calculadas Mindist
   Salida: punto de parada propuesto sp
2 IntervalsList  $\leftarrow$  EncontrarIntervalo(Mindist)
3 c  $\leftarrow$  0
4 valueSelected  $\leftarrow$   $-\infty$ 
5 C  $\leftarrow$  longitud(IntervalsList)
6 para i = 1 hasta C hacer
7     candidateValue  $\leftarrow$  SC(IntervalsList(i))
8     si candidateValue > valueSelected entonces
9         c  $\leftarrow$  i
10        valueSelected  $\leftarrow$  candidateValue
11    fin
12 fin
13 I  $\leftarrow$  IntervalsList(c)
14 sp  $\leftarrow$   $I_p - 1$ 
15 retornar sp
16 fin

```

A continuación se define una familia de criterios de parada basados en las tres curvas propuestas en la sección 4.2.1. A partir de los nuevos criterios definidos se evaluará la utilidad de dichas curvas para la tarea de identificar el punto de parada correcto.

Los criterios de parada definidos en las Expresiones 4.6, 4.7 y 4.8 evalúan de manera separada el comportamiento de cada curva. Todos los criterios son multiplicados por el factor de peso  $LW(I_p)$ , definido anteriormente en la Expre-



sión 4.3, de manera que se garantice una convergencia adecuada del punto de parada.

$$SC1(I_{[b,e]}^i) = hd_{[p,v]}^i \cdot LW(I_p^i) \quad (4.6)$$

$$SC2(I_{[b,e]}^i) = ha_{[b,p]}^i \cdot LW(I_p^i) \quad (4.7)$$

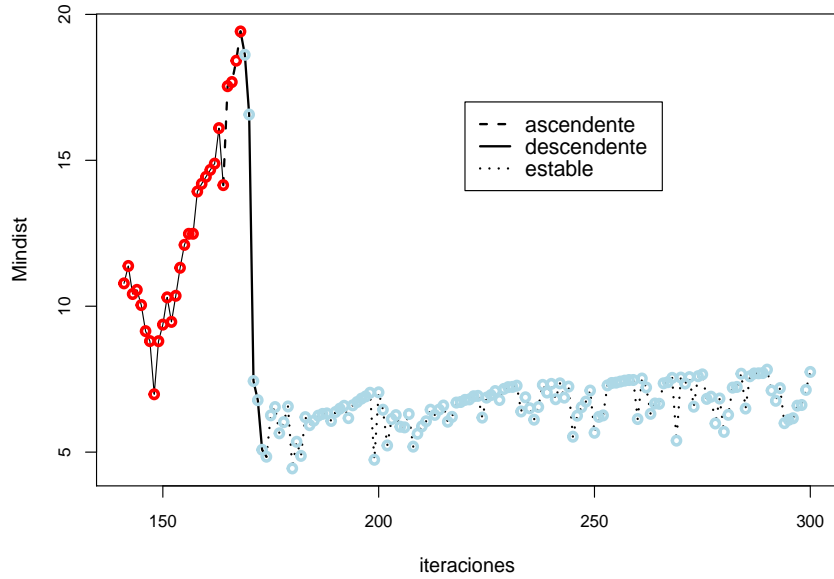
$$SC3(I_{[b,e]}^i) = ws_{[v,e]}^i \cdot LW(I_p^i) \quad (4.8)$$

$$SC4(I_{[b,e]}^i) = \max(ha_{[b,p]}^i, hd_{[p,v]}^i) \cdot LW(I_p^i) \quad (4.9)$$

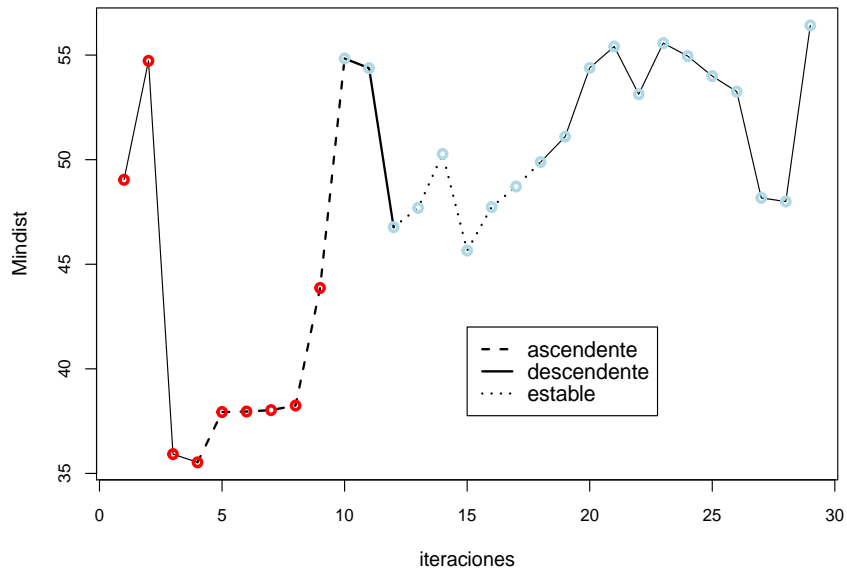
$$SC5(I_{[b,e]}^i) = \max(hdn_{[p,v]}^i, wsn_{[v,e]}^i) \cdot LW(I_p^i) \quad (4.10)$$

La forma específica en que se comportan las curvas depende de las características propias del conjunto de entrenamiento. La curva ascendente resulta más apropiada para identificar el punto de parada en aquellos problemas donde los ejemplos positivos se encuentren más cercanos entre sí que los ejemplos negativos (Figura 4.4 a). La curva descendente es más apropiada en aquellos problemas donde ocurre precisamente lo contrario (Figura 4.4 b). Por otra parte, la curva estable resulta más útil en aquellos problemas con un gran número de instancias negativas lo suficientemente cercanas entre sí (Figura 4.4 b).

La Expresión 4.9 define un criterio de parada combinando las alturas de la curva ascendente y la descendente, con el objetivo de aumentar su efectividad frente a problemas con características diversas. La Expresión 4.10 sigue una idea similar al combinar la altura de la curva descendente con la duración de la curva estable. Para todos los criterios de parada definidos, la complejidad temporal del Algoritmo 4.3 se mantiene siendo un  $O(u)$ , siendo  $u$  la longitud de la tupla *Mindist*.



(a) Conjunto de datos ECG.



(b) Conjunto de datos CBF.

Figura 4.4: Diversidad de comportamiento de las tres curvas definidas.

### 4.2.3 Medidas estadísticas para la evaluación del criterio de parada

La evaluación está enfocada en comprobar las habilidades transductivas de los criterios de parada propuestos para el método CBD-GA. Por tanto, es necesario evaluar los ejemplos positivos en  $U$  que fueron correctamente identificados por los criterios de parada. Para lograr este objetivo se utiliza un conjunto de entrenamiento donde se conoce de antemano el subconjunto real de ejemplos positivos ( $RP$ ) y el de los negativos ( $RN$ ).

El estadístico F-Measure (Expresión 4.11) se selecciona para evaluar los resultados del proceso de auto-etiquetado debido a que este toma en cuenta tanto la precisión (*precision*) como la sensibilidad (*recall*) de la clasificación [159]. Los valores de la matriz de confusión, a partir del conjunto  $RP$  y del conjunto final positivo  $EP$ , son calculados según se muestra en la Tabla 4.1. El conjunto  $RU$  contiene los ejemplos no etiquetados en  $U$  que no fueron seleccionados para formar parte de  $EP$ .

		Clase predicha	
		Positiva	Negativa
Clase real	Positiva	$TP =  EP \cap RP $	$FN =  RP \setminus EP $
	Negativa	$FP =  EP \setminus RP $	$TN =  RU \cap RN $

Tabla 4.1: Matriz de confusión que describe el aprendizaje PU.

$$F\text{-Measure} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (4.11)$$

Aunque el estadístico F-Measure brinda información general sobre el comportamiento del proceso de auto-etiquetado durante la fase de entrenamiento, se hace necesario evaluar la precisión con que los criterios establecen el punto de parada. Con este fin denotamos como punto ideal  $pi$  al índice de la tupla  $RankInst$  donde termina la primera secuencia consecutiva de ejemplos positivos y como punto de parada  $pp$  al índice retornado por el criterio a evaluar. Considerando esta notación, el índice  $pi + 1$  hace referencia al primer ejemplo negativo incluido en  $RankInst$  y esto hace posible cuantificar el número de ejemplos incorrectamente incluidos después de  $pi$ . A partir de estos valores se define como  $TP$  al valor

$\min\{pp, pi\}$  y como  $FP$  el valor  $\max\{0, pp - pi\}$ . La Expresión 4.12 muestra la medida estándar de precisión que se calcula en base a los valores definidos.

$$precision = \frac{TP}{TP + FP} \quad (4.12)$$

En resumen, para evaluar las habilidades transductivas de los criterios de parada se adaptan dos estadísticos clásicos del aprendizaje automático. El F-Measure se utiliza para evaluar el comportamiento general del aprendizaje PU y la precisión evalúa los ejemplos negativos incluidos erróneamente por el criterio de parada.

### 4.3 Marco experimental

En esta sección se brinda una breve descripción de los conjuntos de datos temporales incluidos en el estudio así como el diseño experimental utilizado para evaluar los criterios de parada propuestos. Según se describió en la sección 1.2.5, el adecuado desempeño del método Self-training depende en gran medida del cumplimiento de su principal supuesto, el cual se cumple cuando las clases forman conglomerados bien definidos. La presencia de ejemplos que se encuentran irregularmente alejados del resto (*outliers*) pueden afectar sensiblemente el comportamiento del Self-training. Con el objetivo de evaluar la proximidad de los ejemplos positivos, con respecto a la clase opuesta, seleccionamos el estimado *Leave-One-Out Error* (LOOE)<sup>2</sup> del clasificador 1NN. Como garantía para el cumplimiento del supuesto que asume el Self-training, se ha fijado un umbral (LOOE < 0.10) que asegure un adecuado nivel de separabilidad entre clases y además un número suficiente de conjuntos de datos que permitan realizar análisis estadísticos confiables.

Como resultado de la selección realizada, a partir del umbral fijado, se obtuvieron 15 conjuntos de datos provenientes de repositorios públicos [34, 156]. La

---

<sup>2</sup>Este estadístico se utilizó como medida de complejidad N3 en el estudio del Capítulo 3 donde obtuvo los valores más altos de correlación para el método Self-training con el 1NN como clasificador base.

Tabla 4.2 muestra los detalles de cada conjunto de datos temporales, incluyendo: cantidad de ejemplos en el conjunto de entrenamiento (Tamaño del conjunto), longitud de la series (Longitud), cantidad original de clases (Cantidad de clases), número de ejemplos positivos (Cantidad positivos) y el estimado LOOE evaluado para el 1NN con la distancia euclidiana. Considerando que el interés principal se centra en la evaluación de las capacidades transductivas del método propuesto, sólo se utilizarán las particiones de entrenamiento que provee el repositorio.

Conjunto de datos	Tamaño del conjunto	Longitud	Cantidad de clases	Cantidad positivos	LOOE
ECG	810	85	2	208	0.01
FaceAll	560	131	14	40	0.00
FacesUCR	200	131	14	13	0.07
Gun	50	150	2	24	0.00
Italy	67	24	2	34	0.02
Lighting7	70	319	7	12	0.08
MALLAT	55	1024	8	11	0.00
Olive	30	570	4	13	0.07
Sony	20	70	2	14	0.00
SonyII	27	65	2	11	0.09
Trace	100	275	4	26	0.03
Two	1000	128	4	271	0.00
Wafer	1000	152	2	903	0.00
Word	450	270	50	52	0.07
WordsS	267	270	25	60	0.05

Tabla 4.2: Descripción detallada de los conjuntos de datos temporales.

Para evaluar la efectividad de los criterios de parada propuestos en el contexto del aprendizaje PU, se requieren problemas de clasificación binarios pero muchos de los conjuntos de datos seleccionados tienen más de dos clases. Para este tipo de conjunto de datos se ha seleccionado, como la clase positiva, la primera clase con al menos 10 ejemplos y el resto de las clases es considerado como la clase negativa. Durante el entrenamiento se inicializa el conjunto  $P$  con una sola instancia positiva y el resto de los ejemplos de entrenamiento son considerados como parte del conjunto  $U$ . Para cada conjunto de datos el entrenamiento se repite

tantas veces como el número de ejemplos positivos existentes, iniciando  $P$  con un ejemplo diferente cada vez. Los estadísticos resultantes de estas ejecuciones son promediados.

A lo largo de la experimentación se evalúan dos medidas de disimilitud que han sido ampliamente utilizadas en el dominio temporal (función  $d$  en Algoritmo 4.2): DTW y la distancia euclidiana. La banda de Sakoe-Chiba es utilizada durante el cálculo de DTW, con tamaño de ventana  $w = 4\%$  de la longitud de la serie, siguiendo la recomendaciones de Kurbalija y colaboradores [90]. Para evaluar la distancia euclidiana se utilizó la banda de Sakoe-Chiba con  $w = 0\%$ .

## 4.4 Resultados experimentales

En esta sección se presentan los resultados experimentales obtenidos, los cuales son analizados utilizando pruebas estadísticas que permitan validar las conclusiones derivadas de dichos resultados. El análisis estadístico se ha organizado en dos etapas. La primera etapa está dedicada a la comparación entre los cinco criterios de parada propuestos con el objetivo de seleccionar el más competitivo de ellos (sección 4.4.1). La segunda etapa está orientada a la comparación entre el criterio ganador y las propuestas de la literatura que han abordado el problema del criterio de parada utilizando el método Self-training (WK, RW y BHRK) (sección 4.4.2).

Para contrastar los resultados obtenidos se utilizan pruebas estadísticas no paramétricas. Concretamente, para la primera etapa de evaluación, se sigue la metodología propuesta por García y colaboradores en el año 2010 [62], la cual recomienda utilizar la prueba alineada de Friedman [75] para comparaciones múltiples con el propósito de detectar diferencias significativas entre los criterios de parada propuestos. Posteriormente, se recomienda aplicar la prueba pareada de Hochberg [74] para detectar dónde se encuentran dichas diferencias. Para la segunda etapa, se utiliza la metodología propuesta por García y Herrera [63], la cual recomienda la utilización de la prueba de Friedman [58] y posteriormente la aplicación de la prueba pareada de Bergmann-Hommel [15].

### 4.4.1 Comparación entre los criterios de parada propuestos

Las Tablas 4.3 y 4.4 muestran los valores de precisión y F-Measure tanto para la distancia euclidiana como para DTW. La Figura 4.5 muestra estos resultados mediante un diagrama de cajas.

A partir de los resultados de precisión obtenidos por los criterios propuestos se efectúa una comparación para determinar cuál de ellos es el más competitivo. Con este objetivo se aplica la prueba alineada de Friedman para detectar diferencias significativas con un nivel de significación de  $\alpha = 0.05$  para ambas medidas de disimilitud. La prueba detecta diferencias significativas en ambos casos (para la distancia euclidiana el valor es de 0.013 y para DTW es de 0.014). La Tabla 4.5 muestra el ranking promedio de los criterios.

Conjunto de datos	Precisión					F-Measure				
	<i>SC1</i>	<i>SC2</i>	<i>SC3</i>	<i>SC4</i>	<i>SC5</i>	<i>SC1</i>	<i>SC2</i>	<i>SC3</i>	<i>SC4</i>	<i>SC5</i>
ECG	<b>0.990</b>	0.912	0.852	<b>0.990</b>	<b>0.990</b>	0.601	0.400	<b>0.726</b>	0.601	0.601
FaceAll	<b>1.000</b>	0.782	0.085	0.830	<b>1.000</b>	0.450	<b>0.572</b>	0.193	0.489	0.450
FacesUCR	0.415	<b>0.923</b>	0.415	0.610	0.415	0.484	<b>0.748</b>	0.484	0.617	0.484
Gun	<b>0.697</b>	<b>0.697</b>	0.416	<b>0.697</b>	<b>0.697</b>	0.572	<b>0.637</b>	0.448	0.572	0.572
Italy	0.809	0.673	<b>0.934</b>	0.861	0.809	0.420	<b>0.504</b>	0.400	0.412	0.420
Lighting7	0.393	0.325	<b>0.808</b>	0.325	0.393	<b>0.500</b>	0.442	0.361	0.442	<b>0.500</b>
MALLAT	<b>0.413</b>	0.303	0.303	<b>0.413</b>	<b>0.413</b>	<b>0.625</b>	0.500	0.500	<b>0.625</b>	<b>0.625</b>
Olive	<b>0.923</b>	0.716	0.834	0.728	<b>0.923</b>	0.451	<b>0.812</b>	0.510	0.745	0.451
Sony	<b>1.000</b>	0.949	<b>1.000</b>	0.954	<b>1.000</b>	0.488	<b>0.746</b>	0.293	0.639	0.488
SonyII	<b>0.727</b>	0.615	0.381	0.655	<b>0.727</b>	0.384	0.511	<b>0.742</b>	0.485	0.384
Trace	<b>0.112</b>	<b>0.112</b>	<b>0.112</b>	<b>0.112</b>	<b>0.112</b>	<b>0.704</b>	<b>0.704</b>	<b>0.704</b>	<b>0.704</b>	<b>0.704</b>
Two	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.967	0.920	<b>0.981</b>	0.967	0.967
Wafer	0.683	0.689	<b>0.824</b>	0.683	0.683	0.597	<b>0.598</b>	0.408	0.597	0.597
Word	0.382	0.210	<b>0.694</b>	0.382	0.382	<b>0.734</b>	0.561	0.483	<b>0.734</b>	<b>0.734</b>
WordsS	0.552	0.666	0.645	<b>0.696</b>	0.552	0.554	<b>0.702</b>	0.477	0.616	0.554

Tabla 4.3: Resultados de precisión y F-Measure obtenidos para los criterios de parada propuestos con la distancia euclidiana.

Conjunto de datos	Precisión					F-Measure				
	SC1	SC2	SC3	SC4	SC5	SC1	SC2	SC3	SC4	SC5
ECG	<b>0.982</b>	0.972	0.838	<b>0.982</b>	0.864	0.737	0.609	<b>0.869</b>	0.737	0.839
FaceAll	<b>1.000</b>	0.382	0.236	<b>1.000</b>	<b>1.000</b>	<b>0.896</b>	0.440	0.366	<b>0.896</b>	<b>0.896</b>
FacesUCR	0.467	0.636	0.316	<b>0.734</b>	0.467	0.534	0.677	0.471	<b>0.709</b>	0.534
Gun	<b>0.708</b>	<b>0.708</b>	<b>0.708</b>	<b>0.708</b>	<b>0.708</b>	<b>0.637</b>	<b>0.637</b>	<b>0.637</b>	<b>0.637</b>	<b>0.637</b>
Italy	<b>0.971</b>	0.916	<b>0.971</b>	<b>0.971</b>	<b>0.971</b>	0.366	<b>0.557</b>	0.326	0.485	0.366
Lighting7	<b>0.667</b>	0.583	0.492	<b>0.667</b>	<b>0.667</b>	0.229	0.382	<b>0.477</b>	0.326	0.229
MALLAT	0.395	0.293	<b>0.447</b>	0.395	0.395	<b>0.606</b>	0.488	0.551	<b>0.606</b>	<b>0.606</b>
Olive	0.641	0.651	0.654	<b>0.728</b>	0.641	0.689	<b>0.812</b>	0.471	0.745	0.689
Sony	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.543	<b>0.622</b>	0.270	0.543	0.543
SonyII	0.785	0.586	0.462	<b>0.889</b>	0.818	0.428	0.472	<b>0.674</b>	0.326	0.421
Trace	<b>0.543</b>	<b>0.543</b>	<b>0.543</b>	<b>0.543</b>	<b>0.543</b>	<b>0.704</b>	<b>0.704</b>	<b>0.704</b>	<b>0.704</b>	<b>0.704</b>
Two	<b>1.000</b>	<b>1.000</b>	0.526	<b>1.000</b>	<b>1.000</b>	0.896	<b>0.929</b>	0.668	0.896	0.896
Wafer	0.293	0.294	<b>0.575</b>	0.293	0.293	0.683	<b>0.684</b>	0.409	0.683	0.683
Word	<b>0.730</b>	0.719	0.710	<b>0.730</b>	<b>0.730</b>	0.727	<b>0.789</b>	0.727	0.727	0.727
WordsS	0.539	0.402	<b>0.939</b>	0.539	0.539	<b>0.711</b>	0.690	0.572	<b>0.711</b>	<b>0.711</b>

Tabla 4.4: Resultados de precisión y F-Measure obtenidos para los criterios de parada propuestos con la medida DTW.

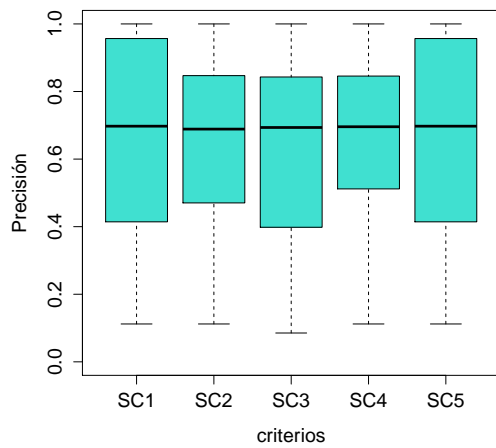
El criterio que resulta más preciso es seleccionado como método de control para la aplicación de las pruebas pareadas. El criterio seleccionado, para ambas medidas de distancia, es el *SC4*. Los criterios *SC2* y *SC3* muestran los más bajos valores de precisión. La prueba pareada de Hochberg detecta diferencias significativas, a favor del criterio *SC4*, en comparación con los criterios *SC2* y *SC3*, para la medida DTW.

Respecto al estadístico F-Measure, la prueba alineada de Friedman detecta diferencias significativas para un nivel de significación de  $\alpha = 0.05$  en ambas medidas de distancia (para la distancia euclidiana el  $p$ -valor es de 0.012 y para DTW es de 0.013). La Tabla 4.6 muestra el ranking obtenido entre los criterios de parada.

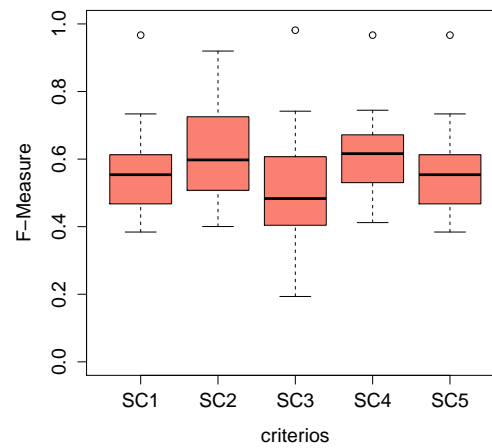
Una vez más, el criterio con mejor valor de F-Measure es seleccionado como método de control para la aplicación de la prueba de Hochberg. En este caso, los criterios seleccionados son el *SC4* y el *SC2*, para las medidas euclidiana y DTW, respectivamente. La prueba de Hochberg detecta diferencias significativas,



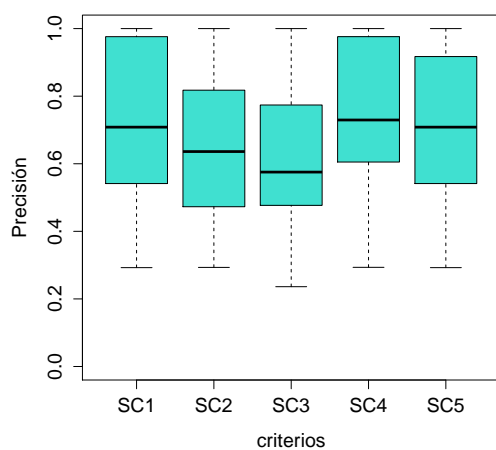
a favor de los métodos de control  $SC4$  y  $SC2$ , con respecto al criterio  $SC3$ , en ambas medidas de disimilitud. Es necesario aclarar que los elevados valores de F-Measure que ostenta el criterio  $SC2$  se deben a sus altos valores de sensibilidad, los cuales logra en detrimento de los valores de precisión que como se mostraron en la Tabla 4.5 son muy bajos.



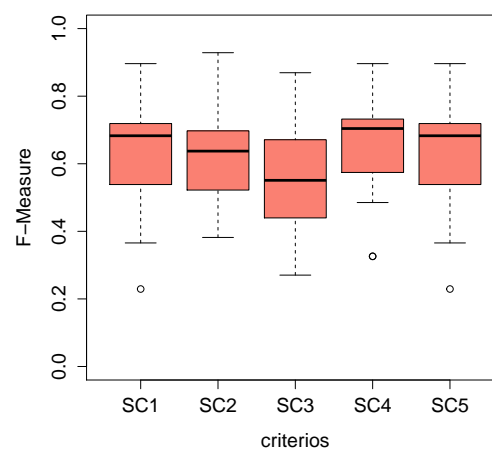
(a) Precisión (euclidiana).



(b) F-Measure (euclidiana).



(c) Precisión (DTW).



(d) F-Measure (DTW).

Figura 4.5: Diagrama de cajas que muestra los resultados de precisión y F-Measure para los cinco criterios propuestos.

Medida	Criterios	Ranking medio	$p_{Hochberg}$	Hipótesis
Euclidiana	<i>SC4</i>	<b>34.2</b>	–	–
	<i>SC1</i>	35.3	0.89	Aceptada
	<i>SC5</i>	35.3	0.89	Aceptada
	<i>SC3</i>	40.1	0.89	Aceptada
	<i>SC2</i>	45.0	0.70	Aceptada
DTW	<i>SC4</i>	<b>26.7</b>	–	–
	<i>SC1</i>	33.6	0.38	Aceptada
	<i>SC5</i>	36.4	0.38	Aceptada
	<i>SC3</i>	46.2	<b>0.04</b>	Rechazada
	<i>SC2</i>	47.0	<b>0.04</b>	Rechazada

Tabla 4.5: Ranking de la precisión obtenido mediante la prueba alineada de Friedman para ambas medidas de distancia. Las últimas columnas muestran los  $p$ -valores obtenidos al aplicar la prueba pareada de Hochberg.

Medida	Criterios	Ranking medio	$p_{Hochberg}$	Hipótesis
Euclidiana	<i>SC4</i>	<b>27.7</b>	–	–
	<i>SC2</i>	29.5	0.81	Aceptada
	<i>SC1</i>	39.8	0.25	Aceptada
	<i>SC5</i>	39.8	0.25	Aceptada
	<i>SC3</i>	53.1	<b><math>5.6 \cdot 10^{-3}</math></b>	Rechazada
DTW	<i>SC2</i>	<b>30.8</b>	–	–
	<i>SC4</i>	31.5	0.92	Aceptada
	<i>SC5</i>	37.0	0.86	Aceptada
	<i>SC1</i>	39.8	0.77	Aceptada
	<i>SC3</i>	50.8	<b>0.04</b>	Rechazada

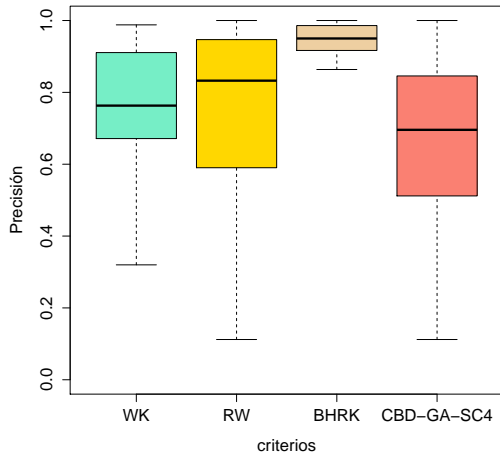
Tabla 4.6: Ranking del F-Measure obtenido mediante la prueba alineada de Friedman para ambas medidas de distancia. Las últimas columnas muestran los  $p$ -valores obtenidos al aplicar la prueba pareada de Hochberg.

Los resultados obtenidos de conjunto con las pruebas estadísticas aplicadas indican que los criterios *SC2* (curva ascendente) y *SC3* (curva estable) no son capaces de identificar correctamente el punto de parada, ya que dichos criterios presentan los valores más bajos de precisión. Del grupo de criterios basados en una curva (*SC1*, *SC2* y *SC3*) solamente resulta competitivo el *SC1* (curva descendente). El resto de los criterios (*SC4* y *SC5*), los cuales combinan elementos de dos curvas, ofrece mejores resultados en la tarea de identificar la frontera entre clases al adaptarse mejor a problemas con características diversas. En general, el *SC4* (combina la curva ascendente y la descendente) es el criterio más competitivo para el método CBD-GA, pues alcanza los mejores resultados tanto de precisión como de F-Measure.

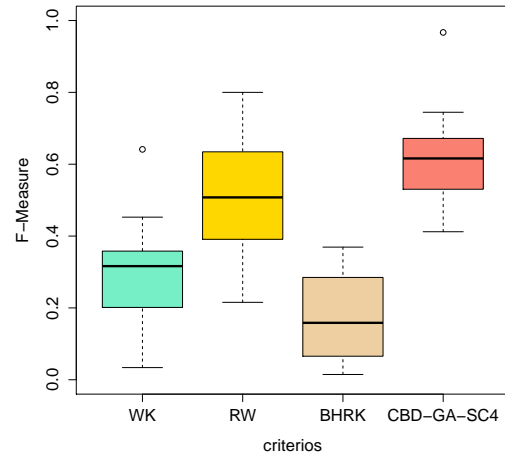
#### 4.4.2 Comparación con las propuestas de la literatura

Para evaluar la efectividad del método propuesto (CBD-GA con *SC4*) se llevó a cabo una comparación entre este y tres métodos del estado del arte. Las Tablas 4.7 y 4.8 muestran los resultados de precisión y F-Measure de estos métodos tanto para la distancia euclidiana como para DTW. La Figura 4.6 muestra estos resultados mediante un diagrama de cajas.

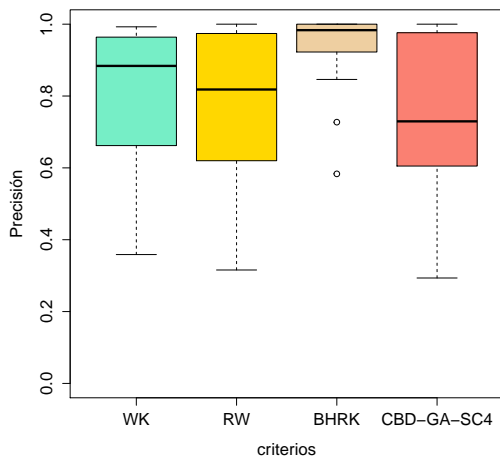
La prueba de Friedman, aplicada a los valores de precisión, detecta diferencias significativas en el grupo de criterios para ambas medidas de disimilitud (para la distancia euclidiana el  $p$ -valor es  $< 3.04 \cdot 10^{-4}$  y para DTW es  $< 5.86 \cdot 10^{-3}$ ). La Tabla 4.9 muestra el ranking promedio obtenido para estos valores de precisión. El método BHRK se destaca con los valores más altos de precisión. En la Tabla 4.10 se muestra una comparación pareada entre todos los métodos utilizando la prueba estadística de Bergmann-Hommel. El método BHRK supera significativamente (para un nivel de significación de  $\alpha = 0.10$ ) la precisión del resto de los métodos para ambas medidas de distancia. No obstante, el método propuesto solamente es superado de forma significativa por el BHRK.



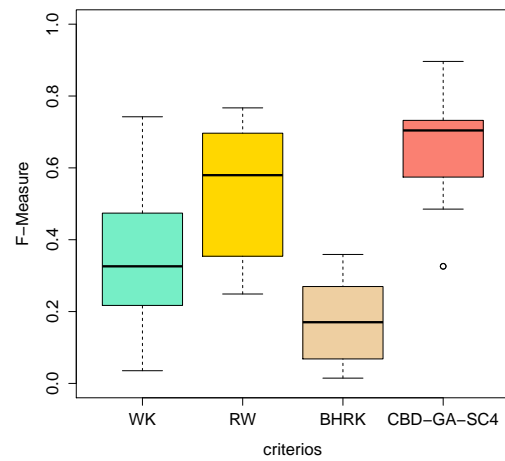
(a) Precisión (euclidiana).



(b) F-Measure (euclidiana).



(c) Precisión (DTW).



(d) F-Measure (DTW).

Figura 4.6: Diagrama de cajas que muestra los resultados de precisión y F-Measure para los criterios de la literatura en comparación con el criterio propuesto CBD-GA-SC4.

Considerando las pruebas estadísticas aplicadas a la precisión es posible concluir que el método BHRK ofrece mejores resultados que el resto, incluyendo nuestra propuesta. Al analizar a fondo la causa de este resultado encontramos que es una consecuencia directa de un comportamiento en extremo conservativo del criterio de parada BHRK. Esto provoca que el punto de parada se establezca demasiado pronto durante el aprendizaje, minimizado así la posibilidad de incluir ejemplos incorrectos en la clase positiva. Sin embargo, este comportamiento está asociado con bajos valores de sensibilidad, pues la terminación temprana del aprendizaje le impide aprender lo suficiente. Atendiendo a esto, consideramos que el método propuesto alcanza resultados competitivos en cuanto a precisión, principalmente cuando se utiliza DTW. Lo ocurrido ilustra la necesidad de tener en cuenta también los resultados de F-Measure al evaluar los criterios de parada, ya que este estadístico considera el balance entre precisión y sensibilidad.

Conjunto de datos	Precisión				F-Measure			
	<i>WK</i>	<i>RW</i>	<i>BHRK</i>	<i>SC4</i>	<i>WK</i>	<i>RW</i>	<i>BHRK</i>	<i>SC4</i>
ECG	0.971	0.986	0.975	<b>0.990</b>	0.109	<b>0.604</b>	0.021	0.601
FaceAll	0.836	0.574	<b>0.975</b>	0.830	0.175	0.327	0.095	<b>0.489</b>
FacesUCR	0.632	0.338	<b>0.923</b>	0.610	0.324	0.486	0.264	<b>0.617</b>
Gun	0.763	0.802	<b>0.917</b>	0.697	0.330	0.413	0.158	<b>0.572</b>
Italy	0.912	<b>0.971</b>	<b>0.971</b>	0.861	0.331	0.298	0.146	<b>0.412</b>
Lighting7	0.710	0.672	<b>0.917</b>	0.325	0.385	<b>0.680</b>	0.320	0.442
MALLAT	0.320	0.606	<b>1.000</b>	0.413	0.316	<b>0.800</b>	0.345	0.625
Olive	0.721	<b>0.923</b>	<b>0.923</b>	0.728	0.453	0.665	0.369	<b>0.745</b>
Sony	0.910	<b>1.000</b>	<b>1.000</b>	0.954	<b>0.641</b>	0.368	0.267	0.639
SonyII	0.558	<b>0.864</b>	<b>0.864</b>	0.655	0.426	0.216	0.303	<b>0.485</b>
Trace	0.619	0.112	<b>0.878</b>	0.112	0.305	<b>0.704</b>	0.237	<b>0.704</b>
Two	0.945	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.098	0.588	0.015	<b>0.967</b>
Wafer	0.988	0.833	<b>0.997</b>	0.683	0.034	0.480	0.038	<b>0.597</b>
Word	0.736	0.526	<b>0.904</b>	0.382	0.269	0.555	0.068	<b>0.734</b>
WordsS	0.894	0.845	<b>0.950</b>	0.696	0.228	0.508	0.063	<b>0.616</b>

Tabla 4.7: Resultados de precisión y F-Measure obtenidos para los criterios de parada propuestos en la literatura con la distancia euclidiana.

Conjunto de datos	Precisión				F-Measure			
	<i>WK</i>	<i>RW</i>	<i>BHRK</i>	<i>SC4</i>	<i>WK</i>	<i>RW</i>	<i>BHRK</i>	<i>SC4</i>
ECG	0.973	0.978	0.981	<b>0.982</b>	0.141	0.689	0.021	<b>0.737</b>
FaceAll	0.980	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.326	0.760	0.098	<b>0.896</b>
FacesUCR	0.679	0.316	<b>1.000</b>	0.734	0.546	0.471	0.286	<b>0.709</b>
Gun	0.743	0.708	<b>0.903</b>	0.708	0.306	0.391	0.170	<b>0.637</b>
Italy	0.755	<b>0.971</b>	<b>0.971</b>	<b>0.971</b>	0.426	0.262	0.149	<b>0.485</b>
Lighting7	0.359	0.533	0.583	<b>0.667</b>	0.222	0.317	0.216	<b>0.326</b>
MALLAT	0.545	0.783	<b>1.000</b>	0.395	0.367	0.580	0.345	<b>0.606</b>
Olive	0.645	<b>0.846</b>	<b>0.846</b>	0.728	0.594	0.725	0.359	<b>0.745</b>
Sony	0.955	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>0.742</b>	0.640	0.267	0.543
SonyII	0.517	0.818	0.727	<b>0.889</b>	<b>0.523</b>	0.269	0.273	0.326
Trace	0.946	0.543	<b>1.000</b>	0.543	0.346	<b>0.704</b>	0.186	<b>0.704</b>
Two	0.993	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.067	0.249	0.015	<b>0.896</b>
Wafer	0.992	0.483	<b>1.000</b>	0.293	0.035	0.567	0.037	<b>0.683</b>
Word	0.907	0.847	<b>0.942</b>	0.730	0.301	0.618	0.072	<b>0.727</b>
WordsS	0.884	0.696	<b>0.983</b>	0.539	0.213	<b>0.767</b>	0.064	0.711

Tabla 4.8: Resultados de precisión y F-Measure obtenidos para los criterios de parada propuestos en la literatura con la medida DTW.

Euclidiana		DTW	
Criterios	Ranking medio	Criterios	Ranking medio
BHRK	<b>1.33</b>	BHRK	<b>1.56</b>
RW	2.50	CBD-GA- <i>SC4</i>	2.60
WK	2.93	RW	2.63
CBD-GA- <i>SC4</i>	3.23	WK	3.20

Tabla 4.9: Ranking medio de la precisión obtenida por los métodos propuestos en la literatura y el método CBD-GA-*SC4* para ambas medidas de distancia.

i	Hipótesis	$p_{Bergmann}$	
		Euclidiana	DTW
1	BHRK vs. CBD-GA-SC4	<b><math>3.3 \cdot 10^{-4}</math></b>	<b>0.07</b>
2	WK vs. BHRK	<b><math>2.0 \cdot 10^{-3}</math></b>	<b><math>3.1 \cdot 10^{-3}</math></b>
3	RW vs. BHRK	<b>0.02</b>	<b>0.07</b>
4	RW vs. CBD-GA-SC4	0.35	0.94
5	WK vs. RW	0.35	0.60
6	WK vs. CBD-GA-SC4	0.52	0.60

Tabla 4.10:  $P$ -valores ajustados para la prueba pareada de Bergmann aplicada a los valores de precisión.

La prueba de Friedman aplicada a los valores de F-Measure detecta diferencias significativas en el grupo de criterios de parada para ambas medidas de disimilitud (para la distancia euclidiana el  $p$ -valor es  $< 7.21 \cdot 10^{-7}$  y para DTW es  $< 8.42 \cdot 10^{-7}$ ). La Tabla 4.11 muestra el ranking medio obtenido para estos valores donde el método CBD-GA-SC4 muestra los mejores resultados. Por su parte el método BHRK presenta los peores resultados de F-Measure, los cuales son causados por el problema de la detención temprana. La Tabla 4.12 muestra las comparaciones pareadas utilizando la prueba de Bergmann-Hommel. En estas, el método CBD-GA-SC4 supera significativamente (con un nivel de significación de  $\alpha = 0.10$ ) a los métodos del estado del arte en la mayoría de las comparaciones realizadas.

Los resultados experimentales indican que el método CBD-GA de conjunto con el criterio de parada SC4 obtiene un balance satisfactorio de precisión y sensibilidad durante la clasificación. La factibilidad de nuestra propuesta se encuentra sustentada mediante la comparación con varios métodos del estado del arte, donde las pruebas estadísticas aplicadas muestran que las capacidades transductivas del método propuesto superan significativamente las propuestas de la literatura.

Euclidiana		DTW	
Criterios	Ranking medio	Criterios	Ranking medio
CBD-GA-SC4	<b>1.30</b>	CBD-GA-SC4	<b>1.30</b>
RW	2.03	RW	2.16
WK	2.86	WK	2.66
BHRK	3.80	BHRK	3.86

Tabla 4.11: Ranking medio de F-Measure obtenido por los métodos propuestos en la literatura y el método CBD-GA-SC4 para ambas medidas de distancia.

i	Hipótesis	$p_{Bergmann}$	
		Euclidiana	DTW
1	BHRK vs. CBD-GA-SC4	<b><math>1.0 \cdot 10^{-6}</math></b>	<b>0.00</b>
2	RW vs. BHRK	<b><math>5.3 \cdot 10^{-4}</math></b>	<b><math>9.3 \cdot 10^{-4}</math></b>
3	WK vs. CBD-GA-SC4	<b><math>2.6 \cdot 10^{-3}</math></b>	<b>0.01</b>
4	WK vs. BHRK	<b>0.09</b>	<b>0.02</b>
5	WK vs. RW	<b>0.09</b>	0.28
6	RW vs. CBD-GA-SC4	0.11	<b>0.06</b>

Tabla 4.12:  $P$ -valores ajustados para la prueba pareada de Bergmann aplicada a los valores de F-Measure.

## 4.5 Sumario

Este capítulo estuvo dedicado al problema de la clasificación PU, la cual tiene como objetivo el aprendizaje de la clase positiva a partir de una cantidad limitada de ejemplos positivos y un volumen considerable de ejemplos sin etiquetar. Específicamente, se propusieron cinco criterios de parada para el algoritmo P1NNPU, el cual ha sido ampliamente utilizado en el dominio de las series temporales. El establecimiento del punto de parada es un elemento de vital importancia para este algoritmo pues es el encargado de limitar la inclusión errónea de ejemplos negativos en la clase positiva.



Se realizaron experimentos a partir de 15 conjuntos de datos temporales donde se compararon entre sí los criterios de parada propuestos y el más competitivo se comparó además con tres criterios de parada del estado del arte. Durante las comparaciones se utilizaron como medidas de rendimiento los estadísticos precisión y F-Measure de la clasificación transductiva, así como pruebas estadísticas no paramétricas para detectar y caracterizar las diferencias significativas.

Una vez analizados los resultados de la experimentación, es posible extraer las siguientes conclusiones:

- Para la mayoría de los criterios de parada, el uso de la medida DTW implica una mejora en los resultados de clasificación respecto a la euclidiana.
- De los cinco criterios de parada definidos para el método CBD-GA, se identificó como el más robusto el criterio *SC4*. Por tanto, la forma más adecuada para describir la frontera entre clases resultó ser el valor máximo entre la altura de la curva ascendente y la descendente.
- El uso de la técnica de análisis gráfico aplicada a las distancias mínimas obtenidas por el clasificador base 1NN ha resultado una técnica efectiva para identificar la frontera entre clases.
- Específicamente, la capacidad transductiva del método CBD-GA-*SC4* supera los métodos existentes del estado del arte y alcanza resultados competitivos tanto para la distancia euclidiana como para DTW.



# Capítulo 5

## **ssc: paquete de R para clasificación semi-supervisada**

Los métodos semi-supervisados de auto-etiquetado han demostrado ser una opción factible para enfrentar problemas de clasificación donde existe un número limitado de ejemplos etiquetados y abundantes ejemplos no etiquetados. Este tipo de método ha mostrado su efectividad en diversos dominios, incluido el dominio temporal, a partir de una adecuada selección de los métodos de clasificación base.

Resulta útil para los usuarios, disponer de una implementación de métodos de auto-etiquetado que permita establecer distintos clasificadores base así como diversas medidas de distancia. Una herramienta de este tipo facilitaría la comparación entre varios métodos de auto-etiquetado como parte del proceso de solución frente a un nuevo problema de clasificación. Además, posibilitaría la inclusión de nuevos métodos o transformaciones a los ya existentes.

Durante el desarrollo de esta tesis surgió la necesidad de contar con una herramienta de software que facilitara la ejecución de métodos de auto-etiquetado aplicados a diversos problemas provenientes del dominio temporal. Esto propició el desarrollo de un paquete de software, desarrollado en lenguaje R [77], el cual se nombró **ssc**. El paquete **ssc** implementa varios métodos de auto-etiquetado de probada efectividad, los cuales en su ma-

yoría formaron parte del estudio experimental desarrollado en el Capítulo 2. Los métodos inician el aprendizaje a partir de un conjunto de entrenamiento donde sólo un número limitado de ejemplos se encuentra etiquetado. Como resultado del entrenamiento semi-supervisado se obtiene un modelo de clasificación que depende directamente del método de auto-etiquetado utilizado. La hipótesis aprendida por dicho modelo puede utilizarse tanto para efectuar clasificación transductiva como inductiva. El paquete `ssc` se encuentra disponible en el repositorio CRAN (*Comprehensive R Archive Network*) a través del siguiente enlace: <https://CRAN.R-project.org/package=ssc>. También se ha diseñado una página web específica para el paquete: <http://sci2s.ugr.es/dicits/software/ssc>.

El capítulo se organiza en las siguientes secciones. En la sección 5.1 se mencionan los principales aspectos del lenguaje R y su utilización en el desarrollo de paquetes. En la sección 5.2 se describen los paquetes de R relacionados con la problemática a tratar así como otras bibliotecas de software. En la sección 5.3 se describen las principales funcionalidades del paquete `ssc`. En la sección 5.4 se ilustra cómo utilizar las funciones principales desde la preparación de los datos de entrenamiento hasta la etapa de clasificación donde se utiliza el modelo entrenado. Todos estos aspectos se describen mediante ejemplos pertenecientes al dominio temporal. Finalmente, en la sección 5.5 se presenta un breve resumen del capítulo.

## 5.1 Lenguaje de programación R

El lenguaje de computación estadística R [77] nació como una versión (*open source*) del lenguaje S, software propietario de Laboratorios Bell. Fue desarrollado por Robert Gentleman y Ross Ihaka en la Universidad de Auckland en 1993 y liberado bajo la licencia GPL (*Free Software Foundation's General Public License*). Aunque en su génesis era un lenguaje fundamentalmente para cálculos estadísticos, se ha convertido en el estándar *de facto* para el análisis de datos, y es empleado por cientos de miles de usuarios, tanto del ámbito académico como empresarial, en todo el mundo.

R es un lenguaje de programación que provee un poderoso ambiente interactivo para el análisis de datos, la visualización, la aplicación de técnicas de

minería de datos y el análisis estadístico, entre otras aplicaciones. En la actualidad es mantenido y desarrollado por un equipo de trabajo [131]. Entre las principales ventajas de R se encuentra la posibilidad de utilizar múltiples paquetes con finalidades específicas. El repositorio CRAN (<https://cran.r-project.org/>) es el sitio oficial y la principal fuente de paquetes y software de R. Otras de las principales características y facilidades que brinda este lenguaje se ofrecen a continuación:

- Provee una variedad de estructuras de datos (*list*, *matrix*, *vector* y *data.frame*, entre otros) que facilitan el procesamiento de los mismos. Además, también se brindan múltiples estrategias de manipulación a partir de estas.
- Soporta programación funcional y orientada a objetos. Para el primer caso, R provee un conjunto de funciones de alto nivel (*Reduce*, *Filter* y *Map*, entre otras) que facilitan la creación de nuevas funcionalidades. Para la programación orientada a objetos se utilizan varias técnicas, entre ellas se encuentran los estilos *S3* y *S4*.
- El desempeño de los paquetes de R puede mejorarse mediante la adición de código escrito en los lenguajes C, C++ y Fortran.
- Provee un conjunto de funciones primitivas adecuadamente documentadas que facilitan el análisis estadístico (*t.test*, *wilcox.test* y *friedman.test*, por sólo mencionar algunos).
- Es un lenguaje libre y de código abierto que funciona en múltiples plataformas (Windows, Mac OS X y Linux, entre otros).

Considerando todos estos beneficios, no cabe duda que R ofrece un entorno adecuado que fomenta la colaboración<sup>1</sup> y el desarrollo de nuevos paquetes<sup>2</sup>. Un estudio reciente realizado por KDnuggets [1] reveló que R alcanzó el primer puesto de los lenguajes de programación utilizados para el análisis y la minería de datos en el año 2015.

---

<sup>1</sup><http://rwiki.sciviews.org/doku.php>

<sup>2</sup><http://r-forge.r-project.org>

### 5.1.1 Repositorio CRAN

Gran parte de la hegemonía alcanzada por el lenguaje R se debe al nutrido conjunto de extensiones en forma de paquetes de software, que actualmente superan los 9000. Pero la razón no es sólo el número sino la política de gestión del sistema de publicación, que lleva inmerso un control de calidad. La comunidad de usuarios de R tiene muy asentada la necesidad de compartir código. Por este motivo, se ha dotado del repositorio de paquetes CRAN.

Para obtener el respaldo “oficial” a un paquete en R, debe ser publicado en el CRAN que es el repositorio oficial de paquetes desarrollados en este lenguaje. Para que el software sea aceptado debe cumplir un conjunto de requisitos<sup>3</sup>. Básicamente, estos establecen aspectos de funcionalidad, calidad de código y documentación. Las propuestas de paquetes son evaluadas detenidamente antes de su aceptación. Si un paquete candidato no pasa esos filtros, es rechazado. Además, estos son criterios dinámicos. Con las sucesivas actualizaciones del motor de R, los paquetes son revisados y actualizados. Si en algún momento, estos dejan de cumplir los requisitos, se solicita al autor que los revise. Si no se realizan las modificaciones pertinentes, los paquetes son retirados del repositorio. Esta política, junto con los múltiples foros<sup>4</sup> de usuarios de R donde se analiza en detalle la funcionalidad y el rendimiento de los paquetes, constituye un filtro de calidad muy efectivo.

El soporte físico del CRAN es una red de servidores web y ftp distribuidos por todo el mundo, que almacenan copias actualizadas del código y documentación de R y sus paquetes. Cada uno de esos servidores se llaman espejos (*mirrors*). Actualmente hay más de 100 servidores espejo<sup>5</sup>, además del central, distribuidos en el mundo según se muestra en la Figura 5.1.

---

<sup>3</sup><https://cran.r-project.org/web/packages/policies.html>

<sup>4</sup><https://www.r-bloggers.com>

<sup>5</sup><https://cran.r-project.org/mirrors.html>



Figura 5.1: Distribución mundial de servidores espejo de CRAN.

### 5.1.2 Estructura de un paquete de R

Un paquete de R consiste en una colección de funciones diseñadas para tratar una clase de problema mediante un determinado conjunto de métodos o conceptos. Normalmente un paquete contiene varios algoritmos relacionados entre sí. La estructura de un paquete [132] es simple y consiste básicamente en un directorio que tiene un nombre similar al del paquete y contiene los siguientes elementos:

- El directorio `R/` que contiene todo el código del paquete programado en dicho lenguaje.
- El directorio `src/` es opcional y contiene, en caso de que existan, los códigos programados en otros lenguajes como C, C++ y Fortran.
- El directorio `data/` es opcional y contiene conjuntos de datos que se utilizan en los ejemplos.
- El directorio `man/` contiene los archivos de documentación relacionados con cada función que se ha incluido en el paquete.
- El archivo de texto “DESCRIPTION” contiene la descripción general del paquete incluyendo título del paquete, los autores, el tipo de licencia, la fecha de la última versión y las dependencias con otros paquetes, entre otras informaciones.

## 5.2 Paquetes y bibliotecas relacionados con la SSC

Existen pocos paquetes públicos de R diseñados para realizar SSC y la mayoría de estos pertenecen al paradigma generativo. A continuación se analizarán los paquetes más representativos, los cuales se describen a continuación:

- `upclass` [140]: utiliza ejemplos de entrenamiento etiquetados y no etiquetados en la construcción de un método de clasificación mediante modelos de mixtura Gaussianos. El algoritmo EM (*Expectation Maximization*) es utilizado en la obtención de los parámetros del modelo y en la clasificación de los ejemplos no etiquetados.
- `bgmm` [16]: este paquete implementa métodos parcialmente supervisados para el modelado de mixturas.
- `Rmixmod` [91]: aporta una herramienta para la exploración y el análisis de datos, la cual brinda solución a problemas de clasificación y búsqueda de conglomerados mediante el ajuste de un modelo de mixtura a un conjunto de datos suministrado. Dicha herramienta puede utilizarse en escenarios semi-supervisados donde el conjuntos de datos se encuentre parcialmente etiquetado.
- `spa` [40]: brinda soporte para realiza SSC mediante modelos basados en grafos y regresión lineal.
- `DMwR` [148]: cubre una amplia colección de métodos para la minería de datos e incluye entre sus funciones el método Self-training.

Respecto a los métodos semi-supervisados de auto-etiquetado, no se han encontrado paquetes de R que implementen esta clase de métodos. Solamente el paquete `DMwR` [148] incluye entre sus funciones un método de este tipo.

Existen otras herramientas populares de minería de datos que han sido extendidas al paradigma semi-supervisado. Por ejemplo, la herramienta KEEL [3] escrita en Java, que ofreció inicialmente soporte sólo para aprendizaje supervisado y no supervisado, ha sido extendida con un módulo para realizar SSC. Dicho



módulo [150] incluye una amplia variedad de métodos de auto-etiquetado. La principal limitante de esta implementación es que los clasificadores base a utilizar se encuentran restringidos a los algoritmos supervisados y las medidas de distancia que están ya incluidas en KEEL.

El paquete *ssc* que se propone supera esta dificultad ofreciendo a los usuarios varias formas de suministrar los conjuntos de datos para entrenamiento así como la posibilidad de definir nuevos algoritmos base.

### 5.3 Arquitectura del paquete *ssc*

En esta sección se presentan las principales funcionalidades que brinda el paquete *ssc* y cómo utilizarlas. Estas se agrupan en cuatro categorías las cuales son: entrenamiento de un modelo semi-supervisado a partir de datos, clasificación de ejemplos mediante un modelo semi-supervisado entrenado, evaluación de los resultados de clasificación en un conjunto de prueba y la posibilidad de especificar nuevos modelos de clasificadores base. En la Figura 5.2 se muestran las funcionalidades mencionadas así como las funciones implementadas que las soportan dentro del paquete *ssc*. El Apéndice E incluye otros detalles adicionales sobre el esfuerzo requerido durante el desarrollo de este software.

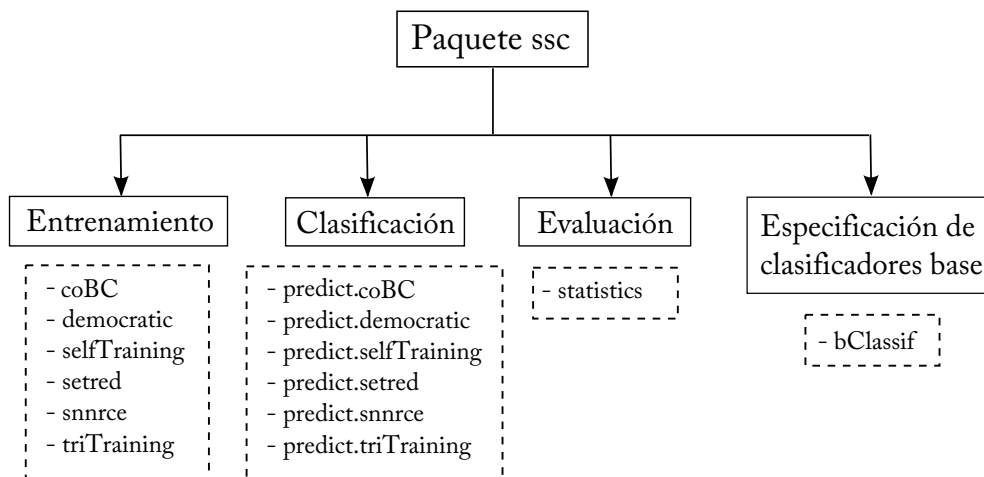


Figura 5.2: Principales funcionalidades y su implementación en el paquete *ssc*.

El paquete *ssc* contiene la implementación de seis métodos de auto-etiquetado que fueron previamente caracterizados en la sección 1.2, estos son:

- Co-bagging [53]
- Democratic [171]
- Self-training [165]
- SETRED [96]
- SNNRCE [155]
- Tri-training [172]

El proceso completo para entrenar un modelo, utilizando el paquete `ssc`, se puede resumir en el diagrama de la Figura 5.3. Durante la etapa de entrenamiento se utiliza alguna de las funciones suministradas en el paquete con este fin, la cual tendrá como argumentos los ejemplos de entrenamiento y otros parámetros específicos del modelo seleccionado. En la etapa de clasificación se utiliza la función `predict`. Dicha función sigue el estilo de clases *S3* y por tanto la clasificación se efectúa en dependencia de la clase del modelo entrenado.

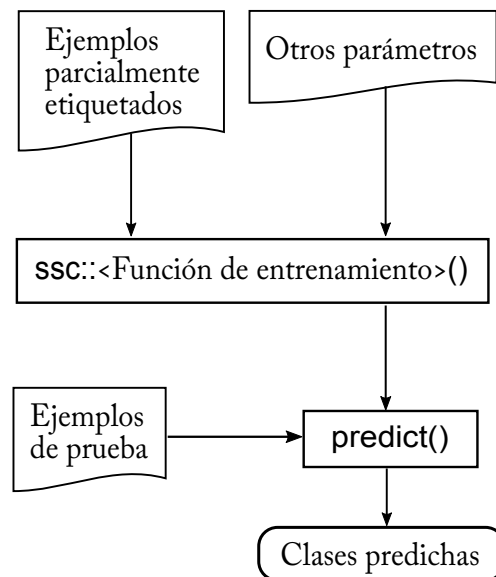


Figura 5.3: Entrenamiento de un modelo semi-supervisado con el paquete `ssc`.

### 5.3.1 Funciones de entrenamiento

Para cada uno de estos métodos semi-supervisados se implementó una función que es la encargada de entrenar y devolver un modelo del tipo seleccionado.

Todos los argumentos de las funciones implementadas se muestran en la Tabla 5.1.

Funciones	<code>x</code>	<code>y</code>	<code>dist</code>	<code>dist.use</code>	<code>bclassif</code>	<code>bclassifs</code>	<code>min.amount</code>	<code>u</code>	<code>max.iter</code>	<code>N</code>	<code>theta</code>	<code>alpha</code>
<code>coBC</code>	✓	✓	✓		✓		✓	✓	✓	✓		
<code>democratic</code>	✓	✓	✓	✓		✓						
<code>selfTraining</code>	✓	✓	✓		✓		✓		✓			
<code>setred</code>	✓	✓	✓		✓		✓		✓		✓	
<code>snrce</code>	✓	✓	✓									✓
<code>triTraining</code>	✓	✓	✓		✓							

Tabla 5.1: Funciones de entrenamiento y parámetros utilizados en `ssc`.

La implementación de los métodos semi-supervisados y de los clasificadores base en el paquete `ssc` sigue un enfoque basado en distancia. Esto significa que durante las fases de entrenamiento y clasificación se necesita suministrar las distancias entre los ejemplos implicados. Con este propósito se han definido dos argumentos que son comunes para todos los métodos: `x` y `dist`, a través de los cuales se suministran los ejemplos de entrenamiento. Específicamente, el parámetro `dist` puede tomar tres tipos de valores posibles:

- **la cadena "matrix"**: significa que el parámetro `x` es una matriz donde se encuentran precalculadas las distancias entre todos los ejemplos de entrenamiento.
- **nombre de distancia**: especifica el nombre de una función de distancia que se encuentre ya implementada en el paquete `proxy`. En este caso `x` se interpreta como una matriz donde cada fila representa un ejemplo.
- **una función**: una función de distancia definida por el usuario, la cual permite calcular la distancia entre cada par de ejemplos almacenados en `x`.

El tercer y último argumento común para todas las funciones es `y`. En este vector se suministran las clases de los ejemplos de entrenamiento inicialmente etiquetados. Los ejemplos no etiquetados se identifican en `y` mediante el valor `NA` (*Not Available*).

Las funciones de entrenamiento correspondientes a los métodos de auto-etiquetado utilizan al menos un clasificador base. Dicho clasificador puede o no estar fijado de antemano en dependencia del método. En el caso de las funciones: `selfTraining`, `setred`, `triTraining` y `coBc`, el tipo de clasificador base puede ser establecido a través del argumento `base.alg`. La función `democratic` tiene definido el argumento `base.algs` para especificar el conjunto de algoritmos que serán utilizados como clasificadores base, siguiendo un enfoque multiaprendizaje. Por otra parte, la función `snnrce` utiliza como clasificador base el 1NN y según la definición del método este no es modificable.

Todas las funciones de entrenamiento con la excepción de `democratic` utilizan por defecto el clasificador base 1NN. Este se construye utilizando la función `baseA0neNN`. Para definir un clasificador base diferente, el paquete `ssc` provee la función `bClassif` la cual se abordará con más detalle en la [sección 5.4.5](#). Los restantes argumentos que aparecen en la [Tabla 5.1](#) especifican aspectos particulares de cada método de auto-etiquetado. Dichos argumentos han sido descritos de manera detallada en el [Apéndice A](#).

### 5.3.2 Funciones de clasificación

A partir del entrenamiento se obtiene como resultado un objeto del mismo tipo que el modelo entrenado. Este objeto guarda la información necesaria para poder realizar clasificación tanto inductiva como transductiva. Esta tarea se realiza mediante la función `predict`. Los parámetros de esta función se describen a continuación:

- `object` un modelo semi-supervisado previamente entrenado.
- `x` una matriz que tiene dos posibles interpretaciones en dependencia de la forma en que se entrenó el modelo.

La función `predict` está implementada utilizando el estilo de clases *S3* y por consiguiente la clase del primer parámetro (`object`), en este caso el modelo, determina la función que se utiliza para clasificar. El segundo parámetro está determinado por la manera en que se entrenó el modelo. Si el modelo se entrenó a partir de una matriz de distancia, entonces el valor esperado en el parámetro

`x` es la matriz de distancia entre los ejemplos a clasificar y los ejemplos de entrenamiento. Si el modelo se entrenó utilizando una matriz de ejemplos, entonces el valor esperado en `x` es la matriz de los ejemplos a clasificar. Como resultado la función devuelve las clases predichas por el modelo.

### 5.3.3 Medidas de evaluación

El paquete `ssc` brinda estadísticos básicos para evaluar el desempeño de los métodos de clasificación en caso de que se encuentre disponible un conjunto de ejemplos de prueba. Estas medidas se evalúan a través de la función `statistics`. Dicha función posee los siguientes parámetros:

- `predicted` las clases predichas mediante un modelo de clasificación.
- `real` las clases reales a las que pertenecen los ejemplos evaluados.

Como resultado la función `statistics` devuelve tres medidas globales que describen el desempeño de la clasificación: kappa de Cohen, exactitud y F-Measure, en ese orden.

### 5.3.4 Especificación de un nuevo clasificador base

Entre las facilidades que ofrece el paquete `ssc` se encuentra la posibilidad de definir nuevos esquemas de clasificación para ser utilizados como clasificadores base por los métodos semi-supervisados. Con este fin se definió la función `bClassif`, la cual requiere los siguientes argumentos:

- `train` una función para entrenar un clasificador supervisado.
- `predClass` una función para obtener la clase de uno o varios ejemplos utilizando el clasificador entrenado con la función `train`.
- `predProb` una función para obtener las probabilidades por clase de uno o varios ejemplos utilizando el clasificador entrenado con la función `train`.
- `...` argumentos adicionales que se pasan a su vez como argumentos de la función `train`.

El funcionamiento de la función `bClassif` consiste en utilizar los argumentos suministrados para construir la especificación de un clasificador. Las funciones de entrenamiento están diseñadas para recibir como argumento dicha especificación en el formato retornado por `bClassif`, la cual utilizan para entrenar uno o varios clasificadores base en dependencia del modelo semi-supervisado. Una llamada típica a la función `train`, durante el entrenamiento de un clasificador base, incluye los siguientes argumentos:

- `x` matriz de distancia entre los ejemplos de entrenamiento.
- `y` las clases de los ejemplos de entrenamiento.
- ... argumentos adicionales requeridos por el clasificador, los cuales fueron previamente especificados en la función que se pasan a su vez como argumentos de la función `bClassif`.

Los datos principales para efectuar el entrenamiento son suministrados a través de los dos primeros argumentos. Como resultado la función `train` devuelve un modelo de clasificación entrenado.

Para obtener las clase o las probabilidades a partir del clasificador entrenado se utilizan las funciones `predClass` y `predProb`. Estas tienen dos parámetros similares en ambas funciones:

- `m` un modelo entrenado con la función `train`.
- `dists` una matriz de distancia entre los ejemplos que van a ser clasificados y los de entrenamiento.

La función `predClass` devuelve un vector con las clases obtenidas mientras que `predProb` retorna como resultado una matriz que contiene las probabilidades por clase. Dicha matriz tiene tantas columnas como clases existentes en el problema y tantas filas como los ejemplos que fueron clasificados.

## 5.4 Ejemplos de uso del paquete `ssc`

En esta sección se ilustran las principales funcionalidades que brinda el paquete `ssc`. Los ejemplos incluidos se han seleccionado específicamente del do-

minio temporal. Estos abarcan todo el proceso de entrenamiento de un modelo semi-supervisado desde la preparación de los datos de entrada hasta el proceso de clasificación utilizando el modelo entrenado. Posteriormente, se muestran otras de las facilidades que brinda el paquete *ssc*.

Como primer paso es necesario realizar la instalación del paquete. Este se encuentra público en CRAN y para su instalación sólo es necesario ejecutar el siguiente mandato en R:

```
install.packages("ssc")
```

### 5.4.1 Preparación de los datos de entrenamiento

El conjunto de datos que será utilizado durante los ejemplos es el GunPoint [135]. Este problema de clasificación pertenece al dominio de los videos de vigilancia y consta de dos clases. Todos los ejemplos fueron creados usando un actor del sexo masculino y uno del sexo femenino, durante una sesión de pruebas en la que los actores fueron grabados en un video. Como se muestra en la Figura 5.4, a través de la secuencia de video, el movimiento de la mano derecha en el eje de abscisas es rastreado y convertido en una serie de tiempo. La clase especifica si el actor realmente porta o no un arma de fuego. El conjunto de datos consta de 200 ejemplos (100 de cada clase).

A continuación se ilustra cómo realizar la preparación de los datos de entrenamiento para simular un contexto semi-supervisado de este problema y poder así utilizar las funciones del paquete *ssc*. El conjunto de datos GunPoint se encuentra público en el paquete de R *LPStimeSeries*. Una vez instalado dicho paquete es posible cargar el conjunto de datos GunPoint. Este, por defecto, se encuentra dividido en dos conjuntos de igual tamaño, los cuales procedemos a unir para agruparlos en un único conjunto de datos. Para crear una nueva partición del conjunto de datos GunPoint se utiliza el siguiente código de R:

```
library(ssc) # cargar paquetes
library(LPStimeSeries)
data("GunPoint") # cargar GunPoint
x <- rbind(GunPoint$trainseries, GunPoint$testseries)
```

```

y <- c(GunPoint$trainclass, GunPoint$testclass)

set.seed(1) # Fijar semilla
tra.idx <- sample(x = length(y), size = ceiling(length(y) * 0.5))
xtrain <- x[tra.idx,] # ejemplos entrenamiento
ytrain <- y[tra.idx] # clases asociadas

tra.na.idx <- sample(x = length(tra.idx),
size = ceiling(length(tra.idx) * 0.7))
ytrain[tra.na.idx] <- NA # eliminar clases del 70%
xttest <- x[tra.idx[tra.na.idx],] # ejemplos no etiquetados
yttest <- y[tra.idx[tra.na.idx]] # clases asociadas

tst.idx <- setdiff(1:length(y), tra.idx)
xitest <- x[tst.idx,] # ejemplos de prueba
yitest <- y[tst.idx] # classes asociadas

```

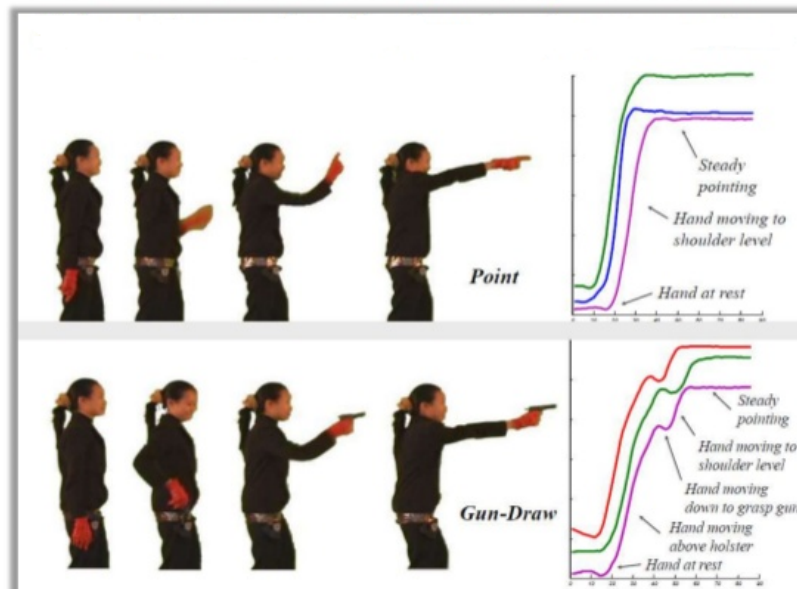


Figura 5.4: Ejemplo de una secuencia de video utilizada durante la creación del conjunto de datos GunPoint [135].



El conjunto de entrenamiento (`xtrain`) está compuesto por el 50% del total de ejemplos mientras que el de prueba (`xitest`) contiene el 50% restante. En el conjunto `xtrain` solamente el 30% de los ejemplos está etiquetado. Esta información se guarda en el vector de clases `ytrain` donde los valores posibles son `{1, 2, NA}`. Las posiciones ocupadas por el valor `NA` se corresponden con los ejemplos no etiquetados en `xtrain`. La selección de los ejemplos etiquetados en `xtrain` se realizó de manera aleatoria aunque se comprobó que todas las clases se encuentren representadas con al menos dos ejemplos etiquetados.

Las variables `xitest` y `xttest` son matrices de ejemplos almacenados por filas, las cuales serán utilizadas para probar la exactitud del modelo entrenado. Los ejemplos en `xitest` se utilizan para probar las capacidades inductivas del modelo mientras que los de `xttest` son para probar las capacidades transductivas. Por tal motivo en `xttest` están los ejemplos no etiquetados que fueron suministrados al modelo durante el entrenamiento. Las variables `yitest` y `yttest` guardan las clases correspondientes.

Con el objetivo de utilizar la opción “matrix” como valor para el parámetro `dist`, se calculan las matrices de distancia necesarias. El siguiente fragmento de código calcula dichas distancias utilizando la función de distancia euclidiana implementada en el paquete `proxy`:

```
library(proxy) # cargar paquete
dtrain <- as.matrix(dist(x = xtrain, method = "euclidean",
                        by_rows = TRUE))
ditest <- as.matrix(dist(x = xitest, y = xtrain,
                        method = "euclidean", by_rows = TRUE))
dttest <- as.matrix(dist(x = xttest, y = xtrain,
                        method = "euclidean", by_rows = TRUE))
```

La matriz `dtrain` es necesaria durante la fase de entrenamiento mientras que las matrices `ditest` y `dttest` se utilizan en la fase de clasificación. Es necesario resaltar el orden de los argumentos `x` e `y` pasados en las dos últimas llamadas a la función `dist`. Para obtener la matriz de distancia, requerida por el modelo, es necesario garantizar que `x` tome como valor el conjunto de prueba y el argumento `y` el de entrenamiento. La matriz de distancia obtenida de esta manera debe te-

ner las siguientes dimensiones: el número de filas igual al tamaño del conjunto de prueba y el número de columnas igual al tamaño del conjunto de entrenamiento.

## 5.4.2 Fase de entrenamiento

Existen varias formas de invocar las funciones de entrenamiento en dependencia de la opción que se utilice para el argumento `dist`. Cada una de estas opciones se ilustra mediante un ejemplo.

Para entrenar el modelo a partir de la matriz de distancia `dtrain`, calculada de antemano, se invoca la función `selfTraining` pasando como argumentos dicha matriz y las clases conocidas de los ejemplos de entrenamiento. Para el resto de los argumentos se utilizan sus valores por defecto:

```
m.selft1 <- selfTraining(x = dtrain, y = ytrain)
```

La segunda opción consiste en utilizar directamente los ejemplos de entrenamiento en `xtrain` y especificar en la función cuál es la distancia que debe usarse. De esta forma la distancia euclidiana se utiliza dentro de la propia función `selfTraining` para calcular la disimilitud entre todos los ejemplos en `xtrain`:

```
m.selft2 <- selfTraining(x = xtrain, y = ytrain,  
                        dist = "euclidean")
```

La tercera opción válida para el argumento `dist` es la especificación de una función. Como ejemplo, en la definición de la función se realiza una llamada a la distancia DTW implementada en el paquete de igual nombre `dtw`. La posibilidad de definir la función permite configurar determinados argumentos, por ejemplo la utilización de la restricción global de Sakoe-Chiba con un tamaño de ventana igual a diez:

```
m.selft3 <- selfTraining(x = xtrain, y = ytrain,  
                        dist = function(x, y){  
                          dtw::dtw(x, y, window.type = "sakoechiba",  
                                    window.size = 10)$distance  
                        })
```

El entrenamiento con otros métodos del paquete `ssc` se realiza de manera similar. A continuación se entrenan los modelos SETRED, SNNRCE, Tri-training y Co-bagging, utilizando la matriz de distancia `dtrain` ya calculada:

```
m.setred <- setred(x = dtrain, y = ytrain)
m.snnrce <- snnrce(x = dtrain, y = ytrain)
m.trit <- triTraining(x = dtrain, y = ytrain)
m.cobc <- coBC(x = dtrain, y = ytrain)
```

### Método Democratic

El método Democratic se diferencia del resto de los métodos incluidos en el paquete `ssc` respecto al esquema de aprendizaje. Dado que dicho método se basa en el multiaprendizaje, la forma de entrenarlo difiere levemente de los demás. En la función de entrenamiento `democratic` se esperan como argumentos las especificaciones de  $F$  clasificadores a partir de las cuales se entrenan los clasificadores base correspondientes. En el siguiente ejemplo se simula la idea del multiaprendizaje a partir de utilizar varios clasificadores 1NN con diferentes medidas de distancia, las cuales son: euclidiana, DTW y EDR. Para efectuar el entrenamiento se utiliza el siguiente código:

```
bclassifs <- list(bClassifOneNN(), bClassifOneNN(),
                 bClassifOneNN())
dist <- list("Euclidean",
            function(x, y){dtw::dtw(x, y)$distance},
            function(x, y){TSdist::EDRDistance(x, y,
                                                epsilon = 0.2)})
dist.use <- matrix(
  data = c(
    TRUE, FALSE, FALSE,
    FALSE, TRUE, FALSE,
    FALSE, FALSE, TRUE
  ),
  nrow = 3, byrow = TRUE
)
```

```
m.demo <- democratic(x = xtrain, y = ytrain,
                    bclassifs, dist, dist.use)
```

En conjunto, los tres últimos argumentos que se pasan a la función `democratic` describen el esquema de aprendizaje a utilizar. El argumento `bclassifs` contiene la lista con la especificación de cada esquema de clasificación. Los otros dos argumentos definen las distancias que van a ser utilizadas por cada clasificador base. La matriz `dist.use` indica qué funciones de distancia de la lista `dist` van a ser utilizadas con cada clasificador. En dicha matriz, las columnas representan las funciones de distancia y las filas las especificaciones de los clasificadores base. En el ejemplo se indica que el primer clasificador utilizará sólo la distancia euclidiana, el segundo DTW y el tercero la EDR.

### 5.4.3 Clasificación transductiva e inductiva

A continuación se ilustran dos variantes de clasificación. Los modelos que se utilizan fueron entrenados previamente en la sección 5.4.2.

En el primer ejemplo se utiliza el modelo `m.selft1`, el cual fue entrenado a partir de una matriz de distancia y por tanto se necesita una matriz de distancia para efectuar la clasificación. Como el objetivo de este ejemplo consiste en evaluar las capacidades inductivas del modelo, se le suministra como segundo argumento la matriz de distancia `ditest`:

```
d <- ditest[, m.selft1$included.insts]
p.selft1 <- predict(m.selft1, d)
confusionMatrix(table(p.selft1, yitest))$overall[1:2]
```

```
## Accuracy      Kappa
## 0.8600000 0.7185364
```

La variable interna `included.insts`, perteneciente al objeto `m.selft1`, guarda los índices de los ejemplos de entrenamiento que se utilizaron durante la construcción del modelo. Estos ejemplos son seleccionados por el esquema de aprendizaje utilizado durante la fase de entrenamiento y pasan a formar parte

del modelo final que se retorna como resultado. Por tal motivo se hace necesario seleccionar de la matriz original `ditest` las columnas que se corresponden con los ejemplos de entrenamiento que fueron seleccionados. La clasificación obtenida se guarda en el vector `p.selft1` y los estadísticos se obtienen utilizando la función `confusionMatrix` del paquete `caret`.

En el segundo ejemplo se utiliza el modelo `m.selft2` que fue entrenado a partir de la especificación de una distancia existente en el paquete `proxy`. Esta especificación se encuentra almacenada en una variable interna del modelo por tanto para evaluar las capacidades transductivas del mismo basta con suministrar directamente la matriz de ejemplos `xttest` como segundo argumento:

```
p.selft2 <- predict(m.selft2, xttest)
confusionMatrix(table(p.selft2, yttest))$overall[1:2]
```

```
## Accuracy      Kappa
## 0.9285714 0.8573757
```

Para este caso la función `predict` calcula la matriz de distancia entre los ejemplos de prueba en `xttest` y los ejemplos de entrenamiento seleccionados, los cuales se encuentran almacenados en la variable `tr.ints` del objeto `m.selft2`. Los resultados transductivos se evalúan a partir de la clasificación obtenida en `p.selft2` y las verdaderas clases guardadas en `yttest`.

#### 5.4.4 Comparación entre los modelos entrenados

Según se ha estudiado en los Capítulos 2 y 3, el comportamiento de los modelos semi-supervisados puede resultar muy diverso al enfrentar un mismo problema de clasificación. Por este motivo resulta importante establecer comparaciones para determinar el modelo más competitivo cuando se trata de resolver un problema concreto. A continuación utilizaremos las facilidades de evaluación que nos ofrece la función `statistics` para comparar el desempeño de los modelos entrenados. Específicamente, se va a evaluar la clasificación inductiva:

```
matrix.stat <- matrix(nrow = 3, ncol = 6)

d <- ditest[, m.selft1$included.insts]
p.selft1 <- predict(m.selft1, d) # clasificar con selfTraining
matrix.stat[,1] <- unlist(statistics(p.selft1, yitest))

d <- ditest[, m.setred$included.insts]
p.setred <- predict(m.setred, d) # clasificar con setred
matrix.stat[,2] <- unlist(statistics(p.setred, yitest))

d <- ditest[, m.snnrce$included.insts]
p.snnrce <- predict(m.snnrce, d) # clasificar con snnrce
matrix.stat[,3] <- unlist(statistics(p.snnrce, yitest))

d <- ditest[, m.trit$included.insts]
p.trit <- predict(m.trit, d) # clasificar con triTraining
matrix.stat[,4] <- unlist(statistics(p.trit, yitest))

d <- ditest[, m.cobc$included.insts]
p.cobc <- predict(m.cobc, d) # clasificar con coBC
matrix.stat[,5] <- unlist(statistics(p.cobc, yitest))

p.demo <- predict(m.demo, xitest) # clasificar con democratic
matrix.stat[,6] <- unlist(statistics(p.demo, yitest))

barplot(matrix.stat, beside = T, names.arg = c("SelfT", "SETRED",
        "SNNRCE", "TriT", "coBC", "Democratic"), ylim = c(0,1),
        col=c("cadetblue2", "cadetblue3", "cadetblue"),
        ylab = "Estadísticos de la clasificación")

legend(x = 18, y=0.4,
        fill = c("cadetblue2", "cadetblue3", "cadetblue"),
        legend=c("kappa", "Exactitud", "F-Measure"))
```

Como parte de la comparación se ha generado un gráfico de barras con los resultados obtenidos por cada modelo. Dicho gráfico se muestra en la Figura 5.5. Es posible apreciar cómo los métodos TriT y Democratic obtienen resultados superiores al resto. Este último es el que muestra mejor comportamiento en los tres estadísticos analizados.

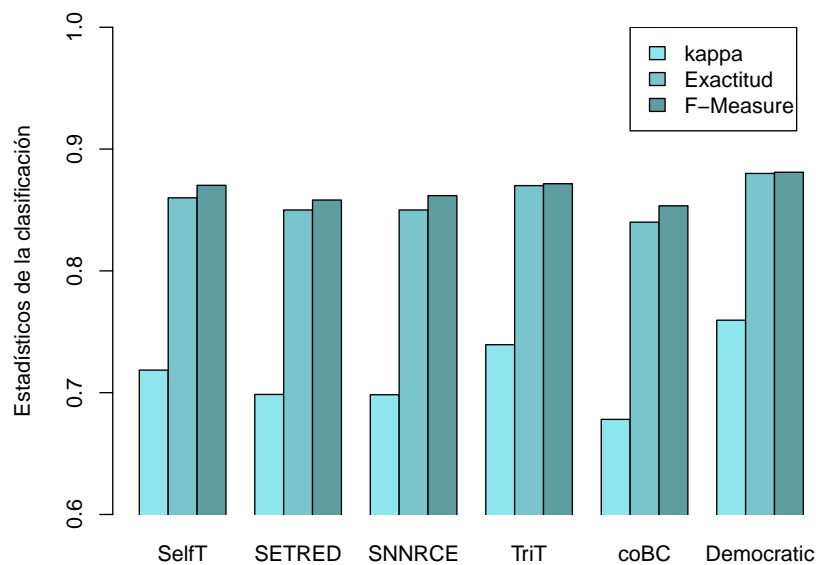


Figura 5.5: Comparación de los resultados de clasificación entre los modelos semi-supervisados entrenados para el problema GunPoint.

Otro análisis que resulta de interés realizar es la comparación del paradigma SSC con las cotas superior e inferior que nos brinda un clasificador supervisado. Al igual que en los análisis realizados en el Capítulo 2, se selecciona el 1NN para determinar las cotas de comparación. El siguiente código de R entrena dicho clasificador supervisado de forma que se puedan determinar ambas cotas:

```
labeled.idx <- which(!is.na(ytrain)) #índices etiquetados al inicio
xilabeled <- xtrain[labeled.idx,] #ejemplos etiquetados
yilabeled <- ytrain[labeled.idx] #clases asociadas
```

```

xflabeled <- x[tra.idx,] #todos ejemplos de entrenamiento
yflabeled <- y[tra.idx] #etiquetas asociadas

barplot(matrix.stat[2,], beside = T, names.arg =
  c("SelfT", "SETRED", "SNNRCE", "TriT", "coBC", "Democratic"),
  ylim = c(0.60, 1), col = "cadetblue3", ylab =
  "Exactitud de la clasificación", xpd = F)
legend(x = 5, y = 0.7, col = c("green", "red"),
  legend = c("Cota superior", "Cota inferior"), lty = 1, lwd = 2)

CIR <- oneNN(x = NULL, yilabeled) #crear el 1NN
pdist <- as.matrix(dist(x = xitest, y = xilabeled,
  method = "euclidean", by_rows = TRUE))
CIRclass <- predict(CIR, pdist) # clasificar con 1NN
abline(h = statistics(predicted = CIRclass, real = yitest)$accuracy,
  col = "red", lwd = 2)

CSR <- oneNN(x = NULL, yflabeled) #crear el 1NN
pdist <- as.matrix(dist(x = xitest, y = xflabeled,
  method = "euclidean", by_rows = TRUE))
CSRclass <- predict(CSR, pdist) # clasificar con 1NN
abline(h = statistics(predicted = CSRclass, real = yitest)$accuracy,
  col = "green", lwd = 2)

```

Como primer paso se hizo necesario construir dos conjuntos de entrenamiento para el clasificador 1NN. El primero contiene sólo los ejemplos etiquetados que se han suministrado a los modelos semi-supervisados (30 ejemplos etiquetados). El segundo contiene todos los ejemplos que se han suministrado durante la etapa de entrenamiento con sus respectivas etiquetas (100 ejemplos etiquetados). Los resultados de la clasificación supervisada, utilizando estos dos conjuntos, determinan las cotas inferior y superior de referencia respectivamente.

Como resultado del código anterior se obtiene una imagen para el estadístico exactitud donde se visualizan las cotas obtenidas y los resultados de exactitud de cada método (Figura 5.6). Los resultados del paradigma SSC ocupan un valor



intermedio entre las cotas inferior y superior para la mayoría de los métodos. La ganancia obtenida al realizar aprendizaje semi-supervisado resulta notable para el modelo Democratic.

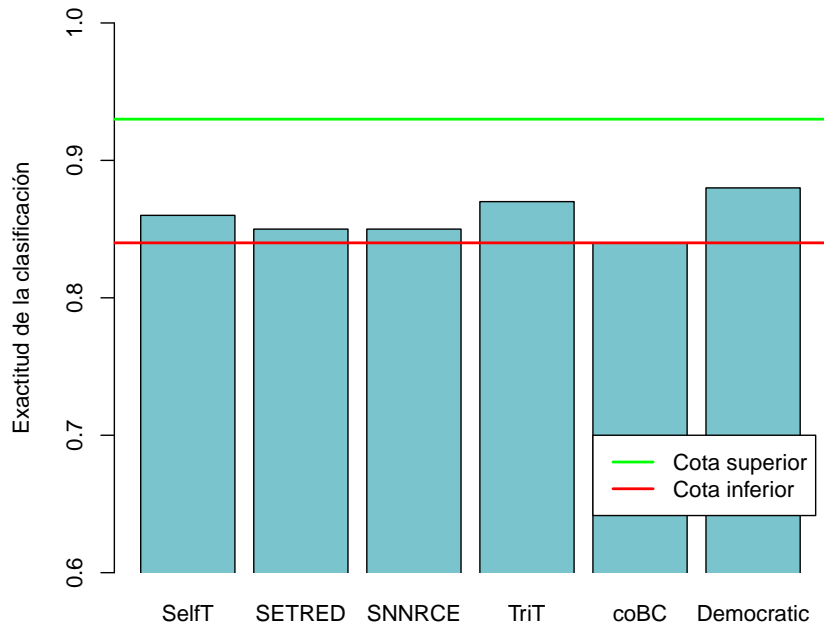


Figura 5.6: Cotas superior e inferior establecidas mediante el paradigma supervisado (1NN) para el problema GunPoint.

### 5.4.5 Otras facilidades

La función `bClassif` facilita la especificación de nuevos clasificadores base. Para ello se hace necesario definir las funciones `train`, `predClass` y `predProb`, suministradas como argumentos. El siguiente código muestra la especificación del clasificador 1NN ofrecida en el paquete `ssc` bajo el nombre de la función `bClassifOneNN`:

```
bClassifOneNN
```

```
## function ()
```

```
## {  
##   bClassif(train = oneNN,  
##           predClass = predClass.OneNN,  
##           predProb = predProb.OneNN)  
## }  
## <environment: namespace:ssc>
```

En el siguiente ejemplo se asume la existencia de las funciones `kNN`, `predClass.kNN` y `predProb.kNN`. La definición de una función `bClassifKNN` que permita crear la especificación de una instancia del clasificador  $k$ NN se puede definir de la siguiente forma:

```
bClassifKNN <- function(k = 3){  
  bClassif(train = kNN,  
            predClass = predClass.KNN,  
            predProb = predProb.KNN,  
            k = k)  
}
```

En este caso existe un argumento adicional que es el parámetro `k`, el cual por defecto tiene valor igual a tres. A partir de la función `bClassifKNN` es posible obtener la especificación de este clasificador para un valor particular de `k`. Por ejemplo, para obtener una especificación con un valor de `k` igual a cinco se invoca la función `bClassifKNN` de la siguiente forma:

```
bc5NN <- bClassifKNN(k = 5)
```

A partir de la especificación `bc5NN` obtenida puede entrenar la función `selfTraining` utilizando el nuevo esquema de clasificación.

```
model2 <- selfTraining(x = dtrain, y = ytrain,  
                      bclassif = bc5NN)
```

## 5.5 Sumario

Este capítulo se ha dedicado a la presentación de un paquete de R diseñado para realizar clasificación semi-supervisada mediante métodos de auto-etiquetado. El paquete `ssc` está escrito totalmente en lenguaje R y para utilizar algunas de las facilidades de la programación orientada a objetos se utilizó el estilo de clases S3. La utilización de este paquete ofrece varias ventajas, las cuales se enumeran a continuación:

- El paquete `ssc` incluye seis métodos semi-supervisados de auto-etiquetado con características diversas. Varios de estos han probado su efectividad frente a problemas provenientes de diversos dominios incluyendo el temporal.
- La selección de un lenguaje multiplataforma para su desarrollo simplifica su utilización con independencia del sistema operativo empleado por el usuario.
- El diseño del paquete `ssc` hace posible su extensión con nuevos algoritmos semi-supervisados de una manera simple. Además, el enfoque basado en distancias de su implementación permite la modelación de problemas diversos siempre que exista una función de disimilitud apropiada para comparar los ejemplos.
- El paquete `ssc` ofrece diferentes formas de introducir los datos de entrenamiento, incluyendo la variante de especificar funciones de distancia que se encuentren implementadas en otros paquetes de R.
- El diseño del paquete incluye una interfaz que permite definir las especificaciones de nuevos clasificadores base, los cuales pueden provenir de otros paquetes de R o ser programados en su totalidad por los usuarios.
- Los modelos retornados como salida del entrenamiento semi-supervisado pueden ser utilizados para realizar clasificación tanto transductiva como inductiva.

El paquete `ssc` se encuentra disponible en los siguientes enlaces:

<https://CRAN.R-project.org/package=ssc>

<http://sci2s.ugr.es/dicits/software/ssc>



# Conclusiones y trabajo futuro

En esta sección se resumen los resultados obtenidos durante el desarrollo de esta tesis así como las conclusiones que se pueden extraer de los mismos. Además, se presentan las publicaciones asociadas a la tesis y algunos comentarios sobre los trabajos futuros y líneas de investigación que se pueden derivar de la presente investigación.

## A. Conclusiones

La investigación realizada en esta tesis se ha centrado en estudiar, diseñar y evaluar métodos semi-supervisados de auto-etiquetado para abordar la clasificación de series temporales, considerando las características distintivas implicadas en el tratamiento de este tipo de dato. Los objetivos propuestos para esta tesis fueron los siguientes:

- Efectuar un estudio comparativo experimental entre varios métodos de auto-etiquetado de probada efectividad en aras de identificar cuáles ofrecen mejores resultados en la clasificación de series temporales.
- Desarrollar un estudio sobre la influencia que ejerce la complejidad de los problemas temporales, desde el punto de vista de la clasificación, en el desempeño de los métodos de auto-etiquetado.
- Proponer y validar criterios de parada para el algoritmo Self-training, orientados al aprendizaje semi-supervisado de una clase, que resulten más efectivos en comparación con los reportados en la literatura.
- Implementar un paquete de código abierto en lenguaje R para la aplicación

de métodos de auto-etiquetado en la solución de problemas de clasificación semi-supervisada.

En relación al primer objetivo, se efectuó un estudio comparativo entre varias técnicas de auto-etiquetado donde se exploraron diversos clasificadores base adaptados específicamente al tratamiento de series temporales. A partir de estos resultados se obtuvieron varias conclusiones sobre la aplicabilidad de este tipo de técnicas al dominio temporal. Respecto al desempeño de los clasificadores base, se ratificó el 1NN como una opción robusta en el contexto semi-supervisado por ofrecer las clasificaciones de mayor exactitud. Los restantes clasificadores base evaluados presentan resultados competitivos cuando se combinan sus hipótesis siguiendo un esquema de multiaprendizaje.

Respecto a los métodos semi-supervisados evaluados, se destacaron particularmente tres métodos de auto-etiquetado: SETRED, TriT-1NN y Democratic. Estos métodos presentaron resultados significativamente superiores al resto en términos de sus capacidades tanto inductivas como transductivas. El enfoque multclasificador, al que pertenecen los métodos TriT-1NN y Democratic, se destacó como una solución prometedora al problema de la clasificación semi-supervisada en el contexto de las series temporales. Finalmente, se realizó un análisis sobre los beneficios reales que aportan los métodos estudiados donde se observó un comportamiento competitivo de los mismos frente al paradigma supervisado.

Como respuesta al segundo objetivo, se desarrolló un estudio experimental acerca de la relación entre la complejidad asociada a los problemas de clasificación de datos temporales y los resultados obtenidos con las técnicas de auto-etiquetado estudiadas. Como parte de dicho estudio se evaluaron varias medidas de complejidad, las cuales fueron aplicadas al dominio temporal mediante el empleo de medidas de disimilitud. Los resultados obtenidos evidenciaron que en general los problemas de clasificación temporales resultan menos complejos cuando se utiliza la medida DTW. Respecto al estudio para diferentes proporciones de ejemplos etiquetados, se pudo constatar que las evaluaciones ofrecidas por las medidas de complejidad resultan coherentes incluso para conjuntos pequeños de ejemplos etiquetados.

Respecto a la relación de la complejidad con la exactitud obtenida por los métodos semi-supervisados evaluados, se obtuvieron fuertes correlaciones las

cuales permitieron identificar los aspectos que más influyen en la clasificación obtenida. Específicamente, se identificaron las medidas de complejidad que resultaron más relevantes: N1, N3 y T1. En cuanto a la competencia de los métodos semi-supervisados en relación a la complejidad, se caracterizaron los métodos de mejor desempeño. Los métodos Democratic, TriT-1NN y TriT-SVM presentaron un mejor comportamiento frente a problemas de baja complejidad. Por el contrario, el método SETRED se mostró más competitivo al enfrentar problemas de mediana y alta complejidad.

Para dar cumplimiento al tercer objetivo, se propusieron cinco criterios de parada para el algoritmo P1NNPU, el cual ha sido ampliamente utilizado en el dominio temporal para tratar la clasificación de problemas de una clase de manera semi-supervisada. Este algoritmo es una variante del Self-training donde el criterio de parada juega un papel crucial para impedir la inclusión errónea de ejemplos negativos en la clase positiva. El uso de la técnica de análisis gráfico, aplicada a las distancias mínimas obtenidas por el clasificador base 1NN, resultó una técnica efectiva para identificar la frontera entre clases. A partir de la experimentación realizada se identificó el criterio *SC4* como el más robusto de los cinco criterios propuestos. La capacidad transductiva del método CBD-GA-*SC4* superó los métodos existentes del estado del arte, alcanzando resultados competitivos tanto para la distancia euclidiana como para DTW.

Por último, para cumplir con el cuarto objetivo, se desarrolló un paquete de R llamado `ssc`. Este incluye métodos semi-supervisados para resolver problemas de clasificación con fines tanto transductivos como inductivos. El paquete `ssc` resulta fácil de extender con nuevos algoritmos semi-supervisados y ofrece diferentes formas de introducir los datos de entrenamiento, incluyendo la especificación de funciones de distancias que se encuentren implementadas en otros paquetes de R. El paquete también incluye la posibilidad de definir nuevos clasificadores base lo cual añade flexibilidad a esta propuesta. Actualmente, el paquete `ssc` se encuentra disponible en <https://CRAN.R-project.org/package=ssc>.

## B. Publicaciones Asociadas a la Tesis

A continuación se presenta un listado de las publicaciones asociadas a la tesis.

- Monografía:
  - M. González, C. Soto, Minería de datos para series temporales, Editorial Samuel Feijóo. ISBN: 978-959-250-924-5.
- Publicaciones en revistas internacionales:
  - M. González, C. Bergmeir, I. Triguero, Y. Rodríguez, J. M. Benítez, On the stopping criteria for  $k$ -Nearest Neighbor in positive unlabeled time series classification problems, *Information Sciences*, Vol. 328, pág. 42-59, 2016.
  - M. González, C. Bergmeir, I. Triguero, Y. Rodríguez, J. M. Benítez, Self-labeling techniques for semi-supervised time series classification: an empirical study, sometido en *Knowledge and Information Systems*.
  - M. González, O. Rosado, J. D. Rodríguez, C. Bergmeir, I. Triguero, J. M. Benítez, **ssc**: An R package for semi-supervised classification, sometido en *The R Journal*.
- Premios:
  - Premio Anual Provincial de la Academia de Ciencias de Cuba. 2016. Algoritmos de clasificación para series temporales.

## C. Trabajos Futuros

Esta última sección está dedicada a ofrecer las nuevas ideas, posibilidades de mejora y líneas de investigación que han surgido a partir del trabajo realizado en esta tesis. Se abren tres líneas principales de investigación:

- Ampliar el estudio comparativo entre técnicas de auto-etiquetado aplicadas a la clasificación de series temporales:



- 
- Estudiar otras técnicas de auto-etiquetado basadas en múltiples clasificadores, las cuales utilicen un mecanismo de tipo modificable para la adición de nuevos ejemplos.
  - Estudiar otras medidas de disimilitud entre series temporales tales como EDR, ERP y TWED.
- Estudiar el comportamiento de la complejidad asociada a los problemas de clasificación temporales en relación con diversas medidas de disimilitud elásticas.
  - Realizar el mantenimiento del paquete **ssc**:
    - Incorporar nuevos esquemas de clasificación para los algoritmos base que no estén restringidos al enfoque basado en distancias.
    - Incorporar en el paquete los avances que se realicen en el área de la SSC.



# Apéndice A

## Parámetros específicos de los métodos semi-supervisados estudiados

Este apéndice contiene la descripción de los parámetros particulares que poseen los métodos semi-supervisados involucrados en este estudio. De cada uno de ellos se incluye el nombre completo y entre paréntesis el nombre del argumento correspondiente tal y como aparece en el paquete `ssc`.

**Cantidad mínima de ejemplos no etiquetados (`min.amount`):** este parámetro se encuentra presente en los métodos Self-training, SETRED y Co-bagging. `min.amount` hace referencia a la cantidad mínima de ejemplos no etiquetados que deben existir en el conjunto  $U$  para que continúe el entrenamiento hacia la próxima iteración. Una vez que dicha cantidad sea menor que el número establecido en (`min.amount`), el entrenamiento termina. De esta forma se intenta evitar que los ejemplos de menor confianza sean etiquetados e incluidos en el modelo.

**Máximo de iteraciones (`max.iter`):** este parámetro se encuentra presente en los métodos: Self-training, SETRED y Co-bagging. `max.iter` representa la cantidad máxima de iteraciones que se pueden realizar durante el entrenamiento. Al igual que el parámetro anterior, intenta evitar que se etiqueten ejemplos dudosos durante el entrenamiento. También resulta útil cuando el

conjunto de entrenamiento es en extremo extenso.

**Cantidad de miembros (N):** este parámetro aparece en el método Co-bagging y representa la cantidad de clasificadores que se utilizan como parte del esquema de multclasificación. Mientras mayor sea este número con mayor lentitud transcurre el aprendizaje.

**Tamaño del subconjunto de ejemplos no etiquetados (u):** este parámetro aparece en el método Co-bagging y define la cantidad de ejemplos no etiquetados  $U'$  que son seleccionados de  $U$  de forma aleatoria en cada iteración. De esta forma, los ejemplos de mayor confianza son seleccionados a partir del subconjunto  $U'$  y no directamente de  $U$ , mitigando así el efecto que produce seleccionar siempre los ejemplos de mayor confianza del conjunto  $U$  que contiene todos los ejemplos disponibles.

**Umbral de significación  $\theta$  (theta):** este parámetro aparece en el método SETRED y está relacionado con la hipótesis que es necesario probar para determinar si un determinado ejemplo será o no removido del conjunto de entrenamiento. Con este fin se construye un grafo de vecindades y para cada ejemplo se determina si la cantidad de vecinos con clase diferente se comporta según lo esperado bajo la hipótesis nula. Cuando el ejemplo evaluado se ubica en la región izquierda de rechazo entonces se considera un buen ejemplo. El parámetro `theta` define la región crítica donde se prueba la hipótesis y mientras menor sea este valor más restrictiva resulta la selección de los ejemplos que son considerados buenos.

**Umbral de significación  $\alpha$  (alpha):** este parámetro aparece en el método SNNRCE y está relacionado con la hipótesis que es necesario probar para determinar si un determinado ejemplo es dudoso. Con este fin se construye un grafo de vecindades y para cada ejemplo se determina si la cantidad de vecinos con clase diferente se comporta según lo esperado bajo la hipótesis nula. Cuando el ejemplo evaluado se ubica en la región derecha de rechazo entonces se considera un mal ejemplo y el método procede a asignarle una etiqueta diferente a la actual. El parámetro `alpha` define la región crítica donde se prueba la hipótesis y mientras mayor sea este valor más ejemplos serán considerados dudosos.

# Apéndice B

## Resultados de los métodos semi-supervisados estudiados

Este apéndice contiene los resultados completos tanto transductivos como inductivos de cada método semi-supervisado evaluado (Sección [B.1](#)). También incluye las evaluaciones del algoritmo 1NN como cota de referencia tanto superior como inferior (Sección [B.2](#)). Todos estos valores son utilizados para calcular la ganancia de exactitud (respecto a la cota inferior de referencia) y la diferencia de exactitud (respecto a la cota superior de referencia).

### B.1 Resultados transductivos e inductivos de los métodos semi-supervisados

De cada método de auto-etiquetado se muestran los resultados de exactitud (E) y kappa (K). De cada uno de estos estadísticos se muestra la media ( $\mu$ ) y la desviación estándar ( $\sigma$ ) obtenida durante el proceso de validación cruzada con cinco particiones. Los estadísticos se evaluaron utilizando las medidas euclidiana y DTW.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.95	0.00	0.92	0.01	0.94	0.02	0.91	0.03	0.99	0.00	0.98	0.01	0.98	0.02	0.97	0.03
Chlorine	0.60	0.01	0.35	0.01	0.59	0.02	0.34	0.02	0.58	0.01	0.31	0.01	0.57	0.02	0.30	0.02
CinCECGt	0.98	0.01	0.97	0.01	0.98	0.01	0.98	0.01	0.98	0.01	0.97	0.01	0.98	0.01	0.98	0.02
Coffee	0.93	0.03	0.87	0.06	0.97	0.05	0.93	0.10	0.91	0.06	0.82	0.13	0.95	0.05	0.89	0.10
CricketX	0.33	0.04	0.27	0.04	0.36	0.05	0.30	0.06	0.48	0.03	0.43	0.03	0.47	0.04	0.42	0.04
CricketY	0.34	0.03	0.28	0.03	0.34	0.04	0.28	0.04	0.44	0.03	0.39	0.04	0.46	0.05	0.41	0.05
CricketZ	0.35	0.04	0.29	0.05	0.35	0.02	0.30	0.02	0.49	0.02	0.44	0.03	0.51	0.05	0.46	0.05
ECG	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.96	0.01	0.98	0.01	0.95	0.03
ECGFiveD	0.93	0.01	0.87	0.03	0.94	0.03	0.87	0.06	0.89	0.03	0.78	0.06	0.91	0.01	0.82	0.02
FaceAll	0.82	0.03	0.80	0.03	0.81	0.03	0.80	0.03	0.88	0.04	0.87	0.05	0.89	0.04	0.88	0.04
FacesUCR	0.80	0.02	0.78	0.03	0.80	0.02	0.78	0.02	0.91	0.02	0.90	0.03	0.91	0.03	0.90	0.03
Fish	0.57	0.05	0.50	0.06	0.62	0.03	0.55	0.04	0.46	0.08	0.37	0.09	0.51	0.11	0.42	0.12
GunPoint	0.80	0.02	0.60	0.04	0.78	0.03	0.57	0.07	0.85	0.06	0.71	0.12	0.82	0.07	0.65	0.14
Haptics	0.33	0.03	0.16	0.04	0.34	0.08	0.17	0.10	0.33	0.04	0.16	0.05	0.36	0.02	0.19	0.03
InlineS	0.25	0.02	0.13	0.03	0.29	0.02	0.17	0.03	0.29	0.03	0.17	0.04	0.32	0.04	0.20	0.05
Italy	0.96	0.01	0.93	0.01	0.96	0.02	0.92	0.03	0.94	0.00	0.88	0.01	0.94	0.01	0.87	0.02
Lighting2	0.59	0.03	0.15	0.09	0.56	0.14	0.07	0.30	0.67	0.12	0.30	0.27	0.59	0.17	0.17	0.34
Lighting7	0.40	0.03	0.30	0.04	0.46	0.10	0.36	0.12	0.49	0.07	0.40	0.08	0.52	0.05	0.43	0.07
MALLAT	0.92	0.02	0.91	0.02	0.92	0.01	0.91	0.01	0.93	0.02	0.92	0.02	0.94	0.01	0.93	0.01
MedicalI	0.52	0.04	0.36	0.05	0.52	0.05	0.37	0.05	0.57	0.03	0.42	0.03	0.58	0.04	0.43	0.04
MoteStrain	0.91	0.01	0.81	0.01	0.90	0.02	0.79	0.05	0.94	0.01	0.89	0.01	0.94	0.01	0.88	0.02
Olive	0.71	0.05	0.60	0.07	0.68	0.07	0.57	0.09	0.69	0.06	0.57	0.08	0.68	0.12	0.56	0.16
OSULeaf	0.43	0.06	0.30	0.08	0.41	0.07	0.28	0.08	0.49	0.03	0.38	0.04	0.44	0.10	0.32	0.12
Sony	0.94	0.03	0.88	0.06	0.95	0.02	0.90	0.04	0.93	0.03	0.87	0.06	0.93	0.02	0.86	0.05
SonyII	0.92	0.02	0.82	0.05	0.91	0.02	0.81	0.06	0.90	0.02	0.79	0.04	0.91	0.02	0.80	0.05
StarLightC	0.84	0.01	0.75	0.02	0.84	0.02	0.75	0.03	0.87	0.01	0.79	0.02	0.87	0.01	0.79	0.02
Synthetic	0.77	0.03	0.72	0.03	0.80	0.04	0.75	0.05	0.76	0.04	0.71	0.05	0.79	0.05	0.75	0.06
Trace	0.59	0.03	0.46	0.04	0.56	0.03	0.41	0.04	0.78	0.04	0.71	0.05	0.80	0.08	0.73	0.10
Two	0.77	0.01	0.69	0.01	0.77	0.01	0.70	0.02	0.97	0.01	0.97	0.01	0.97	0.00	0.96	0.01
TwoLeadE	0.90	0.02	0.81	0.03	0.91	0.02	0.81	0.04	0.97	0.01	0.94	0.02	0.97	0.01	0.93	0.02
uWaveGLX	0.68	0.01	0.63	0.01	0.68	0.02	0.64	0.02	0.68	0.01	0.64	0.02	0.69	0.03	0.65	0.03
uWaveGLY	0.59	0.03	0.53	0.03	0.58	0.03	0.52	0.03	0.61	0.03	0.55	0.03	0.61	0.03	0.55	0.03
uWaveGLZ	0.58	0.02	0.52	0.02	0.58	0.02	0.52	0.02	0.60	0.02	0.54	0.03	0.59	0.02	0.53	0.03
Wafer	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.96	0.01	0.99	0.00	0.97	0.01
Yoga	0.82	0.01	0.64	0.02	0.82	0.02	0.64	0.04	0.84	0.01	0.69	0.02	0.84	0.03	0.67	0.05

Tabla B.1: Resultados del aprendizaje inductivo y transductivo con el método Self-training-1NN en el 10% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.97	0.02	0.96	0.02	0.97	0.02	0.95	0.03	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
Chlorine	0.72	0.01	0.54	0.02	0.73	0.01	0.56	0.01	0.68	0.01	0.48	0.02	0.68	0.02	0.48	0.03
CinCECGt	0.99	0.00	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.98	0.01
Coffee	0.91	0.06	0.82	0.12	0.95	0.05	0.90	0.09	0.86	0.08	0.72	0.15	0.93	0.04	0.86	0.08
CricketX	0.42	0.06	0.37	0.07	0.43	0.05	0.38	0.05	0.60	0.03	0.57	0.03	0.60	0.04	0.56	0.04
CricketY	0.40	0.04	0.35	0.04	0.44	0.06	0.39	0.07	0.54	0.02	0.49	0.02	0.55	0.04	0.51	0.05
CricketZ	0.44	0.01	0.39	0.02	0.46	0.05	0.41	0.05	0.62	0.03	0.58	0.03	0.64	0.02	0.60	0.02
ECG	0.99	0.00	0.98	0.01	0.99	0.00	0.98	0.01	0.99	0.00	0.98	0.01	0.99	0.00	0.98	0.01
ECGFiveD	0.97	0.01	0.94	0.02	0.97	0.01	0.93	0.02	0.94	0.02	0.89	0.03	0.95	0.02	0.89	0.05
FaceAll	0.86	0.02	0.85	0.02	0.87	0.03	0.85	0.03	0.96	0.01	0.95	0.01	0.96	0.01	0.95	0.02
FacesUCR	0.87	0.01	0.86	0.01	0.86	0.02	0.85	0.02	0.92	0.01	0.92	0.02	0.93	0.02	0.93	0.02
Fish	0.61	0.04	0.55	0.05	0.61	0.07	0.55	0.08	0.51	0.03	0.44	0.04	0.52	0.04	0.44	0.05
GunPoint	0.85	0.01	0.71	0.02	0.86	0.03	0.73	0.06	0.93	0.03	0.86	0.07	0.92	0.05	0.83	0.11
Haptics	0.35	0.03	0.19	0.04	0.31	0.06	0.13	0.07	0.38	0.06	0.22	0.07	0.36	0.04	0.19	0.05
InlineS	0.30	0.02	0.18	0.03	0.28	0.04	0.16	0.05	0.36	0.03	0.25	0.04	0.36	0.03	0.24	0.04
Italy	0.96	0.01	0.92	0.02	0.96	0.01	0.92	0.01	0.95	0.00	0.91	0.01	0.95	0.02	0.90	0.04
Lighting2	0.69	0.03	0.35	0.06	0.67	0.10	0.31	0.21	0.72	0.06	0.42	0.11	0.71	0.06	0.38	0.15
Lighting7	0.50	0.04	0.40	0.04	0.50	0.08	0.40	0.10	0.58	0.08	0.50	0.09	0.56	0.11	0.48	0.13
MALLAT	0.94	0.01	0.93	0.01	0.95	0.01	0.94	0.01	0.94	0.00	0.93	0.00	0.95	0.01	0.94	0.01
MedicalI	0.61	0.02	0.46	0.02	0.60	0.02	0.43	0.04	0.64	0.03	0.49	0.03	0.63	0.03	0.48	0.04
MoteStrain	0.90	0.01	0.80	0.02	0.90	0.02	0.79	0.04	0.95	0.01	0.90	0.02	0.96	0.01	0.92	0.02
Olive	0.73	0.08	0.62	0.11	0.75	0.06	0.65	0.08	0.73	0.08	0.62	0.11	0.73	0.07	0.63	0.09
OSULeaf	0.52	0.04	0.42	0.05	0.51	0.07	0.40	0.08	0.54	0.03	0.44	0.04	0.55	0.03	0.46	0.04
Sony	0.95	0.02	0.90	0.03	0.95	0.03	0.91	0.05	0.94	0.01	0.87	0.03	0.94	0.04	0.87	0.08
SonyII	0.94	0.01	0.88	0.03	0.94	0.03	0.87	0.06	0.93	0.01	0.86	0.02	0.94	0.02	0.87	0.04
StarLightC	0.86	0.01	0.77	0.01	0.86	0.01	0.77	0.02	0.88	0.01	0.81	0.01	0.89	0.01	0.81	0.02
Synthetic	0.75	0.03	0.69	0.03	0.80	0.04	0.76	0.05	0.77	0.04	0.72	0.04	0.81	0.04	0.77	0.05
Trace	0.62	0.07	0.50	0.09	0.65	0.06	0.53	0.08	0.86	0.05	0.81	0.07	0.88	0.06	0.83	0.08
Two	0.87	0.00	0.83	0.01	0.87	0.01	0.83	0.01	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
TwoLeadE	0.96	0.01	0.92	0.01	0.97	0.01	0.94	0.03	0.98	0.01	0.97	0.02	0.99	0.01	0.97	0.02
uWaveGLX	0.68	0.02	0.64	0.02	0.69	0.02	0.64	0.02	0.71	0.01	0.67	0.01	0.72	0.02	0.68	0.03
uWaveGLY	0.63	0.01	0.58	0.01	0.63	0.01	0.57	0.01	0.65	0.01	0.60	0.01	0.65	0.02	0.59	0.03
uWaveGLZ	0.61	0.02	0.55	0.02	0.60	0.03	0.54	0.04	0.64	0.01	0.59	0.01	0.63	0.02	0.58	0.02
Wafer	1.00	0.00	0.97	0.00	1.00	0.00	0.98	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.02
Yoga	0.87	0.01	0.74	0.02	0.87	0.01	0.73	0.02	0.88	0.01	0.77	0.02	0.89	0.01	0.78	0.02

Tabla B.2: Resultados del aprendizaje inductivo y transductivo con el método Self-training-1NN en el 20% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.98	0.01	0.97	0.01	0.98	0.02	0.97	0.02	1.00	0.00	0.99	0.00	1.00	0.00	1.00	0.00
Chlorine	0.81	0.01	0.69	0.02	0.82	0.01	0.71	0.02	0.77	0.01	0.62	0.02	0.78	0.01	0.64	0.01
CinCECGt	0.99	0.00	0.99	0.00	1.00	0.01	0.99	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.01
Coffee	0.95	0.04	0.90	0.07	0.97	0.05	0.93	0.10	0.92	0.03	0.84	0.06	0.97	0.05	0.93	0.10
CricketX	0.49	0.02	0.45	0.02	0.49	0.04	0.44	0.04	0.66	0.02	0.63	0.02	0.66	0.03	0.62	0.03
CricketY	0.44	0.04	0.39	0.04	0.46	0.02	0.41	0.03	0.58	0.02	0.54	0.03	0.59	0.02	0.55	0.03
CricketZ	0.50	0.02	0.46	0.02	0.49	0.02	0.45	0.03	0.66	0.03	0.62	0.04	0.67	0.02	0.64	0.02
ECG	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.00
ECGFiveD	0.97	0.01	0.94	0.01	0.96	0.02	0.93	0.03	0.97	0.01	0.94	0.02	0.97	0.01	0.93	0.02
FaceAll	0.90	0.01	0.89	0.01	0.89	0.02	0.88	0.02	0.96	0.01	0.96	0.01	0.97	0.01	0.97	0.01
FacesUCR	0.89	0.02	0.88	0.02	0.89	0.02	0.88	0.02	0.96	0.01	0.95	0.01	0.96	0.01	0.95	0.01
Fish	0.69	0.05	0.64	0.06	0.70	0.05	0.65	0.06	0.53	0.01	0.46	0.01	0.62	0.05	0.56	0.06
GunPoint	0.90	0.05	0.79	0.09	0.90	0.03	0.80	0.06	0.95	0.03	0.90	0.05	0.96	0.03	0.93	0.06
Haptics	0.36	0.03	0.20	0.04	0.38	0.06	0.22	0.07	0.38	0.04	0.23	0.06	0.43	0.06	0.28	0.08
InlineS	0.37	0.03	0.26	0.03	0.37	0.04	0.25	0.04	0.41	0.02	0.30	0.02	0.39	0.03	0.28	0.04
Italy	0.96	0.01	0.92	0.02	0.96	0.01	0.92	0.02	0.95	0.01	0.90	0.02	0.95	0.01	0.91	0.02
Lighting2	0.68	0.09	0.35	0.16	0.69	0.08	0.38	0.14	0.79	0.04	0.57	0.08	0.80	0.09	0.59	0.21
Lighting7	0.54	0.07	0.44	0.09	0.53	0.05	0.43	0.06	0.64	0.05	0.57	0.05	0.66	0.02	0.59	0.03
MALLAT	0.94	0.01	0.93	0.01	0.96	0.01	0.95	0.01	0.94	0.01	0.94	0.01	0.95	0.01	0.94	0.01
MedicalI	0.67	0.01	0.54	0.02	0.67	0.03	0.53	0.04	0.69	0.03	0.56	0.04	0.69	0.05	0.56	0.08
MoteStrain	0.89	0.01	0.78	0.02	0.89	0.03	0.79	0.07	0.96	0.01	0.92	0.01	0.96	0.02	0.92	0.04
Olive	0.70	0.07	0.59	0.08	0.83	0.06	0.77	0.08	0.73	0.06	0.63	0.07	0.78	0.05	0.70	0.06
OSULeaf	0.54	0.03	0.44	0.03	0.55	0.04	0.46	0.05	0.58	0.04	0.49	0.05	0.59	0.03	0.50	0.04
Sony	0.96	0.01	0.92	0.02	0.96	0.02	0.93	0.05	0.95	0.02	0.89	0.03	0.94	0.03	0.88	0.06
SonyII	0.96	0.01	0.92	0.02	0.96	0.02	0.92	0.04	0.95	0.01	0.90	0.02	0.96	0.02	0.91	0.04
StarLightC	0.86	0.01	0.77	0.01	0.86	0.01	0.78	0.02	0.88	0.00	0.80	0.01	0.89	0.01	0.81	0.01
Synthetic	0.79	0.05	0.74	0.06	0.84	0.06	0.81	0.07	0.82	0.09	0.78	0.11	0.88	0.06	0.85	0.07
Trace	0.68	0.04	0.58	0.05	0.66	0.05	0.55	0.07	0.90	0.02	0.86	0.03	0.92	0.02	0.90	0.02
Two	0.92	0.00	0.90	0.00	0.92	0.01	0.90	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
TwoLeadE	0.98	0.01	0.96	0.01	0.98	0.01	0.97	0.01	0.99	0.01	0.98	0.01	0.99	0.01	0.98	0.01
uWaveGLX	0.70	0.01	0.66	0.01	0.72	0.01	0.68	0.02	0.71	0.01	0.67	0.01	0.73	0.01	0.69	0.02
uWaveGLY	0.63	0.01	0.58	0.01	0.65	0.01	0.60	0.01	0.65	0.01	0.60	0.02	0.65	0.03	0.60	0.03
uWaveGLZ	0.62	0.01	0.57	0.01	0.63	0.02	0.58	0.02	0.65	0.02	0.60	0.02	0.67	0.01	0.62	0.01
Wafer	1.00	0.00	0.98	0.00	1.00	0.00	0.98	0.01	1.00	0.00	0.98	0.01	1.00	0.00	0.98	0.01
Yoga	0.89	0.01	0.79	0.01	0.89	0.01	0.78	0.02	0.91	0.01	0.81	0.02	0.90	0.01	0.80	0.02

Tabla B.3: Resultados del aprendizaje inductivo y transductivo con el método Self-training-1NN en el 30% de ejemplos etiquetados.



Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.95	0.00	0.92	0.01	0.94	0.02	0.91	0.03	0.99	0.00	0.98	0.01	0.98	0.02	0.97	0.03
Chlorine	0.60	0.01	0.34	0.01	0.59	0.02	0.33	0.03	0.57	0.01	0.30	0.00	0.57	0.02	0.30	0.02
CinCECGt	0.98	0.01	0.97	0.01	0.98	0.01	0.97	0.02	0.98	0.01	0.98	0.01	0.98	0.01	0.98	0.02
Coffee	0.91	0.09	0.81	0.18	0.93	0.04	0.86	0.08	0.87	0.05	0.74	0.11	0.91	0.07	0.82	0.14
CricketX	0.34	0.03	0.28	0.04	0.36	0.04	0.30	0.05	0.50	0.04	0.46	0.04	0.50	0.04	0.45	0.05
CricketY	0.34	0.03	0.28	0.03	0.36	0.02	0.30	0.02	0.46	0.06	0.42	0.06	0.48	0.04	0.43	0.04
CricketZ	0.35	0.04	0.29	0.04	0.37	0.03	0.31	0.03	0.52	0.04	0.47	0.04	0.53	0.04	0.49	0.05
ECG	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.96	0.01	0.98	0.01	0.95	0.03
ECGFiveD	0.94	0.02	0.88	0.04	0.94	0.03	0.87	0.06	0.89	0.03	0.78	0.06	0.90	0.01	0.81	0.02
FaceAll	0.82	0.03	0.80	0.03	0.82	0.02	0.80	0.02	0.92	0.02	0.91	0.03	0.93	0.02	0.92	0.02
FacesUCR	0.81	0.04	0.79	0.04	0.81	0.03	0.79	0.03	0.92	0.02	0.91	0.02	0.92	0.02	0.92	0.02
Fish	0.62	0.05	0.56	0.06	0.63	0.05	0.56	0.06	0.63	0.07	0.57	0.08	0.63	0.11	0.57	0.12
GunPoint	0.71	0.08	0.43	0.16	0.72	0.08	0.43	0.17	0.86	0.06	0.72	0.12	0.86	0.09	0.71	0.18
Haptics	0.37	0.03	0.20	0.03	0.38	0.03	0.22	0.03	0.34	0.03	0.17	0.03	0.35	0.01	0.18	0.02
InlineS	0.25	0.02	0.12	0.02	0.28	0.02	0.15	0.02	0.29	0.04	0.17	0.05	0.30	0.03	0.19	0.04
Italy	0.96	0.01	0.92	0.01	0.96	0.01	0.92	0.03	0.94	0.00	0.88	0.01	0.94	0.01	0.87	0.02
Lighting2	0.62	0.04	0.19	0.13	0.62	0.14	0.19	0.27	0.72	0.07	0.38	0.16	0.73	0.08	0.41	0.21
Lighting7	0.46	0.02	0.36	0.02	0.47	0.06	0.37	0.08	0.53	0.07	0.44	0.08	0.59	0.06	0.51	0.06
MALLAT	0.94	0.00	0.94	0.00	0.94	0.01	0.93	0.01	0.96	0.01	0.96	0.01	0.97	0.01	0.96	0.01
MedicalI	0.53	0.03	0.38	0.04	0.53	0.04	0.39	0.04	0.57	0.02	0.43	0.03	0.56	0.04	0.42	0.04
MoteStrain	0.91	0.01	0.81	0.01	0.90	0.02	0.79	0.05	0.94	0.01	0.89	0.02	0.94	0.01	0.88	0.02
Olive	0.84	0.06	0.77	0.08	0.82	0.04	0.74	0.06	0.85	0.05	0.78	0.08	0.80	0.07	0.72	0.11
OSULeaf	0.44	0.05	0.32	0.06	0.41	0.06	0.29	0.07	0.51	0.04	0.41	0.04	0.48	0.05	0.37	0.06
Sony	0.92	0.04	0.85	0.07	0.92	0.05	0.85	0.10	0.91	0.04	0.83	0.08	0.91	0.05	0.83	0.09
SonyII	0.92	0.02	0.82	0.05	0.91	0.02	0.81	0.05	0.90	0.02	0.79	0.05	0.91	0.02	0.80	0.04
StarLightC	0.85	0.01	0.75	0.02	0.84	0.02	0.75	0.03	0.87	0.01	0.79	0.01	0.87	0.01	0.79	0.02
Synthetic	0.83	0.05	0.79	0.07	0.83	0.06	0.79	0.08	0.91	0.05	0.90	0.06	0.91	0.06	0.89	0.07
Trace	0.59	0.06	0.45	0.08	0.56	0.05	0.41	0.06	0.79	0.04	0.72	0.06	0.76	0.10	0.69	0.14
Two	0.77	0.01	0.69	0.01	0.77	0.02	0.70	0.02	0.97	0.00	0.96	0.01	0.97	0.00	0.96	0.01
TwoLeadE	0.91	0.02	0.82	0.03	0.91	0.02	0.82	0.03	0.97	0.01	0.94	0.02	0.97	0.01	0.94	0.02
uWaveGLX	0.69	0.01	0.65	0.01	0.70	0.01	0.65	0.01	0.70	0.01	0.66	0.01	0.71	0.01	0.67	0.02
uWaveGLY	0.62	0.02	0.56	0.02	0.61	0.02	0.55	0.02	0.63	0.02	0.58	0.03	0.63	0.02	0.57	0.03
uWaveGLZ	0.60	0.01	0.54	0.01	0.59	0.01	0.54	0.01	0.62	0.02	0.57	0.02	0.61	0.02	0.55	0.02
Wafer	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.96	0.01	0.99	0.00	0.97	0.01
Yoga	0.82	0.01	0.64	0.02	0.82	0.02	0.64	0.04	0.85	0.01	0.69	0.02	0.84	0.02	0.68	0.04

Tabla B.4: Resultados del aprendizaje inductivo y transductivo con el método SETRED en el 10 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.97	0.02	0.96	0.02	0.97	0.02	0.95	0.02	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
Chlorine	0.72	0.01	0.53	0.02	0.72	0.01	0.55	0.01	0.68	0.01	0.47	0.02	0.68	0.01	0.47	0.02
CinCECGt	0.99	0.00	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.98	0.01
Coffee	0.93	0.03	0.85	0.06	0.93	0.07	0.86	0.14	0.90	0.04	0.80	0.07	0.88	0.09	0.76	0.19
CricketX	0.46	0.04	0.41	0.04	0.45	0.06	0.40	0.06	0.62	0.02	0.58	0.03	0.60	0.03	0.56	0.04
CricketY	0.43	0.04	0.38	0.05	0.44	0.06	0.38	0.07	0.58	0.01	0.54	0.01	0.59	0.05	0.56	0.05
CricketZ	0.46	0.02	0.41	0.03	0.47	0.03	0.43	0.03	0.63	0.03	0.60	0.03	0.64	0.02	0.61	0.02
ECG	0.99	0.00	0.98	0.01	0.99	0.00	0.98	0.01	0.99	0.00	0.98	0.01	0.99	0.00	0.98	0.01
ECGFiveD	0.97	0.01	0.93	0.02	0.97	0.01	0.93	0.02	0.94	0.02	0.89	0.03	0.95	0.02	0.89	0.05
FaceAll	0.87	0.01	0.86	0.01	0.87	0.02	0.86	0.02	0.96	0.01	0.96	0.01	0.96	0.02	0.96	0.02
FacesUCR	0.87	0.01	0.86	0.01	0.87	0.01	0.86	0.01	0.95	0.01	0.95	0.01	0.96	0.01	0.95	0.01
Fish	0.66	0.07	0.61	0.08	0.68	0.05	0.63	0.06	0.68	0.05	0.62	0.06	0.67	0.06	0.61	0.06
GunPoint	0.82	0.03	0.64	0.06	0.82	0.04	0.65	0.08	0.92	0.04	0.85	0.08	0.90	0.05	0.80	0.11
Haptics	0.37	0.04	0.22	0.05	0.37	0.06	0.21	0.08	0.41	0.04	0.26	0.06	0.39	0.04	0.24	0.05
InlineS	0.31	0.02	0.19	0.03	0.31	0.02	0.19	0.03	0.38	0.02	0.27	0.02	0.38	0.04	0.27	0.04
Italy	0.96	0.01	0.92	0.02	0.96	0.01	0.92	0.01	0.95	0.00	0.91	0.01	0.95	0.02	0.90	0.04
Lighting2	0.66	0.03	0.29	0.06	0.66	0.12	0.28	0.27	0.72	0.04	0.41	0.06	0.70	0.07	0.37	0.17
Lighting7	0.54	0.06	0.44	0.07	0.55	0.07	0.46	0.08	0.67	0.09	0.60	0.11	0.71	0.07	0.65	0.08
MALLAT	0.96	0.01	0.96	0.01	0.97	0.01	0.96	0.01	0.98	0.00	0.97	0.00	0.98	0.00	0.98	0.00
MedicalI	0.62	0.03	0.48	0.03	0.61	0.03	0.45	0.04	0.67	0.02	0.54	0.02	0.65	0.02	0.52	0.02
MoteStrain	0.90	0.01	0.80	0.02	0.90	0.02	0.79	0.04	0.95	0.01	0.90	0.02	0.96	0.01	0.92	0.01
Olive	0.81	0.05	0.72	0.07	0.82	0.09	0.74	0.12	0.82	0.06	0.73	0.08	0.85	0.11	0.79	0.15
OSULeaf	0.51	0.05	0.40	0.06	0.48	0.05	0.37	0.06	0.55	0.05	0.45	0.06	0.55	0.05	0.46	0.06
Sony	0.95	0.02	0.90	0.03	0.95	0.03	0.91	0.05	0.93	0.01	0.86	0.03	0.94	0.04	0.87	0.07
SonyII	0.94	0.01	0.88	0.03	0.94	0.03	0.86	0.06	0.94	0.01	0.86	0.02	0.94	0.02	0.87	0.04
StarLightC	0.86	0.01	0.77	0.01	0.86	0.01	0.77	0.02	0.88	0.00	0.81	0.01	0.89	0.01	0.81	0.01
Synthetic	0.86	0.03	0.84	0.04	0.88	0.03	0.85	0.04	0.96	0.01	0.95	0.02	0.96	0.01	0.96	0.01
Trace	0.59	0.06	0.45	0.08	0.66	0.10	0.54	0.13	0.89	0.03	0.85	0.04	0.90	0.06	0.87	0.08
Two	0.87	0.00	0.83	0.00	0.87	0.01	0.83	0.01	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
TwoLeadE	0.96	0.01	0.92	0.01	0.97	0.01	0.94	0.03	0.98	0.01	0.97	0.02	0.99	0.01	0.97	0.02
uWaveGLX	0.71	0.00	0.67	0.00	0.72	0.01	0.67	0.02	0.74	0.01	0.70	0.01	0.74	0.02	0.70	0.02
uWaveGLY	0.65	0.01	0.60	0.01	0.65	0.02	0.59	0.02	0.67	0.01	0.62	0.01	0.66	0.02	0.61	0.02
uWaveGLZ	0.64	0.01	0.59	0.01	0.63	0.03	0.58	0.03	0.67	0.01	0.62	0.01	0.66	0.02	0.61	0.02
Wafer	1.00	0.00	0.97	0.00	1.00	0.00	0.98	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.02
Yoga	0.87	0.01	0.74	0.03	0.87	0.01	0.73	0.02	0.88	0.01	0.77	0.02	0.89	0.01	0.78	0.02

Tabla B.5: Resultados del aprendizaje inductivo y transductivo con el método SETRED en el 20% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.98	0.01	0.97	0.01	0.98	0.02	0.97	0.02	1.00	0.00	0.99	0.00	1.00	0.00	1.00	0.00
Chlorine	0.81	0.01	0.68	0.02	0.82	0.01	0.70	0.02	0.77	0.01	0.62	0.02	0.78	0.01	0.63	0.02
CinCECGt	0.99	0.00	0.99	0.00	1.00	0.01	0.99	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.01
Coffee	0.96	0.03	0.92	0.05	0.97	0.07	0.93	0.15	0.93	0.05	0.87	0.09	0.95	0.05	0.90	0.09
CricketX	0.53	0.02	0.49	0.02	0.51	0.02	0.47	0.02	0.68	0.02	0.65	0.02	0.68	0.04	0.65	0.04
CricketY	0.48	0.05	0.43	0.05	0.47	0.03	0.42	0.03	0.64	0.03	0.60	0.03	0.63	0.03	0.60	0.03
CricketZ	0.54	0.02	0.50	0.03	0.55	0.01	0.50	0.01	0.69	0.02	0.66	0.02	0.70	0.01	0.67	0.02
ECG	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.00
ECGFiveD	0.97	0.01	0.94	0.01	0.96	0.02	0.93	0.03	0.97	0.01	0.94	0.02	0.97	0.01	0.93	0.02
FaceAll	0.90	0.01	0.89	0.01	0.90	0.02	0.89	0.02	0.97	0.01	0.96	0.01	0.97	0.01	0.97	0.01
FacesUCR	0.90	0.01	0.89	0.01	0.90	0.02	0.89	0.02	0.97	0.00	0.97	0.01	0.97	0.01	0.97	0.01
Fish	0.76	0.06	0.72	0.07	0.76	0.04	0.72	0.05	0.74	0.04	0.70	0.05	0.77	0.05	0.73	0.05
GunPoint	0.88	0.04	0.77	0.07	0.89	0.06	0.78	0.12	0.94	0.01	0.89	0.02	0.96	0.02	0.92	0.04
Haptics	0.39	0.01	0.24	0.02	0.38	0.05	0.22	0.06	0.43	0.03	0.28	0.04	0.42	0.06	0.27	0.07
InlineS	0.37	0.04	0.26	0.04	0.37	0.05	0.26	0.05	0.43	0.01	0.33	0.01	0.43	0.04	0.33	0.05
Italy	0.96	0.01	0.92	0.02	0.96	0.01	0.92	0.02	0.95	0.01	0.90	0.02	0.95	0.01	0.91	0.02
Lighting2	0.69	0.10	0.37	0.18	0.66	0.09	0.32	0.16	0.81	0.04	0.59	0.07	0.80	0.08	0.58	0.19
Lighting7	0.57	0.05	0.48	0.06	0.56	0.03	0.46	0.04	0.70	0.03	0.64	0.04	0.70	0.06	0.64	0.07
MALLAT	0.97	0.01	0.96	0.01	0.97	0.01	0.96	0.01	0.98	0.00	0.98	0.00	0.98	0.01	0.98	0.01
MedicalI	0.68	0.02	0.55	0.02	0.68	0.03	0.56	0.03	0.71	0.03	0.60	0.04	0.69	0.04	0.57	0.07
MoteStrain	0.89	0.01	0.78	0.02	0.89	0.03	0.78	0.07	0.96	0.01	0.92	0.01	0.96	0.02	0.92	0.04
Olive	0.83	0.06	0.75	0.09	0.88	0.07	0.83	0.11	0.81	0.06	0.72	0.09	0.87	0.07	0.81	0.11
OSULeaf	0.54	0.04	0.44	0.05	0.55	0.04	0.46	0.05	0.61	0.02	0.53	0.02	0.60	0.06	0.52	0.08
Sony	0.96	0.01	0.92	0.02	0.96	0.02	0.93	0.04	0.95	0.02	0.89	0.03	0.95	0.02	0.90	0.05
SonyII	0.97	0.01	0.93	0.02	0.96	0.02	0.91	0.04	0.95	0.01	0.90	0.02	0.96	0.02	0.91	0.04
StarLightC	0.86	0.01	0.78	0.01	0.86	0.01	0.78	0.02	0.89	0.01	0.82	0.01	0.89	0.01	0.83	0.01
Synthetic	0.90	0.03	0.88	0.03	0.89	0.03	0.87	0.04	0.96	0.01	0.95	0.02	0.96	0.01	0.95	0.02
Trace	0.66	0.05	0.54	0.07	0.64	0.05	0.52	0.07	0.92	0.03	0.90	0.04	0.92	0.02	0.90	0.03
Two	0.92	0.00	0.89	0.00	0.92	0.00	0.90	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
TwoLeadE	0.98	0.01	0.96	0.01	0.98	0.01	0.96	0.01	0.99	0.00	0.98	0.01	0.99	0.01	0.98	0.01
uWaveGLX	0.74	0.01	0.70	0.01	0.74	0.01	0.70	0.01	0.76	0.01	0.73	0.01	0.77	0.01	0.73	0.01
uWaveGLY	0.66	0.01	0.61	0.01	0.67	0.01	0.62	0.01	0.69	0.01	0.64	0.01	0.69	0.02	0.64	0.03
uWaveGLZ	0.66	0.01	0.61	0.01	0.65	0.01	0.61	0.01	0.69	0.01	0.64	0.01	0.69	0.01	0.65	0.01
Wafer	1.00	0.00	0.98	0.00	1.00	0.00	0.98	0.01	1.00	0.00	0.98	0.01	1.00	0.00	0.98	0.01
Yoga	0.89	0.01	0.78	0.02	0.89	0.01	0.78	0.02	0.91	0.01	0.81	0.02	0.90	0.01	0.80	0.02

Tabla B.6: Resultados del aprendizaje inductivo y transductivo con el método SETRED en el 30% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.89	0.02	0.83	0.03	0.91	0.03	0.87	0.05	0.92	0.02	0.89	0.03	0.95	0.03	0.92	0.05
Chlorine	0.59	0.03	0.32	0.03	0.59	0.03	0.32	0.04	0.58	0.01	0.30	0.01	0.58	0.01	0.30	0.02
CinCECGt	0.94	0.00	0.92	0.00	0.94	0.01	0.93	0.01	0.95	0.01	0.93	0.02	0.95	0.01	0.94	0.01
Coffee	0.90	0.07	0.81	0.14	0.95	0.05	0.89	0.10	0.86	0.08	0.73	0.15	0.91	0.09	0.82	0.18
CricketX	0.32	0.02	0.26	0.02	0.33	0.04	0.27	0.05	0.48	0.02	0.43	0.03	0.48	0.04	0.43	0.04
CricketY	0.35	0.03	0.29	0.03	0.36	0.04	0.30	0.04	0.47	0.03	0.42	0.03	0.46	0.04	0.42	0.04
CricketZ	0.32	0.02	0.26	0.02	0.34	0.03	0.28	0.03	0.48	0.02	0.44	0.02	0.51	0.03	0.47	0.03
ECG	0.96	0.00	0.88	0.01	0.96	0.00	0.90	0.01	0.95	0.00	0.86	0.01	0.96	0.01	0.90	0.02
ECGFiveD	0.87	0.02	0.74	0.04	0.88	0.03	0.75	0.06	0.86	0.04	0.71	0.08	0.89	0.03	0.79	0.05
FaceAll	0.76	0.01	0.74	0.01	0.76	0.02	0.73	0.02	0.90	0.01	0.89	0.01	0.91	0.03	0.90	0.03
FacesUCR	0.75	0.03	0.72	0.04	0.75	0.02	0.72	0.02	0.90	0.01	0.89	0.01	0.91	0.01	0.90	0.01
Fish	0.57	0.06	0.50	0.07	0.60	0.09	0.54	0.10	0.60	0.04	0.54	0.05	0.63	0.09	0.57	0.11
GunPoint	0.71	0.03	0.42	0.06	0.72	0.05	0.44	0.11	0.82	0.05	0.64	0.10	0.86	0.04	0.71	0.09
Haptics	0.37	0.04	0.21	0.05	0.35	0.06	0.19	0.08	0.36	0.04	0.20	0.05	0.37	0.04	0.20	0.05
InlineS	0.26	0.01	0.13	0.02	0.27	0.03	0.15	0.03	0.31	0.03	0.19	0.04	0.35	0.04	0.24	0.05
Italy	0.95	0.01	0.91	0.02	0.95	0.01	0.89	0.03	0.91	0.02	0.83	0.04	0.95	0.01	0.90	0.02
Lighting2	0.62	0.05	0.18	0.14	0.59	0.10	0.13	0.21	0.68	0.08	0.29	0.19	0.68	0.09	0.27	0.21
Lighting7	0.44	0.04	0.34	0.05	0.44	0.07	0.33	0.08	0.52	0.02	0.44	0.02	0.55	0.07	0.47	0.08
MALLAT	0.93	0.01	0.92	0.01	0.94	0.00	0.93	0.01	0.95	0.01	0.94	0.01	0.96	0.01	0.95	0.01
MedicalI	0.60	0.02	0.42	0.03	0.61	0.04	0.43	0.06	0.62	0.02	0.46	0.02	0.63	0.04	0.47	0.04
MoteStrain	0.88	0.00	0.75	0.01	0.87	0.02	0.73	0.03	0.90	0.02	0.81	0.03	0.92	0.02	0.83	0.04
Olive	0.78	0.11	0.68	0.15	0.72	0.14	0.60	0.19	0.84	0.04	0.76	0.06	0.80	0.07	0.71	0.11
OSULeaf	0.44	0.03	0.31	0.04	0.42	0.05	0.29	0.06	0.48	0.04	0.37	0.05	0.47	0.07	0.35	0.08
Sony	0.91	0.02	0.82	0.03	0.93	0.05	0.86	0.09	0.90	0.02	0.80	0.04	0.90	0.05	0.80	0.09
SonyII	0.88	0.03	0.74	0.06	0.91	0.03	0.80	0.07	0.89	0.01	0.75	0.03	0.91	0.02	0.80	0.04
StarLightC	0.86	0.01	0.77	0.01	0.86	0.01	0.77	0.02	0.89	0.00	0.82	0.01	0.88	0.01	0.81	0.02
Synthetic	0.79	0.04	0.74	0.05	0.79	0.05	0.75	0.05	0.90	0.03	0.89	0.04	0.91	0.03	0.89	0.04
Trace	0.56	0.04	0.42	0.05	0.53	0.05	0.37	0.06	0.70	0.10	0.60	0.14	0.69	0.11	0.59	0.14
Two	0.76	0.01	0.67	0.01	0.76	0.03	0.68	0.03	0.95	0.01	0.93	0.01	0.95	0.00	0.94	0.00
TwoLeadE	0.86	0.03	0.71	0.06	0.87	0.03	0.74	0.07	0.91	0.02	0.83	0.04	0.93	0.02	0.85	0.03
uWaveGLX	0.70	0.01	0.66	0.01	0.70	0.01	0.65	0.01	0.71	0.01	0.67	0.01	0.72	0.02	0.68	0.02
uWaveGLY	0.64	0.01	0.59	0.01	0.64	0.02	0.58	0.02	0.66	0.01	0.61	0.01	0.66	0.02	0.61	0.02
uWaveGLZ	0.62	0.01	0.56	0.01	0.61	0.01	0.55	0.01	0.63	0.01	0.58	0.01	0.63	0.02	0.58	0.02
Wafer	0.98	0.00	0.92	0.01	0.99	0.00	0.92	0.02	0.99	0.00	0.93	0.01	0.99	0.00	0.95	0.01
Yoga	0.80	0.01	0.59	0.02	0.80	0.02	0.60	0.04	0.83	0.01	0.66	0.01	0.83	0.02	0.67	0.03

Tabla B.7: Resultados del aprendizaje inductivo y transductivo con el método SNNRCE en el 10 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.92	0.02	0.88	0.03	0.93	0.04	0.90	0.06	0.95	0.00	0.92	0.01	0.98	0.01	0.97	0.02
Chlorine	0.71	0.02	0.51	0.03	0.72	0.02	0.54	0.04	0.69	0.01	0.48	0.02	0.70	0.01	0.50	0.03
CinCECGt	0.95	0.01	0.94	0.02	0.96	0.01	0.95	0.02	0.95	0.02	0.94	0.02	0.95	0.02	0.94	0.02
Coffee	0.83	0.05	0.65	0.10	0.84	0.09	0.67	0.18	0.83	0.09	0.65	0.19	0.86	0.05	0.71	0.11
CricketX	0.41	0.03	0.36	0.03	0.43	0.05	0.38	0.05	0.60	0.03	0.56	0.03	0.58	0.03	0.55	0.03
CricketY	0.42	0.04	0.37	0.05	0.44	0.04	0.39	0.05	0.58	0.02	0.54	0.02	0.58	0.03	0.54	0.03
CricketZ	0.43	0.03	0.38	0.03	0.43	0.05	0.38	0.05	0.60	0.02	0.57	0.02	0.61	0.02	0.58	0.02
ECG	0.97	0.00	0.91	0.01	0.97	0.00	0.93	0.01	0.96	0.01	0.89	0.02	0.97	0.01	0.92	0.02
ECGFiveD	0.91	0.03	0.83	0.06	0.92	0.03	0.84	0.06	0.91	0.02	0.83	0.03	0.93	0.03	0.86	0.06
FaceAll	0.84	0.01	0.82	0.01	0.84	0.03	0.83	0.03	0.93	0.01	0.92	0.01	0.94	0.01	0.94	0.02
FacesUCR	0.83	0.02	0.81	0.02	0.83	0.01	0.81	0.01	0.92	0.01	0.92	0.01	0.94	0.01	0.93	0.01
Fish	0.66	0.03	0.60	0.03	0.68	0.05	0.63	0.06	0.71	0.03	0.66	0.03	0.71	0.02	0.66	0.02
GunPoint	0.82	0.04	0.64	0.08	0.82	0.04	0.63	0.08	0.90	0.03	0.80	0.06	0.88	0.05	0.77	0.10
Haptics	0.35	0.06	0.19	0.07	0.34	0.06	0.17	0.08	0.39	0.05	0.24	0.07	0.39	0.05	0.24	0.06
InlineS	0.30	0.01	0.18	0.02	0.32	0.04	0.20	0.05	0.38	0.03	0.27	0.03	0.40	0.03	0.29	0.03
Italy	0.95	0.01	0.90	0.02	0.95	0.01	0.89	0.01	0.93	0.01	0.87	0.02	0.94	0.01	0.88	0.02
Lighting2	0.67	0.05	0.31	0.08	0.66	0.12	0.30	0.26	0.71	0.07	0.38	0.12	0.72	0.07	0.40	0.14
Lighting7	0.52	0.04	0.41	0.05	0.51	0.03	0.41	0.03	0.67	0.07	0.60	0.09	0.68	0.04	0.61	0.05
MALLAT	0.95	0.01	0.94	0.01	0.95	0.01	0.95	0.01	0.95	0.01	0.94	0.01	0.96	0.01	0.95	0.01
MedicalI	0.65	0.01	0.49	0.02	0.64	0.02	0.46	0.03	0.70	0.01	0.56	0.02	0.67	0.02	0.52	0.02
MoteStrain	0.87	0.01	0.74	0.02	0.88	0.02	0.76	0.04	0.93	0.01	0.85	0.02	0.95	0.01	0.89	0.03
Olive	0.78	0.08	0.68	0.12	0.83	0.00	0.76	0.01	0.77	0.09	0.67	0.14	0.80	0.10	0.72	0.13
OSULeaf	0.51	0.05	0.40	0.06	0.50	0.08	0.39	0.10	0.54	0.04	0.44	0.05	0.55	0.07	0.45	0.09
Sony	0.92	0.03	0.84	0.07	0.93	0.03	0.85	0.07	0.91	0.02	0.81	0.03	0.93	0.04	0.86	0.07
SonyII	0.90	0.01	0.79	0.02	0.92	0.01	0.83	0.02	0.91	0.01	0.80	0.03	0.91	0.03	0.80	0.07
StarLightC	0.87	0.00	0.79	0.00	0.87	0.01	0.79	0.02	0.90	0.01	0.83	0.01	0.89	0.01	0.83	0.02
Synthetic	0.85	0.01	0.81	0.02	0.85	0.04	0.82	0.04	0.92	0.02	0.91	0.03	0.94	0.02	0.93	0.03
Trace	0.57	0.07	0.42	0.09	0.61	0.11	0.48	0.14	0.82	0.04	0.76	0.06	0.85	0.06	0.80	0.08
Two	0.87	0.00	0.83	0.00	0.87	0.01	0.83	0.01	0.98	0.00	0.97	0.00	0.98	0.01	0.98	0.01
TwoLeadE	0.91	0.01	0.81	0.03	0.93	0.02	0.85	0.04	0.92	0.01	0.85	0.02	0.94	0.03	0.87	0.06
uWaveGLX	0.71	0.00	0.67	0.01	0.72	0.01	0.68	0.01	0.74	0.00	0.71	0.00	0.75	0.01	0.72	0.02
uWaveGLY	0.67	0.01	0.62	0.01	0.65	0.01	0.60	0.01	0.69	0.01	0.64	0.01	0.68	0.01	0.63	0.01
uWaveGLZ	0.64	0.01	0.59	0.01	0.64	0.02	0.59	0.02	0.67	0.01	0.63	0.01	0.66	0.02	0.61	0.02
Wafer	0.99	0.00	0.94	0.01	0.99	0.00	0.95	0.02	0.99	0.00	0.94	0.01	0.99	0.00	0.95	0.02
Yoga	0.85	0.01	0.70	0.02	0.86	0.01	0.72	0.03	0.87	0.02	0.73	0.03	0.88	0.01	0.75	0.03

Tabla B.8: Resultados del aprendizaje inductivo y transductivo con el método SNNRCE en el 20 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.92	0.02	0.88	0.03	0.95	0.01	0.93	0.01	0.95	0.01	0.93	0.01	0.97	0.01	0.96	0.02
Chlorine	0.81	0.02	0.68	0.03	0.82	0.01	0.70	0.01	0.77	0.01	0.62	0.02	0.79	0.01	0.65	0.02
CinCECGt	0.97	0.01	0.96	0.01	0.98	0.01	0.97	0.02	0.96	0.01	0.95	0.01	0.98	0.01	0.97	0.01
Coffee	0.92	0.06	0.83	0.12	0.95	0.11	0.90	0.22	0.89	0.04	0.78	0.08	0.91	0.06	0.83	0.12
CricketX	0.50	0.04	0.45	0.04	0.50	0.03	0.46	0.04	0.67	0.02	0.64	0.02	0.66	0.04	0.63	0.04
CricketY	0.45	0.04	0.40	0.04	0.46	0.03	0.41	0.03	0.62	0.05	0.59	0.05	0.64	0.02	0.60	0.02
CricketZ	0.51	0.03	0.47	0.03	0.51	0.02	0.46	0.02	0.66	0.02	0.63	0.02	0.69	0.02	0.66	0.03
ECG	0.98	0.00	0.94	0.01	0.98	0.00	0.96	0.01	0.96	0.01	0.90	0.02	0.98	0.01	0.94	0.02
ECGFiveD	0.92	0.01	0.83	0.02	0.92	0.02	0.84	0.03	0.93	0.01	0.86	0.01	0.95	0.01	0.90	0.02
FaceAll	0.89	0.00	0.88	0.00	0.88	0.02	0.87	0.02	0.95	0.01	0.94	0.01	0.95	0.01	0.95	0.01
FacesUCR	0.89	0.00	0.88	0.00	0.88	0.02	0.87	0.02	0.94	0.01	0.94	0.01	0.96	0.02	0.95	0.02
Fish	0.74	0.01	0.69	0.01	0.74	0.04	0.70	0.04	0.73	0.03	0.69	0.04	0.75	0.04	0.70	0.05
GunPoint	0.87	0.03	0.74	0.07	0.89	0.03	0.78	0.06	0.92	0.01	0.83	0.03	0.96	0.04	0.92	0.08
Haptics	0.38	0.02	0.22	0.02	0.39	0.05	0.24	0.06	0.41	0.03	0.26	0.03	0.42	0.06	0.27	0.07
InlineS	0.36	0.04	0.25	0.04	0.39	0.02	0.28	0.02	0.44	0.02	0.34	0.02	0.41	0.03	0.31	0.04
Italy	0.95	0.01	0.90	0.02	0.96	0.01	0.92	0.02	0.92	0.02	0.85	0.03	0.94	0.02	0.89	0.05
Lighting2	0.70	0.07	0.38	0.11	0.69	0.04	0.38	0.08	0.77	0.07	0.52	0.12	0.78	0.11	0.54	0.24
Lighting7	0.57	0.04	0.48	0.04	0.56	0.03	0.46	0.04	0.70	0.03	0.64	0.03	0.70	0.06	0.64	0.07
MALLAT	0.96	0.01	0.95	0.01	0.96	0.01	0.96	0.01	0.96	0.01	0.96	0.01	0.97	0.00	0.97	0.00
MedicalI	0.70	0.02	0.56	0.03	0.70	0.02	0.56	0.03	0.74	0.03	0.62	0.04	0.74	0.04	0.63	0.06
MoteStrain	0.87	0.01	0.75	0.01	0.87	0.03	0.75	0.05	0.93	0.01	0.86	0.02	0.95	0.01	0.91	0.01
Olive	0.85	0.06	0.79	0.09	0.90	0.07	0.86	0.10	0.82	0.06	0.73	0.09	0.87	0.07	0.81	0.11
OSULeaf	0.55	0.04	0.45	0.05	0.56	0.05	0.46	0.06	0.61	0.02	0.53	0.03	0.60	0.07	0.52	0.08
Sony	0.93	0.02	0.85	0.04	0.95	0.03	0.90	0.06	0.93	0.01	0.85	0.02	0.94	0.04	0.89	0.08
SonyII	0.93	0.01	0.84	0.01	0.95	0.01	0.90	0.03	0.92	0.01	0.82	0.01	0.93	0.02	0.85	0.04
StarLightC	0.87	0.01	0.79	0.01	0.87	0.01	0.79	0.02	0.90	0.01	0.83	0.01	0.90	0.01	0.83	0.01
Synthetic	0.86	0.03	0.83	0.03	0.86	0.03	0.83	0.03	0.94	0.02	0.93	0.02	0.95	0.02	0.94	0.03
Trace	0.66	0.05	0.54	0.07	0.63	0.05	0.51	0.07	0.89	0.05	0.86	0.06	0.90	0.02	0.87	0.02
Two	0.92	0.00	0.89	0.01	0.92	0.01	0.89	0.01	0.99	0.00	0.98	0.00	0.99	0.00	0.99	0.00
TwoLeadE	0.90	0.01	0.81	0.03	0.94	0.02	0.87	0.03	0.92	0.01	0.84	0.02	0.95	0.01	0.90	0.02
uWaveGLX	0.73	0.00	0.70	0.00	0.74	0.01	0.70	0.02	0.76	0.01	0.73	0.01	0.77	0.01	0.73	0.01
uWaveGLY	0.68	0.01	0.63	0.01	0.67	0.01	0.63	0.01	0.69	0.00	0.65	0.00	0.69	0.01	0.65	0.02
uWaveGLZ	0.66	0.01	0.61	0.01	0.66	0.01	0.61	0.01	0.69	0.01	0.65	0.01	0.70	0.01	0.65	0.01
Wafer	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.96	0.01	0.99	0.00	0.97	0.01
Yoga	0.88	0.01	0.75	0.01	0.89	0.01	0.77	0.02	0.89	0.01	0.78	0.01	0.88	0.01	0.77	0.03

Tabla B.9: Resultados del aprendizaje inductivo y transductivo con el método SNNRCE en el 30 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.94	0.01	0.91	0.02	0.94	0.02	0.91	0.03	0.99	0.01	0.98	0.02	0.98	0.03	0.97	0.04
Chlorine	0.60	0.01	0.34	0.01	0.59	0.02	0.33	0.02	0.57	0.00	0.30	0.01	0.57	0.03	0.29	0.04
CinCECGt	0.98	0.00	0.98	0.01	0.99	0.01	0.98	0.01	0.98	0.01	0.98	0.02	0.98	0.01	0.98	0.02
Coffee	0.97	0.05	0.94	0.09	0.98	0.04	0.97	0.07	0.93	0.05	0.87	0.09	0.93	0.04	0.86	0.08
CricketX	0.33	0.03	0.26	0.03	0.34	0.02	0.28	0.03	0.49	0.01	0.44	0.01	0.50	0.03	0.45	0.03
CricketY	0.35	0.03	0.30	0.03	0.34	0.02	0.28	0.02	0.47	0.03	0.42	0.03	0.47	0.05	0.42	0.05
CricketZ	0.33	0.01	0.26	0.01	0.35	0.03	0.30	0.03	0.49	0.02	0.44	0.02	0.53	0.06	0.48	0.06
ECG	0.99	0.00	0.97	0.01	0.99	0.00	0.98	0.01	0.98	0.01	0.96	0.01	0.98	0.01	0.96	0.02
ECGFiveD	0.94	0.01	0.89	0.03	0.93	0.03	0.87	0.06	0.90	0.01	0.80	0.03	0.92	0.02	0.85	0.03
FaceAll	0.76	0.01	0.74	0.02	0.77	0.03	0.75	0.03	0.94	0.01	0.93	0.01	0.94	0.02	0.93	0.03
FacesUCR	0.75	0.03	0.73	0.04	0.76	0.03	0.73	0.04	0.94	0.01	0.93	0.02	0.94	0.01	0.93	0.01
Fish	0.58	0.07	0.51	0.08	0.62	0.06	0.55	0.07	0.63	0.04	0.57	0.04	0.60	0.07	0.53	0.08
GunPoint	0.73	0.08	0.47	0.16	0.76	0.11	0.51	0.22	0.80	0.11	0.60	0.22	0.82	0.08	0.63	0.16
Haptics	0.37	0.03	0.21	0.03	0.37	0.07	0.21	0.08	0.36	0.04	0.19	0.04	0.35	0.06	0.18	0.07
InlineS	0.27	0.02	0.14	0.02	0.26	0.01	0.13	0.01	0.32	0.03	0.20	0.04	0.30	0.02	0.18	0.03
Italy	0.96	0.01	0.92	0.03	0.96	0.01	0.92	0.02	0.94	0.01	0.88	0.01	0.94	0.01	0.88	0.02
Lighting2	0.65	0.05	0.23	0.13	0.63	0.09	0.19	0.20	0.72	0.07	0.36	0.17	0.70	0.13	0.32	0.31
Lighting7	0.43	0.02	0.33	0.02	0.50	0.07	0.40	0.09	0.52	0.04	0.44	0.04	0.59	0.08	0.52	0.09
MALLAT	0.96	0.01	0.96	0.01	0.96	0.01	0.96	0.01	0.98	0.01	0.97	0.01	0.98	0.02	0.97	0.02
MedicalI	0.58	0.02	0.40	0.04	0.58	0.05	0.41	0.05	0.60	0.01	0.44	0.02	0.61	0.04	0.45	0.05
MoteStrain	0.91	0.01	0.81	0.01	0.88	0.02	0.76	0.04	0.95	0.01	0.90	0.01	0.95	0.01	0.89	0.02
Olive	0.86	0.04	0.79	0.06	0.80	0.07	0.72	0.10	0.84	0.05	0.76	0.08	0.77	0.07	0.67	0.10
OSULeaf	0.43	0.02	0.30	0.03	0.42	0.05	0.28	0.06	0.51	0.03	0.40	0.04	0.49	0.05	0.37	0.06
Sony	0.92	0.05	0.84	0.10	0.91	0.07	0.83	0.13	0.90	0.05	0.80	0.09	0.88	0.03	0.77	0.06
SonyII	0.91	0.03	0.81	0.06	0.91	0.02	0.80	0.05	0.91	0.02	0.80	0.04	0.91	0.02	0.81	0.04
StarLightC	0.85	0.01	0.75	0.01	0.81	0.02	0.71	0.03	0.87	0.01	0.79	0.01	0.84	0.01	0.74	0.02
Synthetic	0.78	0.04	0.74	0.05	0.76	0.04	0.71	0.05	0.90	0.05	0.88	0.06	0.90	0.06	0.88	0.07
Trace	0.58	0.06	0.44	0.08	0.53	0.06	0.37	0.09	0.75	0.06	0.67	0.08	0.74	0.10	0.65	0.13
Two	0.78	0.01	0.71	0.01	0.79	0.02	0.72	0.03	0.98	0.01	0.97	0.01	0.98	0.01	0.98	0.01
TwoLeadE	0.90	0.04	0.80	0.08	0.91	0.04	0.82	0.09	0.98	0.01	0.96	0.01	0.99	0.01	0.97	0.01
uWaveGLX	0.71	0.01	0.67	0.01	0.70	0.01	0.66	0.01	0.73	0.00	0.69	0.00	0.73	0.01	0.69	0.01
uWaveGLY	0.63	0.01	0.57	0.01	0.61	0.01	0.56	0.01	0.65	0.01	0.61	0.01	0.65	0.02	0.60	0.02
uWaveGLZ	0.63	0.00	0.58	0.00	0.60	0.01	0.54	0.01	0.65	0.01	0.61	0.01	0.64	0.01	0.58	0.01
Wafer	0.99	0.00	0.97	0.02	0.99	0.00	0.97	0.02	0.99	0.00	0.97	0.01	1.00	0.00	0.97	0.01
Yoga	0.82	0.01	0.63	0.02	0.82	0.02	0.63	0.03	0.85	0.01	0.69	0.02	0.84	0.02	0.68	0.04

Tabla B.10: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-1NN en el 10 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.97	0.01	0.96	0.02	0.97	0.02	0.96	0.04	1.00	0.00	1.00	0.00	1.00	0.00	0.99	0.01
Chlorine	0.72	0.02	0.54	0.03	0.74	0.01	0.57	0.02	0.68	0.01	0.47	0.02	0.70	0.01	0.50	0.03
CinCECGt	0.99	0.01	0.98	0.01	0.99	0.01	0.98	0.01	0.99	0.01	0.98	0.01	0.99	0.01	0.99	0.01
Coffee	0.94	0.02	0.89	0.04	0.95	0.05	0.89	0.10	0.90	0.03	0.81	0.06	0.91	0.10	0.83	0.20
CricketX	0.41	0.03	0.35	0.03	0.41	0.04	0.36	0.05	0.60	0.03	0.56	0.04	0.58	0.05	0.55	0.05
CricketY	0.42	0.04	0.37	0.04	0.41	0.05	0.36	0.05	0.58	0.02	0.54	0.02	0.57	0.02	0.53	0.02
CricketZ	0.42	0.04	0.37	0.04	0.41	0.03	0.36	0.03	0.61	0.01	0.58	0.01	0.62	0.02	0.59	0.03
ECG	0.99	0.00	0.98	0.00	1.00	0.00	0.99	0.01	0.99	0.01	0.98	0.02	0.99	0.01	0.98	0.02
ECGFiveD	0.97	0.01	0.94	0.02	0.97	0.02	0.94	0.03	0.94	0.02	0.88	0.04	0.95	0.01	0.91	0.02
FaceAll	0.84	0.02	0.83	0.02	0.85	0.02	0.84	0.02	0.96	0.00	0.96	0.00	0.97	0.01	0.96	0.01
FacesUCR	0.84	0.02	0.82	0.03	0.84	0.02	0.83	0.02	0.96	0.01	0.96	0.01	0.97	0.01	0.96	0.01
Fish	0.67	0.02	0.61	0.02	0.67	0.01	0.61	0.01	0.71	0.06	0.66	0.06	0.67	0.05	0.61	0.05
GunPoint	0.80	0.03	0.60	0.07	0.80	0.05	0.59	0.11	0.92	0.03	0.83	0.06	0.90	0.06	0.80	0.13
Haptics	0.36	0.05	0.20	0.06	0.35	0.04	0.18	0.05	0.39	0.05	0.24	0.07	0.37	0.04	0.21	0.04
InlineS	0.32	0.01	0.20	0.02	0.30	0.03	0.18	0.04	0.38	0.02	0.27	0.02	0.35	0.04	0.24	0.04
Italy	0.97	0.01	0.93	0.02	0.96	0.01	0.92	0.02	0.96	0.01	0.92	0.01	0.95	0.01	0.89	0.03
Lighting2	0.67	0.04	0.31	0.09	0.66	0.15	0.29	0.33	0.76	0.03	0.47	0.05	0.75	0.05	0.46	0.13
Lighting7	0.54	0.05	0.44	0.06	0.55	0.07	0.45	0.10	0.66	0.07	0.59	0.08	0.69	0.06	0.63	0.07
MALLAT	0.97	0.00	0.97	0.00	0.97	0.01	0.97	0.01	0.98	0.00	0.98	0.00	0.99	0.01	0.98	0.01
MedicalI	0.64	0.01	0.47	0.01	0.62	0.03	0.43	0.06	0.69	0.03	0.55	0.03	0.67	0.03	0.53	0.03
MoteStrain	0.90	0.01	0.79	0.02	0.89	0.03	0.78	0.06	0.96	0.01	0.91	0.02	0.96	0.01	0.91	0.02
Olive	0.79	0.06	0.70	0.08	0.80	0.13	0.72	0.18	0.81	0.06	0.72	0.08	0.80	0.10	0.72	0.13
OSULeaf	0.51	0.04	0.40	0.04	0.46	0.08	0.34	0.09	0.56	0.05	0.46	0.06	0.54	0.05	0.43	0.06
Sony	0.95	0.03	0.89	0.05	0.95	0.03	0.89	0.05	0.93	0.02	0.85	0.03	0.92	0.02	0.85	0.04
SonyII	0.94	0.02	0.88	0.04	0.95	0.03	0.88	0.06	0.93	0.01	0.85	0.01	0.94	0.03	0.87	0.06
StarLightC	0.86	0.01	0.77	0.01	0.83	0.01	0.73	0.02	0.88	0.00	0.81	0.00	0.85	0.01	0.76	0.02
Synthetic	0.86	0.01	0.83	0.02	0.85	0.02	0.82	0.03	0.96	0.01	0.96	0.01	0.96	0.01	0.96	0.01
Trace	0.59	0.05	0.45	0.06	0.63	0.09	0.51	0.12	0.87	0.04	0.83	0.06	0.88	0.08	0.85	0.10
Two	0.88	0.00	0.84	0.00	0.88	0.01	0.85	0.01	1.00	0.00	0.99	0.00	1.00	0.00	1.00	0.00
TwoLeadE	0.97	0.01	0.95	0.02	0.98	0.01	0.95	0.02	0.99	0.00	0.98	0.01	0.99	0.01	0.99	0.01
uWaveGLX	0.72	0.01	0.68	0.01	0.71	0.01	0.67	0.01	0.75	0.01	0.71	0.01	0.74	0.01	0.70	0.01
uWaveGLY	0.66	0.00	0.61	0.01	0.63	0.01	0.57	0.01	0.68	0.01	0.64	0.01	0.67	0.01	0.62	0.01
uWaveGLZ	0.66	0.01	0.61	0.01	0.63	0.02	0.58	0.02	0.68	0.01	0.64	0.01	0.66	0.02	0.61	0.02
Wafer	1.00	0.00	0.98	0.00	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	1.00	0.00	0.97	0.01
Yoga	0.87	0.01	0.74	0.03	0.87	0.01	0.73	0.02	0.89	0.01	0.77	0.02	0.88	0.01	0.76	0.01

Tabla B.11: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-1NN en el 20 % de ejemplos etiquetados.



Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.98	0.01	0.97	0.01	0.98	0.00	0.97	0.01	1.00	0.00	1.00	0.00	0.99	0.01	0.99	0.01
Chlorine	0.81	0.01	0.68	0.02	0.83	0.01	0.71	0.02	0.77	0.01	0.62	0.02	0.78	0.01	0.64	0.02
CinCECGt	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
Coffee	0.99	0.02	0.97	0.04	0.98	0.04	0.97	0.07	0.97	0.03	0.93	0.07	0.96	0.05	0.93	0.10
CricketX	0.50	0.04	0.45	0.04	0.49	0.04	0.45	0.04	0.67	0.02	0.63	0.02	0.67	0.04	0.63	0.05
CricketY	0.45	0.03	0.40	0.03	0.46	0.03	0.41	0.03	0.63	0.04	0.60	0.05	0.62	0.02	0.58	0.03
CricketZ	0.51	0.04	0.46	0.04	0.49	0.03	0.44	0.03	0.66	0.02	0.63	0.02	0.69	0.03	0.66	0.04
ECG	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.01
ECGFiveD	0.97	0.01	0.94	0.01	0.97	0.01	0.94	0.03	0.97	0.01	0.94	0.02	0.97	0.02	0.94	0.03
FaceAll	0.89	0.01	0.88	0.01	0.89	0.01	0.88	0.01	0.97	0.01	0.97	0.01	0.97	0.00	0.97	0.00
FacesUCR	0.89	0.01	0.88	0.01	0.90	0.02	0.89	0.02	0.97	0.00	0.97	0.00	0.98	0.00	0.97	0.00
Fish	0.73	0.02	0.69	0.02	0.73	0.03	0.68	0.04	0.74	0.03	0.70	0.03	0.73	0.04	0.69	0.05
GunPoint	0.90	0.03	0.80	0.05	0.87	0.04	0.74	0.08	0.96	0.03	0.92	0.05	0.97	0.03	0.94	0.05
Haptics	0.37	0.01	0.21	0.01	0.35	0.07	0.18	0.09	0.39	0.03	0.24	0.04	0.38	0.05	0.22	0.07
InlineS	0.37	0.03	0.25	0.03	0.35	0.04	0.23	0.04	0.45	0.03	0.35	0.03	0.42	0.06	0.32	0.08
Italy	0.96	0.01	0.92	0.02	0.96	0.01	0.93	0.02	0.95	0.01	0.90	0.02	0.95	0.01	0.90	0.02
Lighting2	0.65	0.09	0.31	0.14	0.69	0.06	0.37	0.09	0.80	0.04	0.58	0.07	0.84	0.10	0.66	0.24
Lighting7	0.56	0.03	0.47	0.03	0.56	0.05	0.47	0.06	0.70	0.02	0.64	0.02	0.66	0.06	0.59	0.07
MALLAT	0.97	0.00	0.97	0.00	0.97	0.01	0.97	0.01	0.99	0.00	0.99	0.00	0.99	0.01	0.99	0.01
MedicalI	0.69	0.03	0.55	0.03	0.69	0.03	0.54	0.04	0.73	0.03	0.61	0.04	0.71	0.04	0.58	0.06
MoteStrain	0.90	0.01	0.79	0.02	0.89	0.04	0.78	0.08	0.96	0.01	0.93	0.01	0.97	0.01	0.94	0.02
Olive	0.87	0.03	0.81	0.05	0.88	0.10	0.83	0.14	0.85	0.02	0.78	0.03	0.88	0.05	0.83	0.07
OSULeaf	0.54	0.03	0.44	0.04	0.57	0.03	0.47	0.04	0.61	0.01	0.52	0.02	0.61	0.06	0.53	0.07
Sony	0.96	0.02	0.92	0.04	0.97	0.02	0.94	0.04	0.95	0.01	0.90	0.02	0.93	0.04	0.87	0.08
SonyII	0.96	0.01	0.91	0.02	0.96	0.01	0.92	0.02	0.96	0.01	0.90	0.02	0.96	0.01	0.91	0.02
StarLightC	0.86	0.01	0.78	0.01	0.83	0.01	0.73	0.02	0.89	0.01	0.82	0.01	0.86	0.01	0.77	0.02
Synthetic	0.87	0.02	0.84	0.03	0.86	0.02	0.83	0.02	0.96	0.01	0.96	0.01	0.96	0.02	0.95	0.02
Trace	0.64	0.04	0.52	0.05	0.66	0.06	0.55	0.08	0.90	0.05	0.86	0.06	0.90	0.02	0.86	0.03
Two	0.93	0.00	0.90	0.00	0.93	0.01	0.90	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
TwoLeadE	0.98	0.01	0.96	0.02	0.99	0.01	0.97	0.01	0.99	0.00	0.99	0.01	1.00	0.00	0.99	0.01
uWaveGLX	0.74	0.01	0.70	0.01	0.73	0.02	0.69	0.02	0.77	0.01	0.73	0.01	0.76	0.01	0.73	0.01
uWaveGLY	0.67	0.01	0.62	0.01	0.65	0.01	0.60	0.01	0.69	0.01	0.65	0.01	0.68	0.01	0.64	0.01
uWaveGLZ	0.67	0.01	0.62	0.01	0.65	0.01	0.60	0.01	0.70	0.01	0.65	0.01	0.68	0.02	0.64	0.02
Wafer	1.00	0.00	0.99	0.00	1.00	0.00	0.98	0.00	1.00	0.00	0.98	0.01	1.00	0.00	0.99	0.00
Yoga	0.89	0.01	0.79	0.01	0.89	0.01	0.79	0.01	0.91	0.01	0.81	0.02	0.90	0.01	0.80	0.02

Tabla B.12: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-1NN en el 30 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.91	0.01	0.87	0.02	0.92	0.01	0.89	0.02	0.96	0.03	0.94	0.05	0.96	0.05	0.94	0.07
Chlorine	0.60	0.02	0.33	0.03	0.60	0.03	0.31	0.05	0.59	0.02	0.28	0.03	0.57	0.03	0.24	0.03
CinCECGt	0.97	0.00	0.96	0.01	0.97	0.01	0.97	0.02	0.99	0.00	0.98	0.00	0.99	0.01	0.98	0.01
Coffee	0.89	0.14	0.78	0.27	0.95	0.05	0.89	0.10	0.90	0.05	0.81	0.11	0.95	0.07	0.90	0.15
CricketX	0.30	0.02	0.24	0.02	0.33	0.04	0.26	0.04	0.46	0.03	0.41	0.03	0.43	0.02	0.38	0.02
CricketY	0.34	0.03	0.28	0.03	0.34	0.04	0.28	0.05	0.44	0.02	0.39	0.02	0.42	0.05	0.37	0.05
CricketZ	0.31	0.02	0.24	0.02	0.33	0.02	0.27	0.02	0.46	0.02	0.41	0.02	0.48	0.05	0.43	0.05
ECG	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.98	0.01	0.95	0.03	0.98	0.01	0.95	0.03
ECGFiveD	0.97	0.02	0.94	0.03	0.96	0.02	0.93	0.04	0.88	0.05	0.77	0.09	0.89	0.03	0.79	0.05
FaceAll	0.74	0.03	0.72	0.04	0.74	0.03	0.71	0.03	0.91	0.01	0.90	0.02	0.91	0.03	0.90	0.03
FacesUCR	0.72	0.04	0.70	0.04	0.74	0.04	0.71	0.04	0.90	0.02	0.89	0.02	0.91	0.03	0.90	0.03
Fish	0.55	0.07	0.48	0.08	0.55	0.07	0.48	0.08	0.63	0.05	0.57	0.05	0.63	0.11	0.57	0.13
GunPoint	0.77	0.06	0.54	0.11	0.78	0.04	0.56	0.07	0.79	0.07	0.58	0.13	0.80	0.05	0.59	0.10
Haptics	0.38	0.05	0.22	0.06	0.37	0.07	0.20	0.08	0.36	0.04	0.19	0.06	0.36	0.02	0.20	0.03
InlineS	0.27	0.02	0.14	0.02	0.30	0.02	0.17	0.02	0.29	0.02	0.16	0.03	0.30	0.03	0.18	0.04
Italy	0.96	0.01	0.93	0.02	0.97	0.01	0.94	0.01	0.96	0.01	0.91	0.03	0.95	0.02	0.90	0.03
Lighting2	0.59	0.09	0.15	0.12	0.61	0.11	0.17	0.19	0.70	0.06	0.33	0.14	0.69	0.09	0.31	0.20
Lighting7	0.45	0.05	0.35	0.05	0.42	0.05	0.31	0.05	0.53	0.03	0.44	0.04	0.56	0.06	0.48	0.07
MALLAT	0.97	0.01	0.97	0.01	0.98	0.00	0.97	0.00	0.98	0.00	0.98	0.00	0.98	0.01	0.98	0.01
MedicalI	0.59	0.01	0.40	0.03	0.57	0.03	0.37	0.04	0.60	0.02	0.41	0.04	0.61	0.04	0.41	0.06
MoteStrain	0.89	0.01	0.78	0.01	0.89	0.02	0.78	0.04	0.94	0.02	0.88	0.04	0.94	0.02	0.89	0.03
Olive	0.84	0.05	0.77	0.08	0.83	0.06	0.76	0.08	0.84	0.05	0.76	0.07	0.80	0.10	0.72	0.13
OSULeaf	0.41	0.05	0.28	0.06	0.39	0.06	0.25	0.07	0.49	0.04	0.37	0.05	0.47	0.07	0.35	0.08
Sony	0.94	0.03	0.88	0.06	0.97	0.02	0.94	0.04	0.95	0.02	0.91	0.04	0.96	0.02	0.92	0.05
SonyII	0.91	0.04	0.81	0.08	0.93	0.04	0.84	0.10	0.90	0.02	0.79	0.05	0.91	0.01	0.81	0.03
StarLightC	0.94	0.00	0.90	0.01	0.93	0.01	0.89	0.02	0.91	0.00	0.84	0.01	0.91	0.01	0.84	0.01
Synthetic	0.76	0.04	0.71	0.05	0.80	0.08	0.75	0.10	0.84	0.05	0.81	0.06	0.87	0.06	0.85	0.08
Trace	0.57	0.05	0.44	0.07	0.55	0.06	0.40	0.07	0.68	0.11	0.57	0.14	0.68	0.12	0.57	0.16
Two	0.74	0.01	0.65	0.01	0.73	0.02	0.64	0.02	0.96	0.01	0.94	0.01	0.97	0.01	0.95	0.01
TwoLeadE	0.94	0.02	0.88	0.04	0.95	0.02	0.90	0.04	0.96	0.01	0.92	0.03	0.97	0.01	0.94	0.03
uWaveGLX	0.71	0.01	0.67	0.01	0.70	0.02	0.66	0.03	0.73	0.01	0.70	0.01	0.72	0.02	0.68	0.02
uWaveGLY	0.64	0.01	0.59	0.01	0.64	0.02	0.59	0.03	0.67	0.01	0.62	0.01	0.68	0.01	0.63	0.01
uWaveGLZ	0.65	0.01	0.60	0.01	0.62	0.01	0.57	0.01	0.66	0.00	0.61	0.01	0.63	0.02	0.58	0.02
Wafer	0.99	0.00	0.95	0.02	0.99	0.00	0.96	0.02	0.99	0.00	0.94	0.02	0.99	0.00	0.94	0.02
Yoga	0.82	0.01	0.63	0.03	0.83	0.02	0.65	0.04	0.75	0.03	0.49	0.05	0.76	0.04	0.51	0.07

Tabla B.13: Resultados del aprendizaje inductivo y transductivo con el método Democratic en el 10 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.97	0.02	0.95	0.03	0.97	0.02	0.95	0.03	1.00	0.00	0.99	0.00	1.00	0.00	1.00	0.00
Chlorine	0.71	0.02	0.49	0.04	0.72	0.02	0.51	0.03	0.67	0.01	0.42	0.03	0.67	0.02	0.42	0.03
CinCECGt	0.98	0.01	0.97	0.02	0.98	0.01	0.97	0.02	1.00	0.00	0.99	0.01	0.99	0.01	0.99	0.01
Coffee	0.96	0.03	0.91	0.06	0.98	0.04	0.97	0.07	0.90	0.09	0.80	0.18	0.93	0.07	0.86	0.14
CricketX	0.39	0.03	0.34	0.03	0.40	0.04	0.34	0.04	0.56	0.04	0.52	0.05	0.55	0.02	0.51	0.03
CricketY	0.39	0.03	0.33	0.03	0.39	0.07	0.34	0.07	0.53	0.03	0.49	0.03	0.54	0.03	0.50	0.03
CricketZ	0.40	0.05	0.34	0.05	0.38	0.02	0.32	0.02	0.59	0.02	0.55	0.02	0.57	0.02	0.53	0.02
ECG	0.99	0.00	0.98	0.00	0.99	0.00	0.98	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.98	0.01
ECGFiveD	0.99	0.01	0.98	0.02	0.99	0.01	0.98	0.02	0.94	0.02	0.88	0.05	0.96	0.03	0.92	0.05
FaceAll	0.83	0.02	0.81	0.02	0.83	0.03	0.82	0.03	0.94	0.01	0.94	0.01	0.95	0.02	0.94	0.02
FacesUCR	0.82	0.02	0.80	0.02	0.82	0.02	0.80	0.02	0.94	0.01	0.94	0.01	0.95	0.01	0.95	0.01
Fish	0.64	0.03	0.58	0.04	0.63	0.04	0.57	0.05	0.68	0.04	0.63	0.05	0.67	0.03	0.62	0.03
GunPoint	0.84	0.08	0.69	0.16	0.79	0.07	0.58	0.14	0.92	0.05	0.85	0.10	0.89	0.06	0.78	0.13
Haptics	0.37	0.04	0.21	0.05	0.35	0.05	0.19	0.06	0.39	0.06	0.23	0.07	0.37	0.03	0.21	0.04
InlineS	0.31	0.01	0.19	0.01	0.29	0.03	0.17	0.04	0.36	0.03	0.25	0.03	0.36	0.03	0.24	0.04
Italy	0.97	0.00	0.95	0.01	0.97	0.01	0.95	0.02	0.97	0.00	0.95	0.01	0.96	0.01	0.93	0.02
Lighting2	0.63	0.06	0.22	0.10	0.58	0.10	0.14	0.19	0.70	0.10	0.34	0.20	0.65	0.11	0.25	0.28
Lighting7	0.52	0.06	0.41	0.08	0.51	0.06	0.40	0.07	0.66	0.11	0.59	0.13	0.64	0.05	0.57	0.06
MALLAT	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00	0.99	0.00	0.98	0.00	0.99	0.00	0.98	0.00
MedicalI	0.64	0.02	0.44	0.02	0.61	0.04	0.40	0.04	0.66	0.01	0.46	0.00	0.66	0.03	0.49	0.04
MoteStrain	0.90	0.01	0.80	0.01	0.90	0.01	0.81	0.01	0.94	0.01	0.88	0.03	0.95	0.02	0.90	0.04
Olive	0.79	0.07	0.69	0.09	0.83	0.06	0.77	0.08	0.81	0.07	0.72	0.10	0.82	0.07	0.74	0.09
OSULeaf	0.47	0.07	0.35	0.09	0.44	0.06	0.31	0.08	0.53	0.03	0.43	0.03	0.50	0.07	0.39	0.09
Sony	0.97	0.02	0.93	0.03	0.97	0.02	0.94	0.03	0.96	0.02	0.93	0.04	0.97	0.04	0.93	0.08
SonyII	0.95	0.02	0.90	0.04	0.94	0.01	0.88	0.03	0.94	0.01	0.87	0.02	0.94	0.02	0.86	0.05
StarLightC	0.95	0.01	0.91	0.01	0.94	0.01	0.90	0.01	0.88	0.01	0.78	0.01	0.89	0.01	0.79	0.02
Synthetic	0.85	0.04	0.83	0.05	0.89	0.05	0.87	0.07	0.91	0.02	0.90	0.03	0.93	0.04	0.92	0.05
Trace	0.58	0.08	0.44	0.11	0.60	0.06	0.47	0.08	0.81	0.04	0.74	0.05	0.86	0.03	0.82	0.04
Two	0.85	0.01	0.81	0.01	0.85	0.01	0.81	0.02	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
TwoLeadE	0.99	0.01	0.98	0.02	0.99	0.01	0.99	0.02	0.99	0.00	0.98	0.01	0.99	0.01	0.98	0.02
uWaveGLX	0.75	0.01	0.71	0.01	0.75	0.02	0.71	0.02	0.77	0.01	0.73	0.01	0.75	0.01	0.71	0.02
uWaveGLY	0.68	0.01	0.64	0.01	0.67	0.01	0.63	0.01	0.70	0.01	0.66	0.01	0.68	0.01	0.63	0.01
uWaveGLZ	0.68	0.01	0.64	0.01	0.66	0.03	0.62	0.04	0.70	0.01	0.65	0.01	0.67	0.02	0.63	0.02
Wafer	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.94	0.02	0.99	0.00	0.94	0.02
Yoga	0.88	0.01	0.76	0.03	0.88	0.02	0.75	0.04	0.78	0.00	0.56	0.01	0.80	0.01	0.59	0.02

Tabla B.14: Resultados del aprendizaje inductivo y transductivo con el método Democratic en el 20 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.99	0.01	0.98	0.01	0.99	0.01	0.98	0.02	0.99	0.00	0.99	0.01	1.00	0.00	1.00	0.01
Chlorine	0.81	0.02	0.67	0.03	0.81	0.01	0.69	0.03	0.75	0.01	0.56	0.01	0.76	0.00	0.58	0.01
CinCECGt	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.01
Coffee	0.96	0.04	0.92	0.08	0.98	0.04	0.97	0.07	0.95	0.04	0.91	0.07	1.00	0.00	1.00	0.00
CricketX	0.47	0.04	0.42	0.04	0.46	0.06	0.41	0.06	0.64	0.01	0.61	0.01	0.63	0.03	0.59	0.03
CricketY	0.43	0.04	0.38	0.05	0.41	0.04	0.36	0.05	0.60	0.04	0.57	0.05	0.61	0.05	0.57	0.05
CricketZ	0.50	0.03	0.46	0.03	0.48	0.03	0.44	0.03	0.61	0.04	0.58	0.04	0.65	0.03	0.62	0.03
ECG	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.01	0.99	0.00	0.99	0.01	1.00	0.00	0.99	0.01
ECGFiveD	0.98	0.01	0.97	0.02	0.97	0.01	0.95	0.03	0.97	0.01	0.93	0.03	0.97	0.01	0.94	0.02
FaceAll	0.88	0.00	0.87	0.00	0.88	0.02	0.86	0.02	0.96	0.01	0.96	0.01	0.96	0.01	0.95	0.01
FacesUCR	0.88	0.02	0.87	0.02	0.88	0.02	0.86	0.02	0.96	0.00	0.96	0.00	0.96	0.01	0.96	0.02
Fish	0.73	0.02	0.68	0.02	0.71	0.05	0.66	0.06	0.73	0.04	0.69	0.04	0.71	0.04	0.66	0.04
GunPoint	0.91	0.01	0.81	0.02	0.90	0.04	0.81	0.09	0.94	0.03	0.89	0.05	0.97	0.03	0.94	0.07
Haptics	0.39	0.03	0.23	0.04	0.38	0.06	0.22	0.07	0.40	0.02	0.24	0.03	0.40	0.04	0.24	0.05
InlineS	0.36	0.03	0.25	0.04	0.36	0.02	0.25	0.03	0.43	0.02	0.33	0.02	0.40	0.03	0.29	0.04
Italy	0.97	0.00	0.94	0.01	0.97	0.01	0.95	0.02	0.96	0.00	0.93	0.01	0.97	0.01	0.93	0.02
Lighting2	0.63	0.10	0.23	0.17	0.66	0.06	0.28	0.13	0.80	0.04	0.56	0.09	0.78	0.06	0.51	0.14
Lighting7	0.52	0.02	0.42	0.02	0.55	0.03	0.45	0.04	0.70	0.04	0.64	0.05	0.64	0.04	0.57	0.05
MALLAT	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00	0.99	0.00	0.99	0.00	0.99	0.01	0.99	0.01
MedicalI	0.69	0.02	0.54	0.02	0.70	0.04	0.55	0.06	0.70	0.03	0.54	0.06	0.71	0.04	0.58	0.06
MoteStrain	0.91	0.01	0.83	0.03	0.92	0.02	0.83	0.04	0.96	0.01	0.92	0.02	0.97	0.01	0.93	0.03
Olive	0.84	0.03	0.76	0.05	0.87	0.05	0.81	0.07	0.84	0.05	0.77	0.08	0.85	0.07	0.79	0.10
OSULeaf	0.53	0.04	0.43	0.05	0.56	0.05	0.46	0.06	0.58	0.02	0.49	0.03	0.58	0.03	0.49	0.03
Sony	0.99	0.01	0.97	0.01	0.99	0.01	0.97	0.02	0.99	0.01	0.97	0.01	0.99	0.01	0.97	0.01
SonyII	0.96	0.01	0.92	0.02	0.96	0.02	0.91	0.04	0.95	0.01	0.90	0.02	0.96	0.01	0.91	0.03
StarLightC	0.95	0.02	0.90	0.04	0.93	0.03	0.88	0.05	0.88	0.00	0.78	0.00	0.89	0.01	0.80	0.01
Synthetic	0.88	0.03	0.86	0.03	0.93	0.04	0.92	0.04	0.94	0.01	0.93	0.01	0.96	0.00	0.95	0.00
Trace	0.65	0.04	0.54	0.05	0.63	0.04	0.51	0.06	0.82	0.06	0.76	0.08	0.86	0.05	0.82	0.07
Two	0.92	0.01	0.90	0.01	0.92	0.00	0.89	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
TwoLeadE	0.99	0.00	0.99	0.01	1.00	0.00	0.99	0.01	0.98	0.01	0.97	0.02	0.99	0.01	0.98	0.03
uWaveGLX	0.77	0.01	0.74	0.01	0.77	0.02	0.73	0.02	0.78	0.01	0.75	0.01	0.77	0.02	0.73	0.02
uWaveGLY	0.71	0.01	0.67	0.01	0.69	0.01	0.65	0.01	0.72	0.01	0.68	0.01	0.69	0.01	0.64	0.01
uWaveGLZ	0.69	0.01	0.65	0.01	0.67	0.02	0.63	0.02	0.71	0.01	0.67	0.01	0.70	0.01	0.65	0.01
Wafer	1.00	0.00	0.98	0.00	1.00	0.00	0.98	0.01	0.99	0.00	0.95	0.02	0.99	0.00	0.96	0.02
Yoga	0.91	0.01	0.82	0.03	0.90	0.01	0.81	0.02	0.80	0.01	0.59	0.02	0.83	0.02	0.65	0.04

Tabla B.15: Resultados del aprendizaje inductivo y transductivo con el método Democratic en el 30 % de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.90	0.06	0.85	0.09	0.91	0.06	0.86	0.09	0.95	0.03	0.93	0.05	0.95	0.04	0.93	0.06
Chlorine	0.57	0.05	0.24	0.05	0.57	0.05	0.24	0.06	0.58	0.03	0.22	0.06	0.58	0.03	0.22	0.06
CinCECGt	0.97	0.01	0.96	0.01	0.97	0.01	0.96	0.01	0.97	0.02	0.96	0.03	0.97	0.02	0.95	0.03
Coffee	0.64	0.20	0.29	0.40	0.67	0.26	0.37	0.51	0.68	0.23	0.37	0.44	0.71	0.23	0.43	0.43
CricketX	0.13	0.03	0.05	0.03	0.14	0.04	0.07	0.04	0.22	0.01	0.15	0.01	0.22	0.02	0.15	0.02
CricketY	0.18	0.03	0.11	0.03	0.19	0.06	0.12	0.07	0.23	0.02	0.16	0.02	0.24	0.02	0.17	0.02
CricketZ	0.15	0.05	0.08	0.06	0.15	0.04	0.07	0.05	0.20	0.03	0.13	0.04	0.21	0.01	0.14	0.02
ECG	0.98	0.01	0.96	0.02	0.98	0.01	0.96	0.02	0.97	0.01	0.93	0.03	0.98	0.01	0.94	0.04
ECGFiveD	0.98	0.02	0.95	0.03	0.97	0.02	0.94	0.04	0.91	0.04	0.83	0.07	0.92	0.02	0.85	0.04
FaceAll	0.40	0.06	0.34	0.07	0.42	0.07	0.36	0.07	0.52	0.05	0.47	0.06	0.54	0.06	0.50	0.06
FacesUCR	0.39	0.03	0.33	0.03	0.40	0.05	0.35	0.05	0.57	0.04	0.53	0.04	0.58	0.04	0.54	0.05
Fish	0.42	0.10	0.32	0.11	0.41	0.10	0.31	0.12	0.25	0.10	0.13	0.11	0.25	0.10	0.13	0.12
GunPoint	0.68	0.11	0.37	0.22	0.70	0.12	0.41	0.25	0.74	0.08	0.49	0.15	0.78	0.06	0.55	0.11
Haptics	0.27	0.10	0.08	0.12	0.28	0.09	0.08	0.12	0.31	0.07	0.14	0.09	0.33	0.04	0.16	0.05
InlineS	0.16	0.03	0.01	0.04	0.18	0.03	0.04	0.04	0.18	0.02	0.03	0.02	0.19	0.04	0.05	0.05
Italy	0.96	0.01	0.92	0.02	0.96	0.01	0.92	0.01	0.95	0.03	0.91	0.06	0.94	0.05	0.89	0.10
Lighting2	0.55	0.03	0.07	0.05	0.54	0.10	0.01	0.15	0.55	0.12	0.10	0.18	0.55	0.16	0.07	0.30
Lighting7	0.19	0.04	0.06	0.04	0.24	0.07	0.12	0.07	0.17	0.09	0.05	0.08	0.25	0.14	0.12	0.15
MALLAT	0.71	0.08	0.66	0.10	0.71	0.08	0.67	0.10	0.75	0.09	0.72	0.10	0.76	0.09	0.72	0.10
MedicalI	0.41	0.06	0.21	0.07	0.44	0.06	0.23	0.09	0.44	0.03	0.21	0.04	0.43	0.03	0.20	0.04
MoteStrain	0.87	0.03	0.74	0.06	0.87	0.04	0.75	0.09	0.94	0.02	0.88	0.04	0.93	0.02	0.87	0.03
Olive	0.76	0.12	0.67	0.15	0.75	0.18	0.67	0.22	0.65	0.16	0.52	0.21	0.57	0.19	0.42	0.25
OSULeaf	0.28	0.02	0.13	0.02	0.30	0.03	0.15	0.04	0.28	0.04	0.13	0.04	0.30	0.06	0.15	0.06
Sony	0.95	0.02	0.89	0.04	0.95	0.03	0.91	0.06	0.94	0.03	0.88	0.06	0.96	0.02	0.92	0.03
SonyII	0.92	0.05	0.82	0.10	0.91	0.05	0.82	0.13	0.90	0.02	0.78	0.05	0.89	0.04	0.76	0.09
StarLightC	0.95	0.00	0.91	0.00	0.95	0.01	0.92	0.02	0.90	0.01	0.81	0.02	0.90	0.01	0.81	0.02
Synthetic	0.50	0.10	0.40	0.12	0.57	0.10	0.48	0.12	0.61	0.09	0.53	0.10	0.63	0.12	0.56	0.15
Trace	0.39	0.08	0.20	0.11	0.40	0.12	0.20	0.17	0.46	0.18	0.29	0.23	0.45	0.16	0.27	0.21
Two	0.78	0.03	0.71	0.05	0.77	0.04	0.70	0.06	0.95	0.01	0.93	0.02	0.95	0.01	0.93	0.02
TwoLeadE	0.98	0.01	0.96	0.02	0.98	0.01	0.96	0.02	0.95	0.03	0.89	0.07	0.95	0.04	0.89	0.07
uWaveGLX	0.64	0.05	0.59	0.06	0.66	0.05	0.61	0.05	0.65	0.03	0.60	0.03	0.66	0.04	0.61	0.05
uWaveGLY	0.60	0.03	0.54	0.04	0.60	0.04	0.55	0.05	0.62	0.03	0.56	0.03	0.62	0.04	0.56	0.04
uWaveGLZ	0.58	0.03	0.52	0.03	0.56	0.03	0.50	0.03	0.57	0.04	0.51	0.05	0.57	0.05	0.50	0.06
Wafer	0.99	0.00	0.96	0.02	0.99	0.00	0.96	0.02	0.99	0.00	0.92	0.03	0.99	0.01	0.93	0.03
Yoga	0.83	0.02	0.66	0.03	0.84	0.01	0.67	0.02	0.78	0.01	0.55	0.02	0.78	0.02	0.56	0.04

Tabla B.16: Resultados del aprendizaje inductivo y transductivo con el método Self-training-SVM en el 10% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.99	0.00	0.98	0.01	0.99	0.01	0.98	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.02
Chlorine	0.66	0.06	0.38	0.13	0.67	0.06	0.38	0.13	0.61	0.01	0.27	0.03	0.63	0.02	0.29	0.04
CinCECGt	0.98	0.02	0.97	0.02	0.98	0.02	0.97	0.03	0.96	0.02	0.95	0.03	0.96	0.02	0.95	0.03
Coffee	0.93	0.04	0.87	0.08	0.95	0.05	0.89	0.10	0.92	0.04	0.84	0.07	0.93	0.07	0.86	0.14
CricketX	0.19	0.03	0.12	0.03	0.21	0.04	0.14	0.04	0.29	0.03	0.23	0.04	0.32	0.06	0.26	0.06
CricketY	0.27	0.03	0.20	0.03	0.28	0.02	0.21	0.02	0.38	0.01	0.33	0.01	0.39	0.03	0.33	0.03
CricketZ	0.20	0.03	0.13	0.03	0.21	0.05	0.14	0.06	0.36	0.06	0.30	0.06	0.36	0.04	0.30	0.05
ECG	0.99	0.00	0.97	0.01	0.99	0.00	0.98	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01
ECGFiveD	1.00	0.00	0.99	0.01	0.99	0.00	0.98	0.01	0.92	0.03	0.84	0.06	0.92	0.05	0.84	0.09
FaceAll	0.51	0.04	0.46	0.05	0.56	0.06	0.51	0.06	0.68	0.05	0.64	0.05	0.70	0.08	0.67	0.08
FacesUCR	0.50	0.03	0.46	0.03	0.55	0.03	0.51	0.03	0.67	0.04	0.64	0.05	0.69	0.04	0.66	0.04
Fish	0.49	0.09	0.41	0.11	0.49	0.06	0.40	0.06	0.48	0.09	0.39	0.10	0.46	0.07	0.37	0.08
GunPoint	0.89	0.07	0.78	0.15	0.89	0.09	0.78	0.18	0.90	0.05	0.80	0.10	0.90	0.06	0.81	0.11
Haptics	0.32	0.03	0.15	0.04	0.30	0.07	0.12	0.09	0.31	0.06	0.14	0.08	0.29	0.06	0.11	0.08
InlineS	0.23	0.01	0.09	0.02	0.25	0.03	0.12	0.04	0.26	0.04	0.13	0.04	0.26	0.06	0.13	0.08
Italy	0.97	0.00	0.95	0.01	0.97	0.01	0.93	0.02	0.97	0.01	0.94	0.01	0.97	0.01	0.93	0.02
Lighting2	0.57	0.15	0.13	0.21	0.56	0.11	0.10	0.10	0.63	0.10	0.23	0.20	0.67	0.07	0.29	0.20
Lighting7	0.31	0.11	0.16	0.11	0.32	0.11	0.18	0.11	0.40	0.05	0.29	0.06	0.47	0.09	0.37	0.11
MALLAT	0.86	0.02	0.84	0.02	0.88	0.02	0.86	0.02	0.83	0.03	0.81	0.04	0.84	0.04	0.82	0.04
MedicalI	0.49	0.03	0.27	0.03	0.49	0.01	0.26	0.03	0.51	0.02	0.28	0.02	0.53	0.03	0.29	0.03
MoteStrain	0.90	0.01	0.79	0.02	0.90	0.02	0.80	0.04	0.94	0.01	0.89	0.03	0.95	0.02	0.89	0.03
Olive	0.77	0.04	0.68	0.05	0.83	0.16	0.77	0.21	0.81	0.06	0.72	0.08	0.82	0.12	0.74	0.17
OSULeaf	0.29	0.06	0.14	0.08	0.27	0.06	0.11	0.08	0.32	0.05	0.17	0.06	0.34	0.02	0.19	0.02
Sony	0.97	0.02	0.94	0.03	0.98	0.02	0.95	0.05	0.96	0.03	0.92	0.06	0.96	0.02	0.93	0.04
SonyII	0.95	0.02	0.90	0.04	0.96	0.02	0.91	0.05	0.94	0.01	0.87	0.02	0.94	0.02	0.87	0.04
StarLightC	0.84	0.17	0.76	0.24	0.84	0.17	0.76	0.24	0.88	0.01	0.78	0.02	0.88	0.01	0.79	0.03
Synthetic	0.65	0.03	0.59	0.03	0.70	0.05	0.63	0.06	0.68	0.06	0.62	0.07	0.70	0.04	0.64	0.05
Trace	0.58	0.08	0.44	0.10	0.60	0.06	0.46	0.09	0.70	0.04	0.60	0.06	0.74	0.04	0.66	0.05
Two	0.85	0.02	0.80	0.03	0.84	0.01	0.79	0.02	0.99	0.00	0.98	0.01	0.99	0.01	0.99	0.01
TwoLeadE	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.01	0.98	0.01	0.97	0.01	0.99	0.01	0.97	0.02
uWaveGLX	0.70	0.01	0.66	0.02	0.71	0.02	0.67	0.03	0.72	0.01	0.68	0.02	0.72	0.03	0.68	0.03
uWaveGLY	0.63	0.02	0.58	0.02	0.63	0.01	0.58	0.02	0.63	0.01	0.58	0.02	0.64	0.01	0.58	0.01
uWaveGLZ	0.66	0.01	0.61	0.01	0.66	0.01	0.62	0.01	0.67	0.04	0.62	0.04	0.67	0.05	0.62	0.05
Wafer	0.99	0.00	0.97	0.01	1.00	0.00	0.97	0.01	0.98	0.00	0.92	0.01	0.99	0.00	0.92	0.02
Yoga	0.89	0.01	0.78	0.02	0.89	0.02	0.77	0.03	0.80	0.01	0.59	0.02	0.80	0.02	0.60	0.04

Tabla B.17: Resultados del aprendizaje inductivo y transductivo con el método Self-training-SVM en el 20% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.99	0.01	0.98	0.01	0.98	0.02	0.98	0.02	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.02
Chlorine	0.74	0.03	0.54	0.04	0.74	0.02	0.54	0.03	0.70	0.05	0.44	0.10	0.69	0.04	0.44	0.10
CinCECGt	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.01	0.97	0.02	0.96	0.03	0.98	0.02	0.97	0.03
Coffee	0.93	0.05	0.87	0.10	0.97	0.05	0.93	0.10	0.93	0.04	0.86	0.08	0.96	0.05	0.92	0.10
CricketX	0.23	0.04	0.16	0.04	0.26	0.05	0.19	0.05	0.37	0.02	0.31	0.02	0.42	0.03	0.36	0.04
CricketY	0.29	0.02	0.22	0.03	0.30	0.03	0.23	0.03	0.40	0.03	0.35	0.03	0.40	0.03	0.35	0.03
CricketZ	0.24	0.03	0.17	0.03	0.25	0.04	0.18	0.05	0.39	0.02	0.34	0.02	0.43	0.03	0.38	0.03
ECG	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.01	0.99	0.00	0.99	0.01	1.00	0.00	0.99	0.01
ECGFiveD	0.99	0.00	0.99	0.01	0.99	0.01	0.99	0.02	0.97	0.02	0.94	0.03	0.97	0.02	0.94	0.04
FaceAll	0.62	0.03	0.58	0.04	0.67	0.04	0.64	0.04	0.72	0.03	0.69	0.03	0.76	0.04	0.74	0.04
FacesUCR	0.61	0.04	0.57	0.05	0.67	0.03	0.64	0.03	0.74	0.03	0.72	0.03	0.78	0.01	0.75	0.01
Fish	0.62	0.04	0.55	0.04	0.65	0.06	0.60	0.07	0.57	0.10	0.50	0.11	0.59	0.09	0.53	0.10
GunPoint	0.93	0.04	0.86	0.09	0.94	0.04	0.89	0.07	0.91	0.08	0.83	0.15	0.94	0.05	0.89	0.11
Haptics	0.36	0.02	0.20	0.02	0.43	0.07	0.28	0.08	0.31	0.07	0.14	0.09	0.33	0.08	0.16	0.11
InlineS	0.24	0.03	0.11	0.04	0.28	0.01	0.15	0.01	0.31	0.02	0.19	0.02	0.32	0.06	0.20	0.08
Italy	0.97	0.01	0.94	0.01	0.97	0.01	0.95	0.01	0.96	0.01	0.93	0.01	0.97	0.01	0.93	0.01
Lighting2	0.59	0.09	0.14	0.17	0.60	0.11	0.12	0.25	0.70	0.10	0.37	0.19	0.62	0.07	0.19	0.20
Lighting7	0.38	0.07	0.25	0.09	0.45	0.09	0.33	0.11	0.45	0.07	0.34	0.09	0.50	0.08	0.40	0.11
MALLAT	0.86	0.04	0.85	0.05	0.88	0.03	0.87	0.04	0.86	0.05	0.84	0.05	0.88	0.05	0.86	0.05
MedicalI	0.56	0.01	0.34	0.03	0.59	0.03	0.37	0.04	0.55	0.02	0.32	0.03	0.56	0.05	0.33	0.07
MoteStrain	0.92	0.01	0.83	0.02	0.92	0.02	0.85	0.04	0.97	0.01	0.94	0.01	0.97	0.01	0.94	0.03
Olive	0.81	0.13	0.73	0.15	0.82	0.16	0.74	0.22	0.81	0.07	0.72	0.10	0.87	0.07	0.81	0.11
OSULeaf	0.32	0.08	0.17	0.10	0.37	0.11	0.23	0.14	0.42	0.05	0.30	0.06	0.46	0.05	0.35	0.06
Sony	0.99	0.00	0.98	0.01	0.99	0.01	0.98	0.02	0.99	0.00	0.98	0.01	0.99	0.01	0.98	0.02
SonyII	0.97	0.01	0.94	0.01	0.97	0.01	0.95	0.02	0.95	0.03	0.89	0.06	0.94	0.04	0.87	0.08
StarLightC	0.96	0.01	0.93	0.01	0.96	0.01	0.93	0.01	0.88	0.01	0.79	0.02	0.88	0.01	0.79	0.01
Synthetic	0.66	0.06	0.59	0.07	0.72	0.07	0.66	0.08	0.75	0.06	0.70	0.07	0.79	0.04	0.75	0.04
Trace	0.59	0.05	0.46	0.07	0.60	0.06	0.47	0.08	0.80	0.07	0.73	0.09	0.82	0.05	0.75	0.07
Two	0.90	0.01	0.87	0.01	0.91	0.02	0.88	0.03	0.99	0.01	0.99	0.01	0.99	0.00	0.99	0.00
TwoLeadE	1.00	0.00	0.99	0.00	1.00	0.00	0.99	0.00	0.98	0.01	0.96	0.02	0.98	0.01	0.95	0.02
uWaveGLX	0.73	0.02	0.69	0.02	0.73	0.02	0.69	0.02	0.75	0.02	0.71	0.02	0.74	0.02	0.70	0.02
uWaveGLY	0.65	0.01	0.60	0.01	0.65	0.02	0.60	0.02	0.67	0.01	0.62	0.02	0.67	0.01	0.62	0.01
uWaveGLZ	0.70	0.02	0.65	0.02	0.70	0.01	0.66	0.01	0.71	0.02	0.67	0.02	0.71	0.03	0.67	0.03
Wafer	1.00	0.00	0.98	0.01	1.00	0.00	0.98	0.01	0.99	0.00	0.94	0.01	0.99	0.00	0.94	0.01
Yoga	0.93	0.01	0.86	0.03	0.93	0.02	0.86	0.04	0.80	0.01	0.61	0.02	0.79	0.01	0.59	0.03

Tabla B.18: Resultados del aprendizaje inductivo y transductivo con el método Self-training-SVM en el 30% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.93	0.03	0.90	0.05	0.93	0.05	0.90	0.07	0.97	0.03	0.95	0.04	0.96	0.04	0.94	0.06
Chlorine	0.56	0.06	0.17	0.12	0.56	0.06	0.17	0.12	0.60	0.03	0.19	0.08	0.59	0.03	0.19	0.08
CinCECGt	0.98	0.01	0.98	0.01	0.98	0.01	0.97	0.02	0.99	0.01	0.99	0.01	0.99	0.01	0.98	0.01
Coffee	0.80	0.23	0.61	0.44	0.80	0.22	0.61	0.41	0.75	0.23	0.50	0.45	0.76	0.26	0.53	0.49
CricketX	0.25	0.04	0.19	0.04	0.26	0.03	0.20	0.03	0.44	0.03	0.39	0.04	0.43	0.04	0.37	0.04
CricketY	0.28	0.04	0.21	0.04	0.30	0.04	0.23	0.05	0.38	0.04	0.32	0.04	0.40	0.02	0.35	0.03
CricketZ	0.23	0.08	0.16	0.09	0.25	0.09	0.18	0.10	0.36	0.05	0.30	0.05	0.36	0.05	0.30	0.05
ECG	0.99	0.01	0.96	0.02	0.99	0.01	0.97	0.01	0.98	0.01	0.95	0.04	0.99	0.01	0.96	0.03
ECGFiveD	0.97	0.01	0.94	0.02	0.97	0.02	0.94	0.04	0.92	0.03	0.85	0.06	0.94	0.03	0.87	0.05
FaceAll	0.70	0.03	0.67	0.03	0.70	0.04	0.67	0.04	0.88	0.04	0.86	0.04	0.88	0.04	0.87	0.05
FacesUCR	0.65	0.06	0.62	0.07	0.66	0.08	0.62	0.10	0.87	0.06	0.86	0.07	0.88	0.05	0.86	0.06
Fish	0.46	0.12	0.37	0.14	0.49	0.15	0.40	0.17	0.50	0.14	0.42	0.16	0.53	0.14	0.45	0.17
GunPoint	0.72	0.11	0.44	0.22	0.69	0.14	0.38	0.27	0.75	0.09	0.51	0.18	0.75	0.09	0.50	0.18
Haptics	0.28	0.09	0.10	0.12	0.29	0.06	0.10	0.07	0.28	0.05	0.11	0.06	0.32	0.07	0.14	0.09
InlineS	0.21	0.04	0.07	0.06	0.22	0.06	0.08	0.08	0.25	0.04	0.11	0.05	0.25	0.06	0.10	0.08
Italy	0.96	0.02	0.92	0.05	0.96	0.02	0.93	0.04	0.95	0.02	0.90	0.03	0.94	0.01	0.89	0.03
Lighting2	0.60	0.12	0.12	0.15	0.58	0.11	0.06	0.11	0.61	0.11	0.13	0.15	0.56	0.14	0.02	0.14
Lighting7	0.47	0.04	0.37	0.04	0.46	0.05	0.36	0.05	0.45	0.20	0.35	0.21	0.47	0.22	0.38	0.24
MALLAT	0.98	0.00	0.98	0.00	0.98	0.01	0.97	0.01	0.98	0.00	0.98	0.00	0.98	0.01	0.98	0.01
MedicalI	0.60	0.02	0.37	0.03	0.59	0.03	0.36	0.03	0.57	0.02	0.34	0.04	0.58	0.04	0.35	0.05
MoteStrain	0.89	0.02	0.78	0.04	0.90	0.02	0.81	0.03	0.95	0.01	0.91	0.02	0.96	0.02	0.92	0.03
Olive	0.86	0.04	0.79	0.06	0.82	0.07	0.74	0.09	0.84	0.06	0.76	0.08	0.80	0.10	0.71	0.13
OSULeaf	0.29	0.08	0.14	0.10	0.28	0.10	0.12	0.12	0.37	0.04	0.24	0.05	0.36	0.06	0.22	0.07
Sony	0.96	0.02	0.93	0.04	0.98	0.01	0.95	0.03	0.97	0.02	0.94	0.05	0.98	0.02	0.96	0.03
SonyII	0.92	0.03	0.83	0.06	0.92	0.03	0.83	0.06	0.89	0.04	0.78	0.07	0.89	0.04	0.77	0.07
StarLightC	0.95	0.00	0.92	0.00	0.95	0.01	0.92	0.01	0.90	0.01	0.81	0.02	0.90	0.01	0.82	0.02
Synthetic	0.75	0.23	0.70	0.28	0.76	0.22	0.71	0.26	0.79	0.13	0.75	0.16	0.79	0.12	0.75	0.15
Trace	0.47	0.13	0.31	0.16	0.44	0.13	0.25	0.17	0.68	0.07	0.58	0.10	0.65	0.07	0.53	0.10
Two	0.78	0.02	0.71	0.03	0.78	0.03	0.70	0.03	0.97	0.02	0.96	0.02	0.97	0.02	0.96	0.02
TwoLeadE	0.97	0.02	0.94	0.04	0.98	0.01	0.95	0.03	0.98	0.02	0.95	0.04	0.97	0.02	0.95	0.04
uWaveGLX	0.70	0.02	0.65	0.02	0.70	0.01	0.66	0.02	0.74	0.01	0.71	0.02	0.74	0.03	0.71	0.03
uWaveGLY	0.66	0.02	0.61	0.02	0.66	0.03	0.61	0.03	0.67	0.01	0.62	0.01	0.67	0.02	0.62	0.03
uWaveGLZ	0.65	0.03	0.60	0.03	0.65	0.03	0.60	0.04	0.64	0.06	0.59	0.06	0.64	0.05	0.59	0.06
Wafer	0.99	0.00	0.96	0.02	0.99	0.00	0.96	0.01	0.99	0.00	0.92	0.01	0.99	0.00	0.93	0.03
Yoga	0.83	0.01	0.65	0.03	0.82	0.01	0.64	0.03	0.77	0.01	0.54	0.02	0.77	0.02	0.53	0.05

Tabla B.19: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-SVM en el 10% de ejemplos etiquetados.



Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.98	0.02	0.96	0.03	0.98	0.02	0.96	0.04	0.99	0.02	0.99	0.02	1.00	0.00	1.00	0.00
Chlorine	0.59	0.04	0.18	0.09	0.59	0.03	0.17	0.08	0.63	0.04	0.28	0.12	0.63	0.05	0.27	0.13
CinCECGt	0.98	0.01	0.97	0.02	0.98	0.02	0.97	0.03	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01
Coffee	0.93	0.07	0.85	0.15	0.97	0.05	0.93	0.10	0.90	0.09	0.81	0.19	0.97	0.05	0.93	0.10
CricketX	0.29	0.09	0.22	0.09	0.30	0.08	0.24	0.09	0.46	0.09	0.41	0.10	0.47	0.11	0.43	0.12
CricketY	0.36	0.08	0.30	0.09	0.35	0.09	0.29	0.10	0.50	0.02	0.45	0.02	0.52	0.03	0.47	0.03
CricketZ	0.27	0.09	0.21	0.10	0.27	0.09	0.20	0.09	0.45	0.06	0.41	0.06	0.45	0.05	0.40	0.05
ECG	0.99	0.00	0.97	0.01	0.99	0.00	0.98	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.98	0.01
ECGFiveD	0.99	0.00	0.99	0.01	0.99	0.01	0.98	0.02	0.96	0.02	0.91	0.05	0.95	0.02	0.91	0.04
FaceAll	0.80	0.03	0.78	0.03	0.81	0.01	0.79	0.02	0.92	0.02	0.91	0.02	0.92	0.01	0.92	0.01
FacesUCR	0.74	0.05	0.71	0.05	0.74	0.05	0.72	0.06	0.93	0.02	0.92	0.02	0.92	0.02	0.91	0.02
Fish	0.64	0.06	0.58	0.08	0.64	0.02	0.58	0.02	0.62	0.10	0.56	0.11	0.58	0.08	0.51	0.10
GunPoint	0.81	0.10	0.62	0.20	0.76	0.09	0.53	0.19	0.89	0.07	0.79	0.14	0.86	0.10	0.72	0.20
Haptics	0.31	0.08	0.13	0.11	0.31	0.08	0.12	0.11	0.29	0.08	0.11	0.11	0.31	0.08	0.13	0.10
InlineS	0.25	0.05	0.11	0.05	0.23	0.04	0.09	0.05	0.31	0.04	0.19	0.05	0.33	0.05	0.21	0.06
Italy	0.97	0.01	0.94	0.01	0.96	0.01	0.93	0.02	0.97	0.00	0.94	0.01	0.96	0.01	0.92	0.03
Lighting2	0.61	0.16	0.18	0.23	0.59	0.14	0.14	0.22	0.71	0.09	0.39	0.15	0.71	0.06	0.38	0.13
Lighting7	0.29	0.19	0.16	0.18	0.31	0.22	0.18	0.23	0.62	0.06	0.54	0.07	0.60	0.06	0.51	0.08
MALLAT	0.98	0.00	0.98	0.00	0.98	0.01	0.98	0.01	0.99	0.01	0.98	0.01	0.98	0.01	0.98	0.01
MedicalI	0.65	0.02	0.42	0.04	0.63	0.05	0.41	0.08	0.61	0.01	0.39	0.01	0.61	0.04	0.39	0.05
MoteStrain	0.90	0.01	0.81	0.02	0.91	0.01	0.81	0.03	0.95	0.01	0.89	0.02	0.95	0.02	0.90	0.04
Olive	0.79	0.09	0.69	0.13	0.78	0.14	0.69	0.20	0.81	0.09	0.73	0.12	0.82	0.11	0.74	0.15
OSULeaf	0.34	0.15	0.20	0.18	0.33	0.13	0.18	0.17	0.46	0.15	0.34	0.18	0.46	0.14	0.34	0.18
Sony	0.98	0.01	0.96	0.03	0.98	0.01	0.97	0.03	0.97	0.02	0.95	0.05	0.98	0.02	0.95	0.04
SonyII	0.96	0.01	0.91	0.03	0.96	0.00	0.91	0.01	0.94	0.02	0.86	0.04	0.93	0.01	0.85	0.03
StarLightC	0.96	0.01	0.93	0.01	0.96	0.01	0.93	0.02	0.88	0.01	0.78	0.02	0.88	0.01	0.78	0.01
Synthetic	0.92	0.02	0.91	0.03	0.94	0.02	0.93	0.03	0.93	0.03	0.92	0.03	0.95	0.02	0.94	0.03
Trace	0.53	0.07	0.38	0.08	0.58	0.07	0.45	0.10	0.74	0.03	0.65	0.03	0.74	0.04	0.65	0.05
Two	0.88	0.01	0.84	0.01	0.87	0.00	0.83	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
TwoLeadE	1.00	0.00	1.00	0.00	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.00	0.99	0.00	0.99	0.01
uWaveGLX	0.76	0.01	0.72	0.01	0.76	0.01	0.73	0.02	0.79	0.01	0.76	0.01	0.78	0.01	0.75	0.01
uWaveGLY	0.68	0.01	0.64	0.01	0.68	0.01	0.63	0.02	0.70	0.01	0.66	0.01	0.70	0.01	0.65	0.01
uWaveGLZ	0.70	0.01	0.66	0.01	0.70	0.01	0.66	0.01	0.70	0.02	0.65	0.02	0.70	0.03	0.66	0.03
Wafer	0.99	0.00	0.97	0.01	0.99	0.00	0.97	0.01	0.99	0.00	0.92	0.01	0.98	0.00	0.92	0.02
Yoga	0.89	0.01	0.78	0.01	0.88	0.01	0.76	0.02	0.80	0.01	0.59	0.03	0.78	0.01	0.56	0.03

Tabla B.20: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-SVM en el 20% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.99	0.01	0.98	0.02	0.98	0.02	0.97	0.04	1.00	0.01	0.99	0.01	0.99	0.01	0.99	0.02
Chlorine	0.71	0.08	0.46	0.20	0.72	0.08	0.47	0.20	0.68	0.06	0.38	0.16	0.69	0.07	0.41	0.17
CinCECGt	0.99	0.00	0.99	0.00	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	1.00	0.00	0.99	0.01
Coffee	0.97	0.04	0.93	0.08	0.97	0.05	0.93	0.09	0.98	0.03	0.96	0.06	0.98	0.04	0.97	0.07
CricketX	0.32	0.12	0.26	0.13	0.33	0.13	0.27	0.14	0.54	0.06	0.50	0.07	0.57	0.07	0.53	0.08
CricketY	0.40	0.06	0.35	0.06	0.38	0.06	0.33	0.06	0.53	0.04	0.49	0.04	0.51	0.02	0.46	0.02
CricketZ	0.42	0.05	0.37	0.05	0.41	0.04	0.36	0.04	0.56	0.04	0.52	0.04	0.57	0.07	0.54	0.07
ECG	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.01	1.00	0.00	0.99	0.01
ECGFiveD	1.00	0.00	1.00	0.00	1.00	0.01	0.99	0.01	0.97	0.02	0.94	0.03	0.97	0.01	0.94	0.02
FaceAll	0.84	0.03	0.83	0.03	0.85	0.02	0.83	0.03	0.94	0.02	0.93	0.02	0.94	0.03	0.93	0.03
FacesUCR	0.83	0.06	0.82	0.06	0.82	0.04	0.81	0.05	0.93	0.01	0.92	0.01	0.93	0.02	0.92	0.02
Fish	0.71	0.09	0.66	0.10	0.73	0.07	0.68	0.08	0.66	0.06	0.61	0.07	0.65	0.08	0.59	0.09
GunPoint	0.87	0.12	0.75	0.23	0.88	0.10	0.75	0.21	0.96	0.05	0.91	0.09	0.98	0.02	0.96	0.04
Haptics	0.38	0.07	0.22	0.08	0.38	0.08	0.22	0.10	0.32	0.07	0.14	0.09	0.33	0.08	0.15	0.10
InlineS	0.27	0.06	0.14	0.07	0.26	0.05	0.13	0.06	0.34	0.07	0.22	0.08	0.34	0.07	0.22	0.09
Italy	0.97	0.00	0.94	0.00	0.97	0.01	0.94	0.02	0.96	0.01	0.93	0.01	0.96	0.01	0.93	0.03
Lighting2	0.67	0.08	0.29	0.12	0.65	0.09	0.23	0.20	0.72	0.14	0.40	0.25	0.63	0.09	0.20	0.19
Lighting7	0.46	0.14	0.34	0.14	0.48	0.09	0.36	0.11	0.64	0.05	0.56	0.06	0.67	0.08	0.59	0.10
MALLAT	0.98	0.00	0.98	0.00	0.98	0.01	0.98	0.01	0.99	0.00	0.99	0.00	0.99	0.01	0.99	0.01
MedicalI	0.67	0.02	0.47	0.04	0.66	0.03	0.47	0.06	0.65	0.02	0.44	0.02	0.65	0.05	0.44	0.07
MoteStrain	0.92	0.01	0.84	0.01	0.93	0.02	0.86	0.03	0.96	0.01	0.93	0.02	0.97	0.01	0.94	0.01
Olive	0.84	0.03	0.77	0.04	0.85	0.04	0.78	0.05	0.83	0.08	0.75	0.13	0.85	0.12	0.78	0.18
OSULeaf	0.44	0.12	0.32	0.15	0.44	0.10	0.31	0.12	0.48	0.10	0.36	0.12	0.48	0.08	0.35	0.10
Sony	0.99	0.01	0.98	0.02	0.99	0.01	0.99	0.01	0.99	0.00	0.98	0.01	0.99	0.01	0.98	0.03
SonyII	0.97	0.01	0.93	0.01	0.97	0.02	0.93	0.04	0.95	0.01	0.90	0.02	0.95	0.01	0.89	0.03
StarLightC	0.96	0.00	0.93	0.01	0.96	0.01	0.94	0.01	0.88	0.01	0.78	0.02	0.88	0.02	0.79	0.03
Synthetic	0.94	0.03	0.93	0.03	0.95	0.03	0.94	0.03	0.97	0.01	0.97	0.02	0.97	0.04	0.97	0.04
Trace	0.63	0.05	0.51	0.06	0.66	0.07	0.55	0.09	0.78	0.05	0.71	0.06	0.77	0.11	0.69	0.14
Two	0.92	0.01	0.90	0.01	0.92	0.01	0.90	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
TwoLeadE	1.00	0.00	0.99	0.00	1.00	0.01	0.99	0.01	0.99	0.00	0.98	0.01	0.99	0.01	0.98	0.01
uWaveGLX	0.79	0.01	0.75	0.01	0.78	0.01	0.75	0.01	0.80	0.01	0.77	0.01	0.80	0.01	0.77	0.02
uWaveGLY	0.72	0.01	0.68	0.01	0.72	0.01	0.68	0.01	0.73	0.01	0.69	0.01	0.72	0.01	0.68	0.02
uWaveGLZ	0.72	0.01	0.68	0.01	0.72	0.02	0.68	0.02	0.73	0.02	0.69	0.02	0.73	0.02	0.69	0.02
Wafer	1.00	0.00	0.99	0.00	1.00	0.00	0.98	0.00	0.99	0.00	0.94	0.01	0.99	0.00	0.95	0.01
Yoga	0.92	0.01	0.85	0.03	0.92	0.01	0.84	0.02	0.80	0.01	0.60	0.01	0.79	0.02	0.58	0.03

Tabla B.21: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-SVM en el 30% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.69	0.06	0.53	0.09	0.67	0.06	0.51	0.09	0.82	0.01	0.74	0.02	0.83	0.03	0.75	0.04
Chlorine	0.56	0.01	0.27	0.02	0.54	0.01	0.24	0.03	0.52	0.01	0.20	0.03	0.50	0.02	0.18	0.03
CinCECGt	0.73	0.04	0.64	0.05	0.74	0.02	0.66	0.03	0.83	0.04	0.77	0.05	0.84	0.04	0.79	0.05
Coffee	0.89	0.08	0.78	0.16	0.93	0.08	0.85	0.16	0.89	0.09	0.79	0.18	0.95	0.07	0.90	0.15
CricketX	0.18	0.02	0.11	0.02	0.20	0.04	0.13	0.05	0.23	0.03	0.16	0.03	0.22	0.03	0.15	0.03
CricketY	0.22	0.03	0.15	0.04	0.21	0.01	0.14	0.01	0.26	0.03	0.20	0.03	0.26	0.02	0.20	0.03
CricketZ	0.18	0.03	0.11	0.03	0.20	0.02	0.13	0.02	0.22	0.03	0.15	0.04	0.25	0.02	0.18	0.02
ECG	0.93	0.04	0.79	0.12	0.93	0.03	0.80	0.10	0.92	0.03	0.78	0.09	0.93	0.04	0.79	0.13
ECGFiveD	0.90	0.02	0.80	0.03	0.90	0.03	0.81	0.07	0.80	0.02	0.60	0.04	0.80	0.05	0.61	0.09
FaceAll	0.46	0.05	0.40	0.05	0.45	0.05	0.40	0.05	0.59	0.05	0.55	0.06	0.57	0.05	0.53	0.05
FacesUCR	0.45	0.03	0.40	0.03	0.47	0.02	0.42	0.03	0.56	0.02	0.52	0.02	0.57	0.02	0.53	0.02
Fish	0.48	0.07	0.39	0.08	0.49	0.11	0.40	0.13	0.55	0.08	0.48	0.10	0.53	0.05	0.45	0.06
GunPoint	0.65	0.09	0.30	0.18	0.67	0.13	0.34	0.26	0.80	0.03	0.60	0.07	0.82	0.08	0.63	0.16
Haptics	0.32	0.04	0.15	0.05	0.33	0.08	0.16	0.09	0.33	0.06	0.16	0.07	0.32	0.06	0.15	0.08
InlineS	0.24	0.02	0.11	0.02	0.24	0.05	0.11	0.06	0.24	0.01	0.11	0.01	0.24	0.01	0.11	0.02
Italy	0.96	0.01	0.91	0.02	0.97	0.01	0.94	0.03	0.95	0.02	0.90	0.04	0.95	0.02	0.89	0.04
Lighting2	0.56	0.09	0.11	0.20	0.58	0.09	0.11	0.23	0.60	0.10	0.16	0.21	0.60	0.07	0.17	0.15
Lighting7	0.25	0.03	0.12	0.03	0.27	0.10	0.16	0.12	0.32	0.03	0.21	0.03	0.32	0.12	0.20	0.13
MALLAT	0.92	0.03	0.91	0.03	0.92	0.03	0.91	0.04	0.92	0.02	0.91	0.02	0.93	0.02	0.92	0.02
MedicalI	0.50	0.01	0.30	0.01	0.49	0.03	0.29	0.04	0.50	0.02	0.29	0.02	0.48	0.03	0.28	0.06
MoteStrain	0.80	0.04	0.59	0.08	0.81	0.05	0.62	0.09	0.84	0.05	0.68	0.10	0.82	0.06	0.64	0.11
Olive	0.74	0.07	0.64	0.09	0.75	0.10	0.65	0.13	0.74	0.07	0.63	0.09	0.70	0.07	0.59	0.10
OSULeaf	0.30	0.04	0.15	0.05	0.29	0.07	0.14	0.08	0.31	0.03	0.16	0.04	0.31	0.07	0.16	0.08
Sony	0.86	0.04	0.73	0.09	0.87	0.04	0.74	0.09	0.86	0.08	0.70	0.16	0.88	0.05	0.76	0.10
SonyII	0.78	0.05	0.51	0.13	0.79	0.05	0.54	0.13	0.81	0.02	0.58	0.05	0.79	0.01	0.55	0.01
StarLightC	0.90	0.01	0.83	0.01	0.91	0.01	0.83	0.02	0.91	0.01	0.84	0.01	0.92	0.01	0.85	0.01
Synthetic	0.53	0.03	0.44	0.03	0.52	0.05	0.42	0.06	0.58	0.03	0.50	0.04	0.57	0.06	0.49	0.07
Trace	0.52	0.05	0.36	0.07	0.48	0.04	0.31	0.05	0.69	0.06	0.58	0.07	0.64	0.10	0.53	0.14
Two	0.37	0.01	0.16	0.01	0.37	0.01	0.16	0.02	0.55	0.02	0.41	0.03	0.55	0.02	0.40	0.03
TwoLeadE	0.86	0.05	0.71	0.10	0.86	0.03	0.72	0.05	0.86	0.02	0.71	0.05	0.86	0.01	0.71	0.03
uWaveGLX	0.56	0.03	0.50	0.03	0.56	0.01	0.50	0.02	0.58	0.02	0.52	0.02	0.58	0.03	0.52	0.03
uWaveGLY	0.51	0.02	0.44	0.02	0.50	0.03	0.43	0.03	0.54	0.02	0.47	0.02	0.53	0.02	0.46	0.02
uWaveGLZ	0.52	0.02	0.45	0.03	0.52	0.03	0.45	0.03	0.53	0.02	0.46	0.02	0.54	0.01	0.48	0.02
Wafer	0.92	0.01	0.35	0.09	0.92	0.01	0.36	0.08	0.93	0.02	0.50	0.17	0.93	0.01	0.54	0.13
Yoga	0.73	0.01	0.45	0.02	0.73	0.03	0.46	0.05	0.76	0.02	0.52	0.03	0.74	0.02	0.48	0.05

Tabla B.22: Resultados del aprendizaje inductivo y transductivo con el método Self-training-DT en el 10% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.74	0.04	0.61	0.06	0.73	0.04	0.59	0.06	0.87	0.05	0.80	0.08	0.86	0.03	0.79	0.05
Chlorine	0.65	0.03	0.42	0.05	0.66	0.03	0.44	0.04	0.59	0.01	0.33	0.02	0.59	0.03	0.32	0.05
CinCECGt	0.82	0.02	0.76	0.03	0.82	0.02	0.76	0.02	0.87	0.02	0.83	0.02	0.87	0.02	0.83	0.02
Coffee	0.93	0.05	0.87	0.09	0.91	0.12	0.81	0.25	0.88	0.07	0.76	0.13	0.93	0.04	0.86	0.08
CricketX	0.22	0.02	0.15	0.03	0.19	0.03	0.12	0.03	0.30	0.03	0.24	0.03	0.29	0.03	0.22	0.04
CricketY	0.23	0.02	0.16	0.03	0.23	0.03	0.16	0.04	0.32	0.03	0.26	0.03	0.32	0.04	0.25	0.04
CricketZ	0.22	0.02	0.15	0.03	0.18	0.01	0.10	0.01	0.31	0.02	0.25	0.02	0.29	0.03	0.22	0.04
ECG	0.92	0.05	0.77	0.16	0.93	0.04	0.80	0.12	0.96	0.01	0.88	0.03	0.96	0.01	0.89	0.03
ECGFiveD	0.93	0.04	0.85	0.07	0.94	0.03	0.87	0.07	0.86	0.02	0.71	0.05	0.84	0.04	0.69	0.07
FaceAll	0.54	0.02	0.50	0.02	0.54	0.03	0.50	0.03	0.63	0.04	0.59	0.04	0.63	0.03	0.60	0.03
FacesUCR	0.54	0.03	0.50	0.03	0.53	0.03	0.49	0.04	0.63	0.02	0.60	0.02	0.63	0.03	0.60	0.03
Fish	0.58	0.03	0.51	0.04	0.59	0.10	0.52	0.12	0.54	0.03	0.46	0.03	0.57	0.05	0.50	0.06
GunPoint	0.82	0.05	0.65	0.09	0.82	0.08	0.64	0.16	0.88	0.02	0.76	0.05	0.82	0.04	0.65	0.08
Haptics	0.33	0.04	0.16	0.05	0.34	0.03	0.17	0.03	0.33	0.02	0.17	0.03	0.31	0.09	0.14	0.12
InlineS	0.29	0.01	0.16	0.01	0.25	0.02	0.12	0.02	0.28	0.01	0.15	0.01	0.28	0.05	0.15	0.06
Italy	0.96	0.02	0.92	0.03	0.95	0.02	0.91	0.04	0.96	0.01	0.92	0.02	0.96	0.01	0.92	0.02
Lighting2	0.54	0.03	0.06	0.07	0.63	0.03	0.22	0.10	0.64	0.09	0.23	0.19	0.62	0.09	0.18	0.22
Lighting7	0.33	0.06	0.21	0.07	0.38	0.07	0.26	0.07	0.46	0.05	0.36	0.06	0.45	0.10	0.35	0.13
MALLAT	0.95	0.02	0.95	0.02	0.95	0.03	0.94	0.03	0.95	0.02	0.95	0.02	0.96	0.01	0.95	0.01
MedicalI	0.55	0.02	0.33	0.02	0.54	0.03	0.33	0.05	0.57	0.02	0.37	0.03	0.56	0.04	0.35	0.05
MoteStrain	0.82	0.01	0.64	0.01	0.84	0.02	0.67	0.04	0.83	0.04	0.65	0.07	0.83	0.05	0.66	0.10
Olive	0.66	0.08	0.52	0.11	0.65	0.09	0.51	0.11	0.73	0.12	0.62	0.16	0.72	0.05	0.61	0.06
OSULeaf	0.31	0.06	0.17	0.07	0.34	0.06	0.21	0.07	0.39	0.05	0.26	0.06	0.36	0.06	0.23	0.07
Sony	0.90	0.04	0.79	0.09	0.91	0.03	0.82	0.06	0.92	0.04	0.84	0.08	0.92	0.05	0.84	0.09
SonyII	0.81	0.04	0.57	0.11	0.84	0.05	0.64	0.13	0.82	0.05	0.61	0.11	0.84	0.03	0.65	0.07
StarLightC	0.91	0.01	0.84	0.02	0.91	0.01	0.84	0.02	0.93	0.01	0.88	0.01	0.93	0.01	0.88	0.02
Synthetic	0.56	0.05	0.48	0.07	0.53	0.04	0.44	0.05	0.64	0.04	0.56	0.04	0.64	0.05	0.56	0.06
Trace	0.56	0.05	0.41	0.07	0.62	0.09	0.49	0.11	0.77	0.07	0.69	0.09	0.76	0.03	0.69	0.04
Two	0.39	0.01	0.19	0.02	0.38	0.02	0.17	0.03	0.62	0.01	0.49	0.02	0.62	0.01	0.50	0.01
TwoLeadE	0.91	0.02	0.82	0.03	0.90	0.02	0.80	0.04	0.90	0.04	0.80	0.08	0.91	0.03	0.81	0.07
uWaveGLX	0.61	0.01	0.55	0.01	0.60	0.02	0.54	0.02	0.61	0.01	0.56	0.01	0.62	0.02	0.56	0.02
uWaveGLY	0.56	0.02	0.50	0.02	0.55	0.02	0.49	0.03	0.57	0.02	0.51	0.02	0.57	0.02	0.51	0.03
uWaveGLZ	0.56	0.01	0.50	0.01	0.55	0.01	0.49	0.02	0.57	0.00	0.51	0.00	0.58	0.02	0.52	0.02
Wafer	0.93	0.02	0.50	0.18	0.94	0.01	0.55	0.15	0.95	0.03	0.65	0.23	0.95	0.02	0.71	0.19
Yoga	0.79	0.01	0.57	0.03	0.78	0.01	0.56	0.02	0.79	0.02	0.57	0.04	0.78	0.01	0.56	0.02

Tabla B.23: Resultados del aprendizaje inductivo y transductivo con el método Self-training-DT en el 20% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.79	0.01	0.69	0.01	0.80	0.03	0.70	0.05	0.87	0.02	0.80	0.02	0.87	0.03	0.81	0.04
Chlorine	0.75	0.01	0.58	0.02	0.76	0.01	0.60	0.02	0.63	0.01	0.39	0.02	0.64	0.02	0.40	0.03
CinCECGt	0.84	0.02	0.78	0.03	0.85	0.04	0.81	0.05	0.87	0.04	0.83	0.06	0.88	0.03	0.83	0.04
Coffee	0.92	0.07	0.85	0.13	0.91	0.09	0.82	0.18	0.94	0.04	0.88	0.07	0.98	0.04	0.96	0.09
CricketX	0.23	0.01	0.16	0.02	0.22	0.05	0.15	0.05	0.34	0.01	0.28	0.01	0.33	0.06	0.27	0.06
CricketY	0.25	0.03	0.18	0.03	0.24	0.05	0.17	0.05	0.33	0.04	0.27	0.04	0.37	0.05	0.31	0.06
CricketZ	0.23	0.02	0.16	0.03	0.23	0.01	0.16	0.02	0.34	0.03	0.28	0.03	0.33	0.05	0.27	0.05
ECG	0.97	0.02	0.91	0.04	0.96	0.02	0.89	0.05	0.96	0.01	0.88	0.04	0.96	0.01	0.88	0.04
ECGFiveD	0.95	0.04	0.90	0.08	0.95	0.04	0.90	0.08	0.87	0.04	0.74	0.07	0.88	0.05	0.77	0.10
FaceAll	0.58	0.02	0.54	0.02	0.58	0.01	0.54	0.01	0.68	0.02	0.64	0.02	0.67	0.06	0.64	0.07
FacesUCR	0.58	0.03	0.54	0.03	0.58	0.03	0.54	0.03	0.65	0.04	0.61	0.04	0.65	0.04	0.61	0.04
Fish	0.64	0.04	0.58	0.04	0.61	0.05	0.55	0.06	0.64	0.02	0.58	0.03	0.63	0.06	0.57	0.07
GunPoint	0.82	0.10	0.64	0.20	0.82	0.07	0.63	0.14	0.91	0.04	0.81	0.07	0.91	0.04	0.82	0.08
Haptics	0.36	0.01	0.19	0.02	0.35	0.05	0.19	0.06	0.35	0.05	0.18	0.06	0.34	0.06	0.18	0.08
InlineS	0.33	0.02	0.22	0.02	0.32	0.04	0.20	0.05	0.35	0.04	0.23	0.05	0.34	0.01	0.22	0.01
Italy	0.97	0.01	0.93	0.01	0.97	0.01	0.93	0.03	0.96	0.01	0.91	0.03	0.95	0.02	0.91	0.04
Lighting2	0.58	0.09	0.16	0.15	0.62	0.07	0.23	0.16	0.68	0.05	0.34	0.08	0.69	0.08	0.35	0.17
Lighting7	0.36	0.05	0.24	0.06	0.35	0.10	0.22	0.12	0.51	0.03	0.41	0.04	0.43	0.10	0.32	0.11
MALLAT	0.96	0.01	0.95	0.01	0.96	0.01	0.95	0.01	0.96	0.01	0.96	0.01	0.96	0.01	0.96	0.01
MedicalI	0.59	0.03	0.40	0.04	0.58	0.02	0.38	0.05	0.60	0.01	0.42	0.02	0.58	0.02	0.39	0.04
MoteStrain	0.85	0.02	0.69	0.04	0.86	0.02	0.71	0.04	0.87	0.02	0.73	0.03	0.87	0.01	0.74	0.03
Olive	0.68	0.07	0.55	0.09	0.73	0.15	0.63	0.20	0.73	0.08	0.62	0.11	0.75	0.12	0.64	0.17
OSULeaf	0.36	0.07	0.22	0.09	0.38	0.04	0.25	0.04	0.41	0.05	0.29	0.06	0.44	0.04	0.32	0.05
Sony	0.90	0.06	0.80	0.11	0.91	0.03	0.82	0.07	0.94	0.04	0.88	0.07	0.95	0.02	0.90	0.04
SonyII	0.84	0.04	0.65	0.09	0.85	0.02	0.67	0.05	0.86	0.05	0.69	0.12	0.85	0.04	0.67	0.08
StarLightC	0.93	0.01	0.87	0.01	0.92	0.00	0.87	0.01	0.93	0.01	0.87	0.01	0.93	0.01	0.88	0.02
Synthetic	0.61	0.05	0.53	0.05	0.59	0.05	0.51	0.06	0.73	0.03	0.67	0.03	0.71	0.03	0.65	0.03
Trace	0.55	0.05	0.40	0.07	0.53	0.13	0.37	0.17	0.80	0.04	0.74	0.05	0.81	0.02	0.75	0.03
Two	0.40	0.00	0.20	0.00	0.41	0.01	0.22	0.02	0.66	0.01	0.54	0.01	0.65	0.01	0.54	0.02
TwoLeadE	0.93	0.01	0.86	0.03	0.93	0.02	0.87	0.04	0.92	0.02	0.85	0.04	0.90	0.01	0.81	0.02
uWaveGLX	0.63	0.01	0.58	0.01	0.62	0.02	0.57	0.02	0.64	0.01	0.59	0.01	0.65	0.02	0.60	0.03
uWaveGLY	0.58	0.01	0.52	0.01	0.58	0.02	0.52	0.03	0.59	0.01	0.54	0.01	0.59	0.01	0.54	0.01
uWaveGLZ	0.57	0.01	0.51	0.01	0.58	0.02	0.53	0.02	0.58	0.01	0.52	0.01	0.59	0.01	0.53	0.01
Wafer	0.95	0.01	0.64	0.09	0.95	0.01	0.63	0.10	0.96	0.02	0.75	0.17	0.96	0.02	0.74	0.16
Yoga	0.79	0.02	0.59	0.04	0.78	0.03	0.55	0.06	0.82	0.01	0.64	0.02	0.83	0.01	0.65	0.02

Tabla B.24: Resultados del aprendizaje inductivo y transductivo con el método Self-training-DT en el 30% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.75	0.03	0.63	0.05	0.77	0.05	0.65	0.08	0.85	0.05	0.77	0.07	0.84	0.06	0.75	0.09
Chlorine	0.59	0.01	0.31	0.02	0.58	0.03	0.31	0.03	0.56	0.02	0.25	0.02	0.54	0.04	0.24	0.04
CinCECGt	0.85	0.04	0.80	0.05	0.84	0.04	0.78	0.05	0.90	0.01	0.86	0.01	0.90	0.03	0.86	0.04
Coffee	0.94	0.06	0.88	0.13	0.93	0.08	0.85	0.16	0.93	0.03	0.87	0.07	0.94	0.09	0.89	0.18
CricketX	0.18	0.01	0.11	0.01	0.19	0.03	0.11	0.03	0.26	0.01	0.19	0.01	0.27	0.03	0.21	0.04
CricketY	0.22	0.02	0.15	0.03	0.20	0.03	0.13	0.03	0.30	0.03	0.24	0.03	0.28	0.05	0.22	0.05
CricketZ	0.18	0.01	0.11	0.01	0.19	0.02	0.12	0.02	0.26	0.04	0.19	0.04	0.26	0.05	0.19	0.05
ECG	0.94	0.02	0.85	0.05	0.94	0.02	0.85	0.06	0.95	0.01	0.87	0.03	0.95	0.01	0.87	0.04
ECGFiveD	0.93	0.03	0.86	0.05	0.92	0.05	0.83	0.10	0.88	0.03	0.76	0.07	0.88	0.03	0.76	0.06
FaceAll	0.52	0.03	0.47	0.04	0.50	0.03	0.46	0.03	0.69	0.03	0.66	0.03	0.69	0.03	0.65	0.03
FacesUCR	0.50	0.03	0.45	0.03	0.51	0.02	0.46	0.02	0.67	0.03	0.64	0.03	0.67	0.04	0.64	0.05
Fish	0.53	0.07	0.45	0.09	0.53	0.11	0.45	0.13	0.56	0.03	0.49	0.03	0.53	0.05	0.45	0.06
GunPoint	0.71	0.08	0.42	0.15	0.74	0.06	0.48	0.13	0.77	0.07	0.55	0.13	0.77	0.08	0.54	0.16
Haptics	0.31	0.01	0.14	0.01	0.33	0.05	0.17	0.07	0.32	0.03	0.15	0.04	0.33	0.04	0.16	0.05
InlineS	0.22	0.04	0.09	0.04	0.24	0.05	0.11	0.05	0.23	0.02	0.10	0.02	0.26	0.06	0.13	0.07
Italy	0.96	0.01	0.92	0.02	0.97	0.01	0.94	0.02	0.96	0.02	0.92	0.04	0.96	0.02	0.92	0.04
Lighting2	0.54	0.08	0.08	0.11	0.58	0.07	0.12	0.12	0.63	0.04	0.17	0.10	0.69	0.15	0.32	0.33
Lighting7	0.36	0.06	0.23	0.06	0.34	0.04	0.20	0.05	0.41	0.06	0.30	0.05	0.44	0.04	0.33	0.06
MALLAT	0.95	0.02	0.95	0.02	0.95	0.01	0.94	0.02	0.97	0.01	0.96	0.01	0.97	0.01	0.96	0.01
MedicalI	0.56	0.03	0.35	0.03	0.55	0.03	0.33	0.03	0.55	0.02	0.32	0.03	0.56	0.04	0.33	0.07
MoteStrain	0.86	0.02	0.72	0.05	0.86	0.01	0.72	0.03	0.89	0.01	0.77	0.02	0.86	0.01	0.72	0.03
Olive	0.84	0.07	0.77	0.09	0.80	0.07	0.72	0.10	0.81	0.04	0.72	0.05	0.77	0.11	0.67	0.15
OSULeaf	0.32	0.04	0.18	0.04	0.29	0.06	0.15	0.07	0.35	0.03	0.22	0.04	0.29	0.06	0.15	0.08
Sony	0.91	0.03	0.82	0.06	0.93	0.03	0.85	0.07	0.93	0.03	0.86	0.07	0.94	0.05	0.87	0.09
SonyII	0.85	0.05	0.67	0.11	0.85	0.04	0.67	0.11	0.86	0.04	0.71	0.09	0.86	0.04	0.69	0.10
StarLightC	0.93	0.00	0.88	0.01	0.93	0.01	0.87	0.02	0.93	0.01	0.88	0.01	0.93	0.01	0.88	0.01
Synthetic	0.52	0.02	0.43	0.02	0.51	0.03	0.42	0.04	0.64	0.07	0.56	0.08	0.62	0.05	0.55	0.06
Trace	0.57	0.03	0.42	0.04	0.52	0.04	0.37	0.06	0.70	0.06	0.60	0.08	0.66	0.09	0.55	0.12
Two	0.39	0.01	0.18	0.02	0.39	0.01	0.19	0.01	0.61	0.05	0.48	0.07	0.61	0.05	0.48	0.06
TwoLeadE	0.88	0.03	0.75	0.06	0.86	0.04	0.72	0.09	0.91	0.04	0.81	0.08	0.91	0.04	0.82	0.07
uWaveGLX	0.61	0.02	0.55	0.02	0.62	0.02	0.56	0.02	0.62	0.02	0.57	0.02	0.63	0.02	0.58	0.02
uWaveGLY	0.56	0.03	0.50	0.03	0.57	0.01	0.51	0.01	0.58	0.03	0.52	0.03	0.58	0.03	0.52	0.03
uWaveGLZ	0.55	0.02	0.49	0.02	0.55	0.02	0.48	0.02	0.57	0.01	0.51	0.01	0.57	0.02	0.51	0.02
Wafer	0.95	0.00	0.66	0.03	0.94	0.01	0.62	0.06	0.96	0.02	0.76	0.13	0.96	0.01	0.75	0.11
Yoga	0.76	0.03	0.52	0.05	0.76	0.01	0.51	0.02	0.79	0.02	0.58	0.04	0.80	0.02	0.59	0.04

Tabla B.25: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-DT en el 10% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.83	0.05	0.75	0.07	0.82	0.03	0.72	0.05	0.90	0.04	0.85	0.06	0.91	0.03	0.86	0.05
Chlorine	0.68	0.04	0.47	0.07	0.69	0.03	0.48	0.05	0.62	0.03	0.37	0.05	0.63	0.01	0.39	0.01
CinCECGt	0.91	0.04	0.88	0.05	0.92	0.05	0.89	0.06	0.94	0.02	0.92	0.02	0.93	0.03	0.91	0.03
Coffee	0.95	0.02	0.90	0.05	0.95	0.05	0.89	0.10	0.92	0.07	0.83	0.14	0.93	0.08	0.85	0.15
CricketX	0.21	0.02	0.14	0.02	0.19	0.05	0.12	0.05	0.31	0.02	0.25	0.02	0.32	0.04	0.25	0.04
CricketY	0.26	0.02	0.19	0.02	0.25	0.03	0.18	0.03	0.35	0.02	0.29	0.03	0.36	0.06	0.30	0.06
CricketZ	0.23	0.03	0.16	0.03	0.21	0.03	0.14	0.03	0.32	0.02	0.25	0.02	0.30	0.02	0.24	0.02
ECG	0.95	0.03	0.86	0.09	0.95	0.03	0.86	0.08	0.94	0.02	0.85	0.06	0.94	0.01	0.83	0.04
ECGFiveD	0.95	0.02	0.90	0.05	0.95	0.02	0.89	0.03	0.90	0.02	0.80	0.05	0.91	0.02	0.83	0.03
FaceAll	0.57	0.02	0.53	0.03	0.59	0.05	0.55	0.05	0.74	0.02	0.71	0.02	0.75	0.02	0.73	0.02
FacesUCR	0.59	0.02	0.56	0.02	0.60	0.05	0.56	0.05	0.74	0.03	0.72	0.03	0.73	0.04	0.71	0.04
Fish	0.62	0.08	0.55	0.09	0.62	0.06	0.56	0.08	0.61	0.05	0.54	0.06	0.58	0.04	0.51	0.05
GunPoint	0.78	0.06	0.57	0.12	0.78	0.09	0.56	0.17	0.86	0.04	0.72	0.08	0.82	0.08	0.63	0.16
Haptics	0.31	0.02	0.14	0.03	0.33	0.02	0.16	0.03	0.36	0.03	0.19	0.04	0.33	0.06	0.15	0.07
InlineS	0.29	0.02	0.17	0.02	0.27	0.02	0.14	0.02	0.28	0.02	0.15	0.02	0.28	0.05	0.15	0.06
Italy	0.97	0.01	0.93	0.02	0.96	0.01	0.92	0.02	0.96	0.01	0.91	0.03	0.96	0.01	0.92	0.02
Lighting2	0.63	0.10	0.24	0.17	0.62	0.12	0.21	0.19	0.67	0.08	0.27	0.19	0.66	0.10	0.24	0.28
Lighting7	0.41	0.03	0.27	0.04	0.37	0.10	0.23	0.13	0.51	0.08	0.41	0.09	0.47	0.14	0.35	0.16
MALLAT	0.97	0.00	0.96	0.00	0.97	0.01	0.97	0.01	0.97	0.01	0.97	0.01	0.97	0.01	0.96	0.01
MedicalI	0.58	0.03	0.35	0.03	0.56	0.04	0.32	0.06	0.60	0.02	0.40	0.03	0.59	0.03	0.37	0.04
MoteStrain	0.85	0.01	0.70	0.02	0.84	0.02	0.67	0.04	0.89	0.02	0.77	0.04	0.89	0.01	0.78	0.03
Olive	0.76	0.06	0.65	0.08	0.75	0.10	0.65	0.12	0.81	0.04	0.72	0.06	0.80	0.13	0.71	0.18
OSULeaf	0.36	0.07	0.23	0.08	0.38	0.08	0.25	0.09	0.40	0.04	0.28	0.05	0.41	0.05	0.29	0.06
Sony	0.93	0.03	0.86	0.07	0.94	0.01	0.88	0.03	0.91	0.03	0.82	0.06	0.90	0.05	0.80	0.09
SonyII	0.88	0.01	0.74	0.02	0.89	0.02	0.76	0.04	0.89	0.04	0.77	0.09	0.90	0.03	0.79	0.07
StarLightC	0.93	0.01	0.88	0.01	0.93	0.01	0.88	0.01	0.94	0.01	0.89	0.01	0.94	0.01	0.89	0.01
Synthetic	0.62	0.04	0.55	0.04	0.58	0.03	0.50	0.03	0.72	0.04	0.67	0.04	0.72	0.04	0.66	0.04
Trace	0.54	0.03	0.39	0.04	0.56	0.08	0.41	0.11	0.80	0.07	0.73	0.09	0.80	0.06	0.74	0.08
Two	0.42	0.02	0.22	0.03	0.41	0.01	0.21	0.02	0.70	0.03	0.61	0.04	0.70	0.03	0.60	0.04
TwoLeadE	0.95	0.03	0.89	0.06	0.94	0.04	0.88	0.07	0.92	0.03	0.85	0.06	0.92	0.03	0.84	0.06
uWaveGLX	0.64	0.02	0.58	0.02	0.64	0.02	0.59	0.02	0.65	0.02	0.60	0.02	0.64	0.02	0.59	0.03
uWaveGLY	0.58	0.01	0.52	0.01	0.58	0.02	0.52	0.02	0.59	0.01	0.53	0.02	0.59	0.01	0.53	0.02
uWaveGLZ	0.57	0.01	0.51	0.01	0.57	0.02	0.51	0.02	0.60	0.01	0.54	0.01	0.60	0.01	0.54	0.01
Wafer	0.96	0.00	0.73	0.03	0.96	0.01	0.73	0.07	0.97	0.01	0.79	0.07	0.97	0.01	0.80	0.05
Yoga	0.81	0.02	0.61	0.04	0.80	0.02	0.61	0.04	0.84	0.01	0.68	0.03	0.83	0.02	0.66	0.05

Tabla B.26: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-DT en el 20% de ejemplos etiquetados.

Conjuntos de datos	Euclidiana								DTW							
	Transductivo				Inductivo				Transductivo				Inductivo			
	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$	$\mu(E)$	$\sigma(E)$	$\mu(K)$	$\sigma(K)$
CBF	0.86	0.03	0.79	0.05	0.85	0.03	0.77	0.05	0.91	0.02	0.86	0.03	0.89	0.01	0.84	0.01
Chlorine	0.80	0.02	0.66	0.03	0.80	0.01	0.67	0.02	0.69	0.01	0.48	0.03	0.69	0.03	0.49	0.04
CinCECGt	0.94	0.01	0.92	0.01	0.94	0.02	0.91	0.02	0.94	0.02	0.92	0.03	0.94	0.01	0.92	0.01
Coffee	0.94	0.01	0.88	0.03	0.93	0.07	0.86	0.14	0.95	0.03	0.90	0.06	0.91	0.08	0.83	0.17
CricketX	0.25	0.02	0.18	0.02	0.24	0.05	0.17	0.05	0.37	0.02	0.31	0.02	0.36	0.03	0.30	0.03
CricketY	0.25	0.02	0.19	0.02	0.28	0.06	0.21	0.06	0.38	0.04	0.32	0.04	0.37	0.04	0.31	0.05
CricketZ	0.25	0.02	0.18	0.02	0.23	0.04	0.16	0.05	0.39	0.02	0.34	0.02	0.38	0.04	0.32	0.04
ECG	0.96	0.01	0.88	0.03	0.96	0.01	0.90	0.02	0.96	0.01	0.90	0.04	0.96	0.02	0.88	0.05
ECGFiveD	0.95	0.01	0.89	0.01	0.93	0.04	0.87	0.08	0.91	0.02	0.82	0.03	0.91	0.02	0.82	0.04
FaceAll	0.62	0.02	0.59	0.02	0.63	0.01	0.60	0.01	0.77	0.03	0.75	0.04	0.77	0.04	0.75	0.04
FacesUCR	0.63	0.02	0.60	0.02	0.63	0.02	0.60	0.02	0.76	0.02	0.74	0.03	0.76	0.03	0.74	0.03
Fish	0.64	0.03	0.58	0.04	0.65	0.06	0.60	0.07	0.69	0.02	0.64	0.03	0.68	0.05	0.62	0.06
GunPoint	0.81	0.06	0.62	0.12	0.78	0.08	0.55	0.16	0.92	0.04	0.84	0.07	0.91	0.04	0.82	0.08
Haptics	0.37	0.02	0.21	0.02	0.36	0.04	0.19	0.05	0.39	0.03	0.23	0.04	0.37	0.06	0.21	0.08
InlineS	0.35	0.02	0.24	0.02	0.37	0.03	0.26	0.04	0.36	0.03	0.25	0.04	0.37	0.04	0.25	0.05
Italy	0.97	0.00	0.94	0.01	0.97	0.01	0.94	0.02	0.96	0.01	0.92	0.02	0.95	0.02	0.91	0.03
Lighting2	0.59	0.07	0.19	0.08	0.64	0.09	0.27	0.16	0.77	0.06	0.50	0.15	0.69	0.04	0.35	0.07
Lighting7	0.40	0.03	0.27	0.04	0.35	0.08	0.20	0.09	0.54	0.06	0.45	0.08	0.56	0.09	0.47	0.11
MALLAT	0.97	0.00	0.97	0.00	0.97	0.01	0.96	0.01	0.97	0.01	0.97	0.01	0.97	0.01	0.97	0.01
MedicalI	0.61	0.02	0.43	0.04	0.62	0.03	0.42	0.06	0.64	0.02	0.45	0.04	0.65	0.03	0.46	0.05
MoteStrain	0.87	0.02	0.74	0.05	0.91	0.02	0.82	0.05	0.91	0.01	0.81	0.02	0.91	0.02	0.82	0.04
Olive	0.81	0.04	0.72	0.07	0.83	0.10	0.76	0.14	0.78	0.12	0.69	0.15	0.80	0.15	0.71	0.21
OSULeaf	0.40	0.03	0.28	0.04	0.40	0.04	0.29	0.05	0.43	0.04	0.32	0.04	0.39	0.05	0.26	0.05
Sony	0.96	0.01	0.92	0.02	0.95	0.04	0.89	0.08	0.93	0.03	0.86	0.06	0.93	0.04	0.85	0.07
SonyII	0.91	0.01	0.81	0.03	0.91	0.02	0.82	0.04	0.92	0.02	0.84	0.04	0.92	0.03	0.84	0.06
StarLightC	0.94	0.01	0.89	0.02	0.93	0.01	0.88	0.01	0.94	0.00	0.89	0.01	0.94	0.01	0.89	0.01
Synthetic	0.66	0.04	0.59	0.05	0.65	0.04	0.58	0.05	0.78	0.04	0.74	0.04	0.79	0.03	0.75	0.04
Trace	0.60	0.05	0.46	0.06	0.58	0.02	0.44	0.03	0.82	0.07	0.76	0.10	0.84	0.07	0.79	0.09
Two	0.42	0.02	0.22	0.02	0.42	0.02	0.22	0.03	0.75	0.05	0.67	0.06	0.76	0.03	0.68	0.04
TwoLeadE	0.93	0.02	0.87	0.04	0.94	0.02	0.87	0.03	0.93	0.02	0.86	0.04	0.94	0.02	0.89	0.04
uWaveGLX	0.65	0.02	0.60	0.02	0.65	0.02	0.60	0.02	0.67	0.01	0.62	0.01	0.66	0.02	0.61	0.03
uWaveGLY	0.60	0.01	0.54	0.01	0.60	0.01	0.54	0.01	0.61	0.02	0.55	0.02	0.61	0.02	0.55	0.02
uWaveGLZ	0.59	0.01	0.53	0.01	0.61	0.02	0.55	0.02	0.61	0.02	0.56	0.02	0.61	0.01	0.56	0.01
Wafer	0.96	0.00	0.72	0.02	0.95	0.00	0.72	0.02	0.97	0.01	0.79	0.07	0.97	0.01	0.79	0.08
Yoga	0.84	0.02	0.68	0.04	0.82	0.02	0.65	0.04	0.86	0.02	0.71	0.04	0.85	0.04	0.70	0.07

Tabla B.27: Resultados del aprendizaje inductivo y transductivo con el método Tri-training-DT en el 30% de ejemplos etiquetados.



## B.2 Resultados obtenidos con el 1NN como cota de referencia

La Tabla B.28 muestra los resultados obtenidos con el algoritmo 1NN mediante el esquema de validación cruzada de cinco particiones. Dicho algoritmo es entrenado a partir del subconjunto  $L$  de ejemplos etiquetados y evaluado en el conjunto de prueba. Estos resultados se utilizan para calcular la cota inferior de referencia (CIR) y posteriormente la ganancia de exactitud. La Tabla B.29 muestra los resultados obtenidos con el 1NN entrenado con el conjunto de entrenamiento en su totalidad ( $L \cup U$ ). Estos valores se utilizan para calcular la cota superior de referencia (CSR) y posteriormente la competitividad del paradigma semi-supervisado frente al supervisado.

Conjuntos de datos	Euclidiana			DTW		
	10%	20%	30%	10%	20%	30%
CBF	0.93	0.97	0.98	0.98	1.00	0.99
Chlorine	0.59	0.72	0.82	0.57	0.68	0.78
CinCECGt	0.98	0.99	1.00	0.98	0.99	1.00
Coffee	0.97	0.93	0.97	0.89	0.95	0.98
CricketX	0.35	0.43	0.51	0.48	0.57	0.67
CricketY	0.36	0.43	0.44	0.45	0.58	0.64
CricketZ	0.36	0.43	0.50	0.52	0.62	0.68
ECCG	0.99	0.99	1.00	0.98	0.99	1.00
ECCGFiveD	0.94	0.96	0.96	0.90	0.95	0.97
FaceAll	0.76	0.84	0.87	0.93	0.96	0.96
FacesUCR	0.74	0.82	0.88	0.91	0.96	0.97
Fish	0.59	0.67	0.73	0.63	0.67	0.77
GunPoint	0.73	0.82	0.90	0.84	0.89	0.96
Haptics	0.35	0.35	0.40	0.37	0.37	0.40
InlineS	0.28	0.32	0.39	0.33	0.40	0.43
Italy	0.96	0.96	0.96	0.93	0.95	0.95
Lighting2	0.61	0.63	0.62	0.70	0.72	0.79
Lighting7	0.45	0.53	0.54	0.58	0.68	0.68
MALLAT	0.96	0.97	0.97	0.98	0.99	0.98
MedicalI	0.59	0.63	0.69	0.59	0.67	0.73
MoteStrain	0.90	0.90	0.90	0.94	0.96	0.96
Olive	0.80	0.78	0.88	0.80	0.82	0.87
OSULeaf	0.42	0.49	0.55	0.47	0.55	0.61
Sony	0.93	0.95	0.96	0.90	0.92	0.95
SonyII	0.91	0.93	0.96	0.91	0.94	0.96
StarLightC	0.85	0.86	0.87	0.88	0.89	0.90
Synthetic	0.79	0.85	0.85	0.92	0.96	0.95
Trace	0.53	0.64	0.64	0.72	0.88	0.93
Two	0.77	0.87	0.92	0.97	0.99	1.00
TwoLeadE	0.90	0.97	0.98	0.97	0.99	0.99
uWaveGLX	0.71	0.73	0.74	0.73	0.75	0.77
uWaveGLY	0.63	0.65	0.67	0.66	0.68	0.69
uWaveGLZ	0.62	0.64	0.67	0.64	0.67	0.70
Wafer	0.99	1.00	1.00	0.99	0.99	1.00
Yoga	0.82	0.87	0.89	0.84	0.89	0.90

Tabla B.28: Resultados de la CIR evaluada mediante el algoritmo 1NN.

Conjuntos de datos	Euclidiana			DTW		
	10%	20%	30%	10%	20%	30%
CBF	0.99	0.99	0.99	1.00	1.00	1.00
Chlorine	0.99	0.99	0.99	0.97	0.97	0.97
CinCECGt	1.00	1.00	1.00	1.00	1.00	1.00
Coffee	1.00	1.00	1.00	1.00	1.00	1.00
CricketX	0.64	0.64	0.64	0.78	0.78	0.78
CricketY	0.60	0.60	0.60	0.76	0.76	0.76
CricketZ	0.63	0.63	0.63	0.75	0.75	0.75
ECG	1.00	1.00	1.00	1.00	1.00	1.00
ECGFiveD	1.00	1.00	1.00	0.99	0.99	0.99
FaceAll	0.96	0.96	0.96	0.99	0.99	0.99
FacesUCR	0.95	0.95	0.95	0.99	0.99	0.99
Fish	0.82	0.82	0.82	0.81	0.81	0.81
GunPoint	0.95	0.95	0.95	0.98	0.98	0.98
Haptics	0.40	0.40	0.40	0.43	0.43	0.43
InlineS	0.51	0.51	0.51	0.54	0.54	0.54
Italy	0.97	0.97	0.97	0.95	0.95	0.95
Lighting2	0.72	0.72	0.72	0.85	0.85	0.85
Lighting7	0.67	0.67	0.67	0.72	0.72	0.72
MALLAT	0.98	0.98	0.98	0.99	0.99	0.99
MedicalI	0.77	0.77	0.77	0.82	0.82	0.82
MoteStrain	0.92	0.92	0.92	0.98	0.98	0.98
Olive	0.88	0.88	0.88	0.87	0.87	0.87
OSULeaf	0.66	0.66	0.66	0.72	0.72	0.72
Sony	0.99	0.99	0.99	0.96	0.96	0.96
SonyII	0.99	0.99	0.99	0.98	0.98	0.98
StarLightC	0.89	0.89	0.89	0.92	0.92	0.92
Synthetic	0.91	0.91	0.91	0.98	0.98	0.98
Trace	0.86	0.86	0.86	1.00	1.00	1.00
Two	0.99	0.99	0.99	1.00	1.00	1.00
TwoLeadE	1.00	1.00	1.00	1.00	1.00	1.00
uWaveGLX	0.78	0.78	0.78	0.80	0.80	0.80
uWaveGLY	0.72	0.72	0.72	0.74	0.74	0.74
uWaveGLZ	0.70	0.70	0.70	0.74	0.74	0.74
Wafer	1.00	1.00	1.00	1.00	1.00	1.00
Yoga	0.94	0.94	0.94	0.94	0.94	0.94

Tabla B.29: Resultados de la CSR evaluada mediante el algoritmo 1NN.



# Apéndice C

## Resultados de las medidas de complejidad

Este apéndice contiene en la Sección [C.1](#) los resultados completos obtenidos a partir de la evaluación en el conjunto de datos experimentales de las medidas de complejidad involucradas en el estudio: N1, N2, N3, T1 y T2. A continuación se muestran las tablas con los valores obtenidos con cada proporción diferente de ejemplos etiquetados y las medidas euclidiana y DTW. Los valores mostrados se obtuvieron mediante un esquema de cinco particiones seleccionadas de forma aleatoria.

La Sección [C.2](#) contiene los gráficos de dispersión entre la ganancia de exactitud y la medida de complejidad que obtuvo una correlación más fuerte con la ganancia. En el gráfico se incluye además la línea que se obtiene mediante al ajustar un modelo de regresión lineal a los datos mencionados.

### C.1 Evaluación de las medidas de complejidad

Conjuntos de datos	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>T1</b>	<b>T2</b>
CBF	0.1351	0.8007	0.0560	0.4293	0.5859
Chlorine	0.4744	0.7994	0.3930	0.6186	2.0783
CinCECGt	0.0708	0.4698	0.0281	0.2228	0.0696
Coffee	0.5000	0.9341	0.4000	0.6400	0.0175
CricketX	0.7258	1.1258	0.6317	0.8762	0.2100
CricketY	0.7710	1.1978	0.7175	0.9048	0.2100
CricketZ	0.7710	1.1638	0.7175	0.9079	0.2100
ECG	0.0408	0.3036	0.0135	0.0774	1.9153
ECGFiveD	0.1343	0.5353	0.0620	0.3352	0.5221
FaceAll	0.3846	0.8378	0.2638	0.6131	1.3771
FacesUCR	0.3675	0.8241	0.2592	0.6024	1.3786
Fish	0.5111	0.9614	0.4000	0.6929	0.0605
GunPoint	0.3600	0.6701	0.2125	0.4500	0.1067
Haptics	0.7155	1.2154	0.7229	0.8560	0.0344
InlineS	0.8280	1.5203	0.8046	0.9157	0.0277
Italy	0.0575	0.5003	0.0500	0.1136	3.6667
Lighting2	0.3778	0.8965	0.3400	0.5600	0.0157
Lighting7	0.8000	1.2177	0.7143	0.9286	0.0439
MALLAT	0.0691	0.5991	0.0365	0.1531	0.1875
MedicalI	0.5495	0.9404	0.5000	0.6826	0.9293
MoteStrain	0.1267	0.7024	0.1176	0.3980	1.2143
Olive	0.5143	0.9695	0.2500	0.7500	0.0140
OSULeaf	0.6343	1.0678	0.5667	0.8667	0.0843
Sony	0.1020	0.6943	0.0440	0.3640	0.7143
SonyII	0.1410	0.6297	0.0835	0.2329	1.2154
StarLightC	0.1686	0.4060	0.1572	0.2671	0.7217
Synthetic	0.3234	0.8375	0.2042	0.5917	0.8000
Trace	0.5200	1.0257	0.4875	0.7250	0.0582
Two	0.3362	0.8606	0.2244	0.6392	3.1266
TwoLeadE	0.1391	0.6262	0.0860	0.3484	1.1341
uWaveGLX	0.3050	0.8083	0.2808	0.4730	1.1397
uWaveGLY	0.4073	0.9365	0.3721	0.6072	1.1397
uWaveGLZ	0.4056	0.8859	0.3694	0.5660	1.1397
Wafer	0.0126	0.2165	0.0084	0.0380	3.7750
Yoga	0.2106	0.6268	0.1591	0.3098	0.6197

Tabla C.1: Resultados de la evaluación de las cinco medidas de complejidad en los 35 conjuntos de datos para la distancia euclidiana y el 10% de ejemplos etiquetados.

Conjuntos de datos	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>T1</b>	<b>T2</b>
CBF	0.0743	0.7921	0.0215	0.3221	1.1641
Chlorine	0.3764	0.6463	0.2718	0.5281	4.1542
CinCECGt	0.0308	0.3710	0.0088	0.1412	0.1391
Coffee	0.1722	0.6734	0.1089	0.4133	0.0322
CricketX	0.6290	1.0393	0.5376	0.8032	0.4167
CricketY	0.6774	1.0777	0.6128	0.8512	0.4167
CricketZ	0.6532	1.0602	0.5648	0.8416	0.4167
ECG	0.0185	0.2863	0.0055	0.0505	3.8212
ECGFiveD	0.0610	0.4467	0.0394	0.2197	1.0441
FaceAll	0.2382	0.7339	0.1537	0.4822	2.7511
FacesUCR	0.2531	0.7517	0.1858	0.5031	2.7527
Fish	0.4436	0.9321	0.3357	0.6643	0.1210
GunPoint	0.2903	0.4837	0.1875	0.4000	0.2133
Haptics	0.6443	1.1288	0.6384	0.8635	0.0683
InlineS	0.7170	1.3442	0.6775	0.8368	0.0554
Italy	0.0480	0.4705	0.0386	0.0875	7.3333
Lighting2	0.3263	0.8967	0.2700	0.6700	0.0314
Lighting7	0.6763	1.1430	0.6388	0.8627	0.0727
MALLAT	0.0460	0.5346	0.0266	0.1141	0.3750
MedicalI	0.4330	0.7803	0.3628	0.5738	1.8485
MoteStrain	0.0975	0.6304	0.0892	0.2882	2.4286
Olive	0.4444	0.7974	0.2000	0.6600	0.0175
OSULeaf	0.6000	1.0236	0.5493	0.7944	0.1663
Sony	0.0747	0.6640	0.0460	0.2540	1.4286
SonyII	0.0974	0.6195	0.0637	0.2471	2.4154
StarLightC	0.1481	0.3922	0.1379	0.2530	1.4434
Synthetic	0.2253	0.8060	0.1479	0.5188	1.6000
Trace	0.4839	0.9511	0.4312	0.7125	0.1164
Two	0.2152	0.8130	0.1295	0.5349	6.2516
TwoLeadE	0.0919	0.5956	0.0409	0.2817	2.2683
uWaveGLX	0.2891	0.7978	0.2795	0.4321	2.2762
uWaveGLY	0.3640	0.9045	0.3434	0.5448	2.2762
uWaveGLZ	0.3617	0.8520	0.3450	0.5093	2.2762
Wafer	0.0084	0.1774	0.0038	0.0269	7.5447
Yoga	0.1742	0.6177	0.1367	0.2852	1.2394

Tabla C.2: Resultados de la evaluación de las cinco medidas de complejidad en los 35 conjuntos de datos para la distancia euclidiana y el 20% de ejemplos etiquetados.

Conjuntos de datos	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>T1</b>	<b>T2</b>
CBF	0.0583	0.7884	0.0241	0.3554	1.7500
Chlorine	0.3038	0.5435	0.1911	0.4783	6.2289
CinCECGt	0.0165	0.3549	0.0035	0.0968	0.2081
Coffee	0.1538	0.6760	0.1000	0.3714	0.0490
CricketX	0.6021	1.0050	0.5213	0.8000	0.6267
CricketY	0.6214	1.0451	0.5574	0.8117	0.6267
CricketZ	0.6171	1.0183	0.5309	0.8021	0.6267
ECG	0.0156	0.2735	0.0037	0.0440	5.7271
ECGFiveD	0.0425	0.4022	0.0300	0.1869	1.5662
FaceAll	0.1936	0.7008	0.1277	0.4589	4.1252
FacesUCR	0.1901	0.6992	0.1206	0.4384	4.1267
Fish	0.3855	0.8722	0.3071	0.5929	0.1814
GunPoint	0.1830	0.4557	0.1292	0.3708	0.3200
Haptics	0.5969	1.0882	0.5898	0.8245	0.1022
InlineS	0.6804	1.2367	0.6338	0.8079	0.0830
Italy	0.0502	0.4474	0.0409	0.0758	10.9917
Lighting2	0.4126	0.9261	0.3451	0.7028	0.0465
Lighting7	0.5981	1.0777	0.5639	0.7761	0.1091
MALLAT	0.0438	0.5067	0.0288	0.1052	0.5625
MedicalI	0.3909	0.7343	0.3209	0.5252	2.7697
MoteStrain	0.0964	0.6296	0.0922	0.3039	3.6429
Olive	0.3429	0.7768	0.2000	0.4667	0.0263
OSULeaf	0.5626	0.9958	0.5067	0.8087	0.2496
Sony	0.0430	0.6507	0.0254	0.2136	2.1400
SonyII	0.0784	0.6031	0.0382	0.1985	3.6277
StarLightC	0.1394	0.3740	0.1303	0.2313	2.1650
Synthetic	0.2084	0.7994	0.1597	0.5014	2.4000
Trace	0.4340	0.7720	0.3250	0.6167	0.1745
Two	0.1478	0.7869	0.0793	0.4801	9.3766
TwoLeadE	0.0511	0.5549	0.0222	0.2215	3.4024
uWaveGLX	0.2782	0.7881	0.2649	0.4019	3.4127
uWaveGLY	0.3456	0.9004	0.3260	0.5343	3.4127
uWaveGLZ	0.3428	0.8466	0.3334	0.4945	3.4127
Wafer	0.0073	0.1567	0.0041	0.0208	11.3145
Yoga	0.1461	0.5978	0.1098	0.2452	1.8592

Tabla C.3: Resultados de la evaluación de las cinco medidas de complejidad en los 35 conjuntos de datos para la distancia euclidiana y el 30% de ejemplos etiquetados.



Conjuntos de datos	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>T1</b>	<b>T2</b>
CBF	0.0351	0.6861	0.0053	0.1867	0.5859
Chlorine	0.4924	0.8534	0.4301	0.6545	2.0783
CinCECGt	0.0885	0.3462	0.0386	0.2000	0.0696
Coffee	0.6000	1.0192	0.4800	0.7600	0.0175
CricketX	0.6355	1.2035	0.5238	0.7365	0.2100
CricketY	0.6581	1.2915	0.5873	0.7905	0.2100
CricketZ	0.6710	1.2661	0.6000	0.8159	0.2100
ECG	0.0395	0.3203	0.0098	0.0811	1.9153
ECGFiveD	0.1486	0.6013	0.0958	0.3099	0.5221
FaceAll	0.1694	0.7725	0.0864	0.3969	1.3771
FacesUCR	0.1704	0.7646	0.0897	0.4297	1.3786
Fish	0.4519	0.9522	0.3071	0.6357	0.0605
GunPoint	0.2667	0.5451	0.1500	0.4000	0.1067
Haptics	0.6994	1.5089	0.7235	0.8354	0.0344
InlineS	0.8047	1.9759	0.7509	0.8583	0.0277
Italy	0.0759	0.5936	0.0659	0.1273	3.6667
Lighting2	0.3778	0.8074	0.3600	0.5200	0.0157
Lighting7	0.7385	1.3291	0.6571	0.8714	0.0439
MALLAT	0.0754	0.5393	0.0385	0.1729	0.1875
MedicalI	0.5363	0.8745	0.4522	0.6609	0.9293
MoteStrain	0.0812	0.5082	0.0549	0.1765	1.2143
Olive	0.5429	0.8950	0.2750	0.7500	0.0140
OSULeaf	0.6114	1.1931	0.5556	0.7500	0.0843
Sony	0.1224	0.7704	0.0760	0.3920	0.7143
SonyII	0.1256	0.7519	0.0785	0.3392	1.2154
StarLightC	0.1347	0.1482	0.1261	0.1432	0.7217
Synthetic	0.2298	0.7797	0.1250	0.4792	0.8000
Trace	0.4133	0.8427	0.2500	0.5625	0.0582
Two	0.0827	0.5419	0.0350	0.2954	3.1266
TwoLeadE	0.0674	0.6469	0.0409	0.2903	1.1341
uWaveGLX	0.2827	0.6759	0.2552	0.3911	1.1397
uWaveGLY	0.3894	0.8760	0.3588	0.4953	1.1397
uWaveGLZ	0.3737	0.7986	0.3432	0.4758	1.1397
Wafer	0.0189	0.0645	0.0101	0.0240	3.7750
Yoga	0.1848	0.5423	0.1386	0.2924	0.6197

Tabla C.4: Resultados de la evaluación de las cinco medidas de complejidad en los 35 conjuntos de datos para la medida DTW y el 10 % de ejemplos etiquetados.

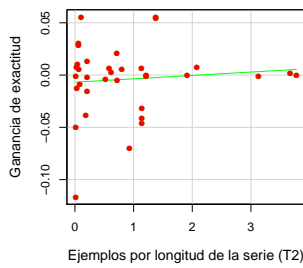
Conjuntos de datos	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>T1</b>	<b>T2</b>
CBF	0.0162	0.6964	0.0067	0.1503	1.1641
Chlorine	0.3901	0.7082	0.3022	0.5580	4.1542
CinCECGt	0.0344	0.2655	0.0114	0.1193	0.1391
Coffee	0.2194	0.7935	0.1511	0.4333	0.0322
CricketX	0.5145	0.9709	0.3840	0.6608	0.4167
CricketY	0.5468	1.0581	0.4544	0.7248	0.4167
CricketZ	0.5274	1.0157	0.4224	0.6640	0.4167
ECG	0.0241	0.3072	0.0086	0.0567	3.8212
ECGFiveD	0.0922	0.5754	0.0521	0.2761	1.0441
FaceAll	0.0874	0.7291	0.0427	0.3135	2.7511
FacesUCR	0.0979	0.7284	0.0494	0.3217	2.7527
Fish	0.4545	0.9570	0.3857	0.6500	0.1210
GunPoint	0.1742	0.3152	0.0938	0.2938	0.2133
Haptics	0.6439	1.3075	0.6112	0.7856	0.0683
InlineS	0.7015	1.6113	0.6276	0.7582	0.0554
Italy	0.0777	0.5838	0.0670	0.1023	7.3333
Lighting2	0.2526	0.7497	0.2200	0.4500	0.0314
Lighting7	0.6040	1.2545	0.5707	0.7855	0.0727
MALLAT	0.0423	0.4835	0.0234	0.1292	0.3750
MedicalI	0.4308	0.6845	0.3552	0.5475	1.8485
MoteStrain	0.0512	0.4482	0.0422	0.1333	2.4286
Olive	0.4222	0.8220	0.2000	0.7200	0.0175
OSULeaf	0.5343	1.0247	0.5099	0.6592	0.1663
Sony	0.0707	0.7424	0.0480	0.2640	1.4286
SonyII	0.1090	0.7440	0.0803	0.3478	2.4154
StarLightC	0.1202	0.1357	0.1127	0.1222	1.4434
Synthetic	0.1116	0.7430	0.0521	0.3521	1.6000
Trace	0.3484	0.6945	0.2438	0.5188	0.1164
Two	0.0180	0.4981	0.0075	0.1957	6.2516
TwoLeadE	0.0259	0.6453	0.0118	0.2215	2.2683
uWaveGLX	0.2642	0.6611	0.2483	0.3593	2.2762
uWaveGLY	0.3366	0.8181	0.3158	0.4510	2.2762
uWaveGLZ	0.3433	0.7453	0.3222	0.4396	2.2762
Wafer	0.0119	0.0570	0.0035	0.0159	7.5447
Yoga	0.1564	0.5514	0.1231	0.2633	1.2394

Tabla C.5: Resultados de la evaluación de las cinco medidas de complejidad en los 35 conjuntos de datos para la medida DTW y el 20 % de ejemplos etiquetados.

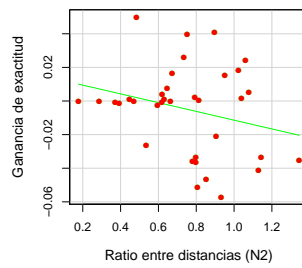
Conjuntos de datos	<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>T1</b>	<b>T2</b>
CBF	0.0126	0.6971	0.0036	0.1268	1.7500
Chlorine	0.3402	0.6339	0.2348	0.5205	6.2289
CinCECGt	0.0194	0.2347	0.0041	0.0868	0.2081
Coffee	0.1846	0.8193	0.1857	0.4143	0.0490
CricketX	0.4545	0.9009	0.3500	0.6032	0.6267
CricketY	0.4652	0.9830	0.3755	0.6596	0.6267
CricketZ	0.4524	0.9049	0.3457	0.5915	0.6267
ECG	0.0181	0.2990	0.0049	0.0444	5.7271
ECGFiveD	0.0679	0.5303	0.0376	0.2216	1.5662
FaceAll	0.0697	0.7133	0.0352	0.2813	4.1252
FacesUCR	0.0715	0.7107	0.0385	0.2601	4.1267
Fish	0.3518	0.8687	0.2786	0.6143	0.1814
GunPoint	0.1149	0.2801	0.0750	0.2208	0.3200
Haptics	0.5806	1.1789	0.5737	0.7762	0.1022
InlineS	0.6340	1.3231	0.5773	0.7080	0.0830
Italy	0.0609	0.5644	0.0546	0.0796	10.9917
Lighting2	0.3914	0.8569	0.3310	0.5807	0.0465
Lighting7	0.5160	1.0316	0.4499	0.6787	0.1091
MALLAT	0.0313	0.4577	0.0174	0.1042	0.5625
MedicalII	0.3616	0.6635	0.2954	0.4996	2.7697
MoteStrain	0.0544	0.4541	0.0412	0.1405	3.6429
Olive	0.3143	0.7305	0.1600	0.4667	0.0263
OSULeaf	0.4813	0.9428	0.4261	0.6212	0.2496
Sony	0.0725	0.7482	0.0534	0.2617	2.1400
SonyII	0.0886	0.7443	0.0611	0.3265	3.6277
StarLightC	0.1066	0.1276	0.0980	0.1022	2.1650
Synthetic	0.0825	0.7437	0.0431	0.3597	2.4000
Trace	0.2000	0.3883	0.0667	0.4375	0.1745
Two	0.0072	0.4840	0.0027	0.1456	9.3766
TwoLeadE	0.0137	0.6192	0.0057	0.1720	3.4024
uWaveGLX	0.2596	0.6504	0.2463	0.3392	3.4127
uWaveGLY	0.3238	0.8053	0.3070	0.4216	3.4127
uWaveGLZ	0.3205	0.7359	0.3055	0.4095	3.4127
Wafer	0.0091	0.0527	0.0035	0.0142	11.3145
Yoga	0.1317	0.5582	0.0990	0.2439	1.8592

Tabla C.6: Resultados de la evaluación de las cinco medidas de complejidad en los 35 conjuntos de datos para la medida DTW y el 30 % de ejemplos etiquetados.

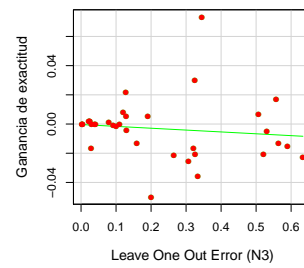
## C.2 Dispersión entre la ganancia de exactitud y la complejidad



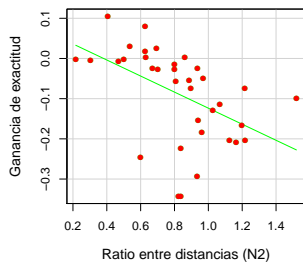
(a) SelfT-1NN (10%)



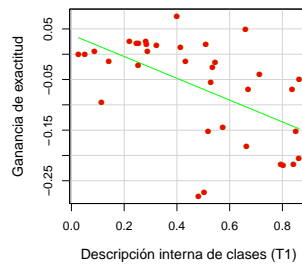
(b) SelfT-1NN (20%)



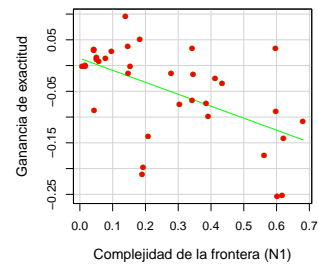
(c) SelfT-1NN (30%)



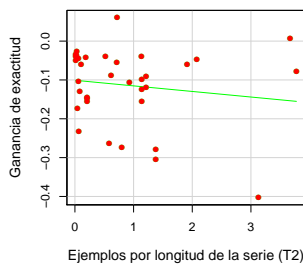
(d) SelfT-SVM (10%)



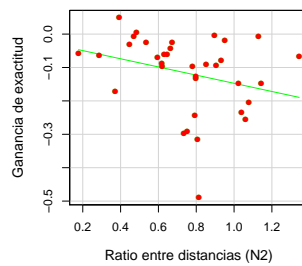
(e) SelfT-SVM (20%)



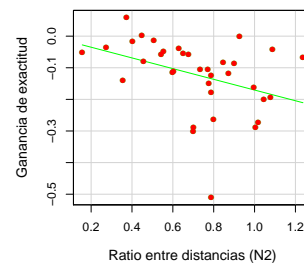
(f) SelfT-SVM (30%)



(g) SelfT-DT (10%)

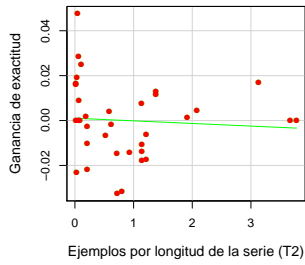


(h) SelfT-DT (20%)

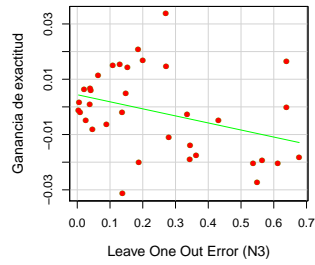


(i) SelfT-DT (30%)

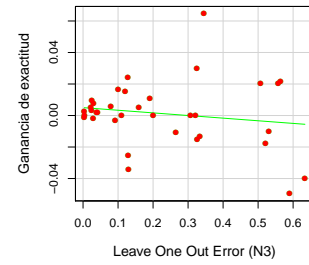
Figura C.1: Gráfico de dispersión entre la ganancia de exactitud obtenida con el método Self-training y la complejidad de mayor correlación para la distancia euclidiana.



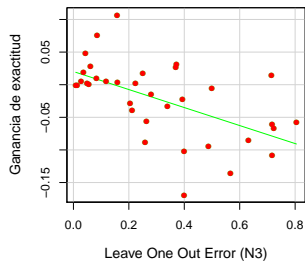
(a) TriT-1NN (10%)



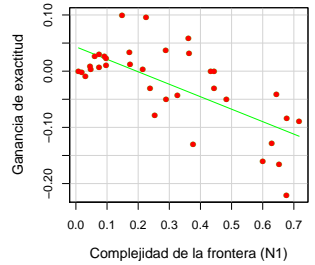
(b) TriT-1NN (20%)



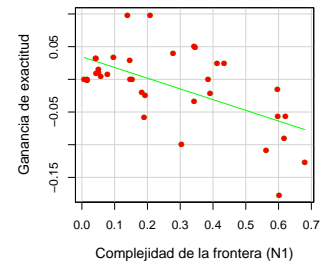
(c) TriT-1NN (30%)



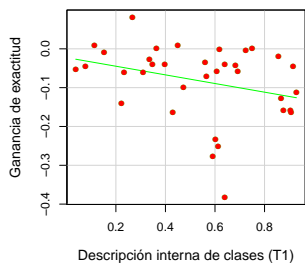
(d) TriT-SVM (10%)



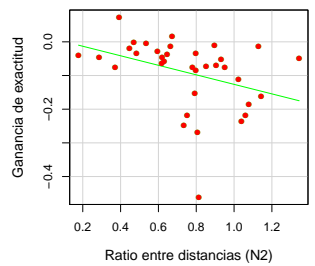
(e) TriT-SVM (20%)



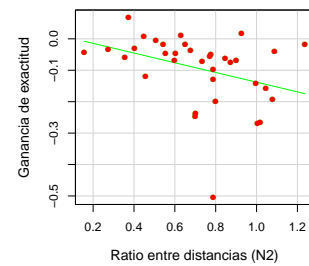
(f) TriT-SVM (30%)



(g) TriT-DT (10%)

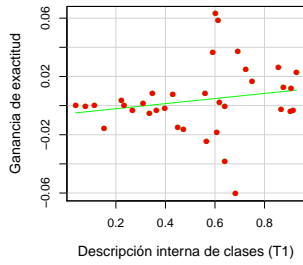


(h) TriT-DT (20%)

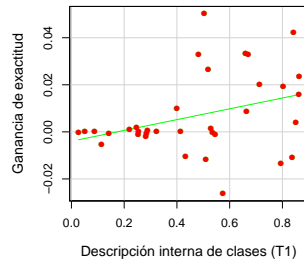


(i) TriT-DT (30%)

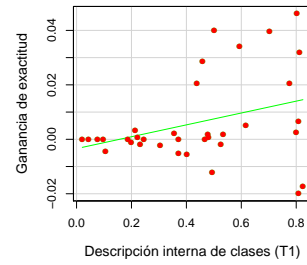
Figura C.2: Gráfico de dispersión entre la ganancia de exactitud obtenida con el método Tri-training y la complejidad de mayor correlación para la distancia euclidiana.



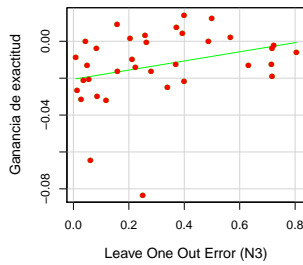
(a) SETRED (10%)



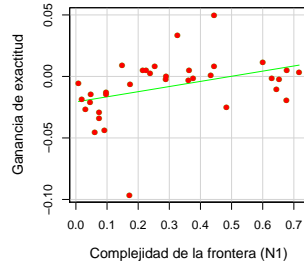
(b) SETRED (20%)



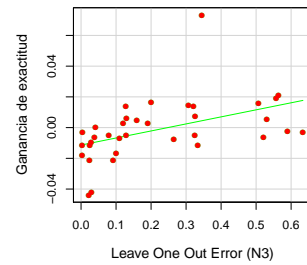
(c) SETRED (30%)



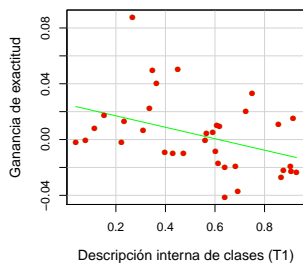
(d) SNNRCE (10%)



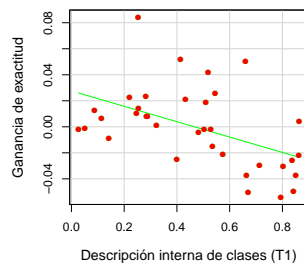
(e) SNNRCE (20%)



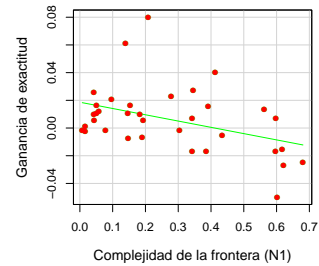
(f) SNNRCE (30%)



(g) Democratic (10%)

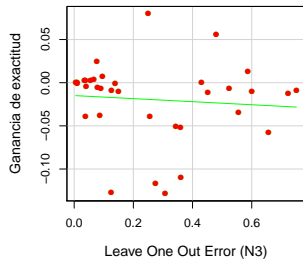


(h) Democratic (20%)

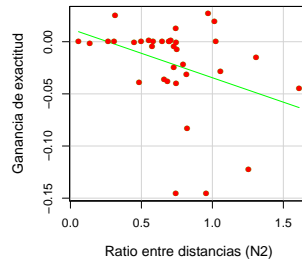


(i) Democratic (30%)

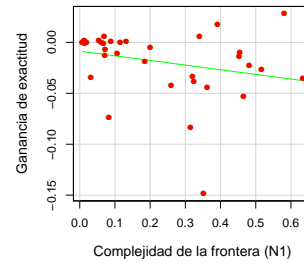
Figura C.3: Gráfico de dispersión entre la ganancia de exactitud obtenida con varios métodos semi-supervisados y la complejidad de mayor correlación para la distancia euclidiana.



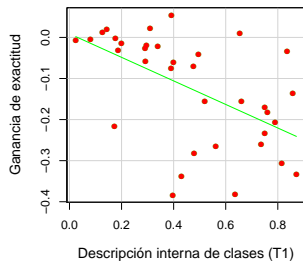
(a) SelfT-1NN (10 %)



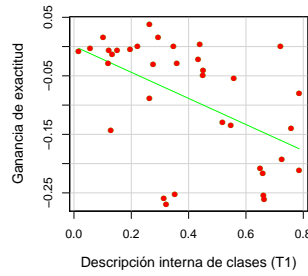
(b) SelfT-1NN (20 %)



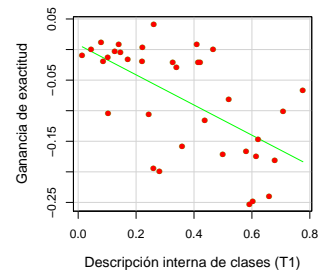
(c) SelfT-1NN (30 %)



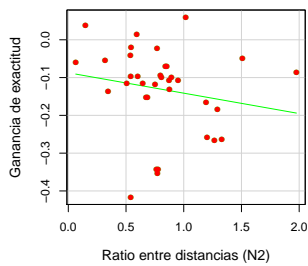
(d) SelfT-SVM (10 %)



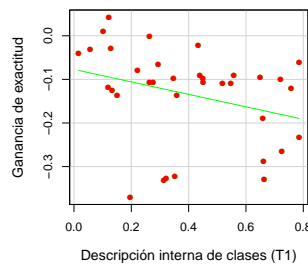
(e) SelfT-SVM (20 %)



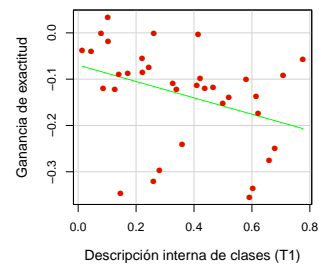
(f) SelfT-SVM (30 %)



(g) SelfT-DT (10 %)



(h) SelfT-DT (20 %)



(i) SelfT-DT (30 %)

Figura C.4: Gráfico de dispersión entre la ganancia de exactitud obtenida con el método Self-training y la complejidad de mayor correlación para DTW.

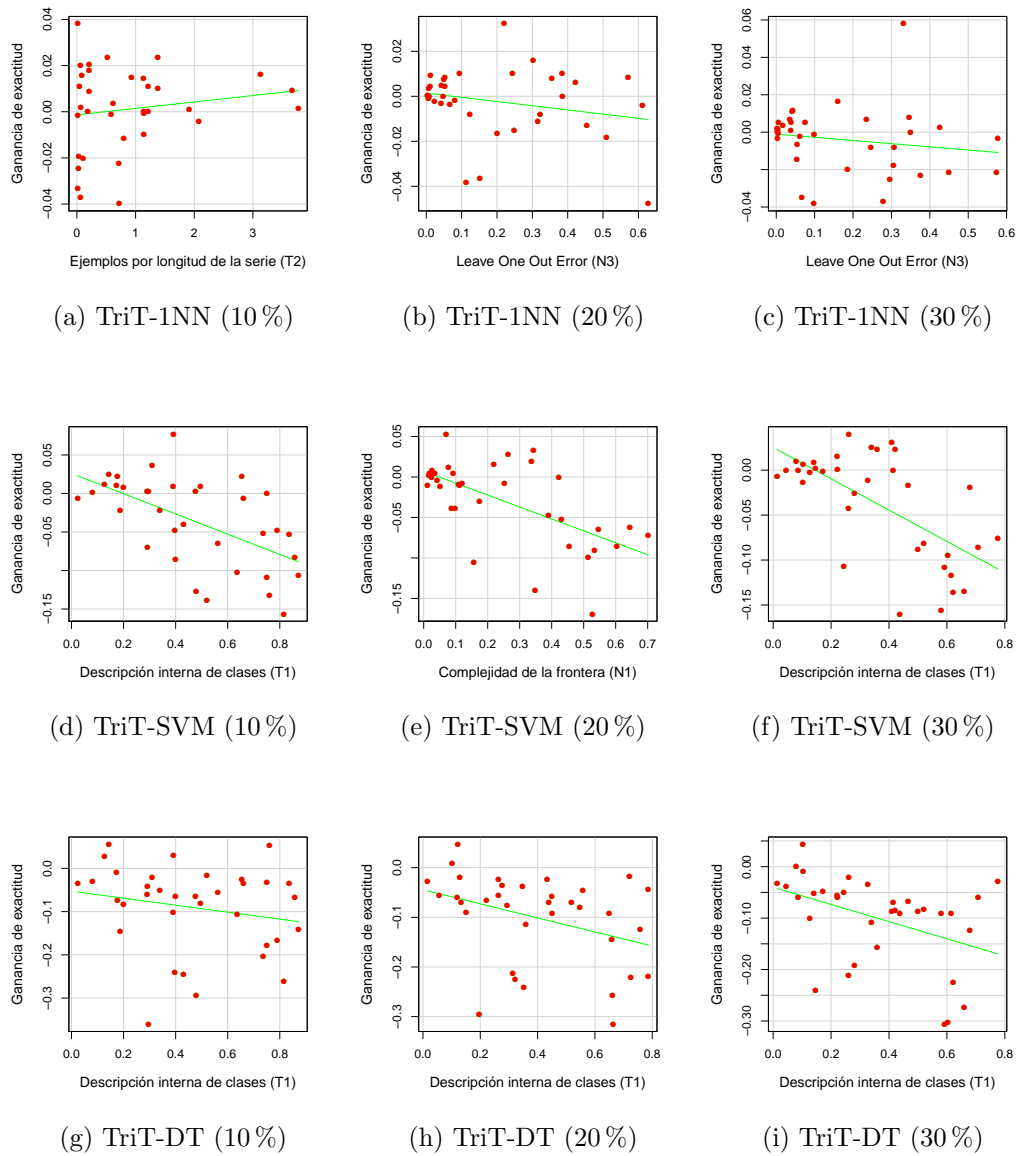
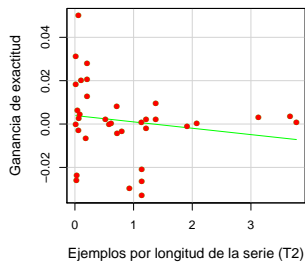
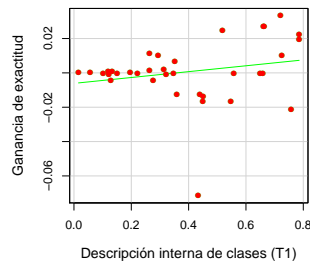


Figura C.5: Gráfico de dispersión entre la ganancia de exactitud obtenida con el método Tri-training y la complejidad de mayor correlación para la medida DTW.

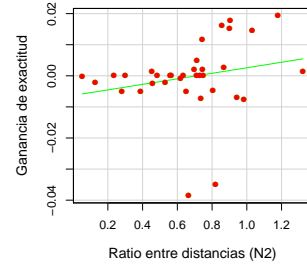




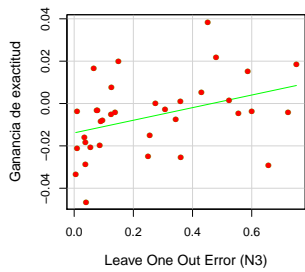
(a) SETRED (10%)



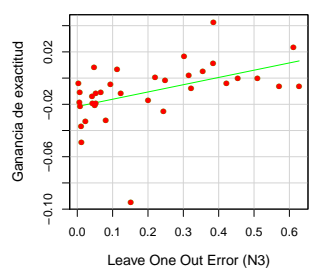
(b) SETRED (20%)



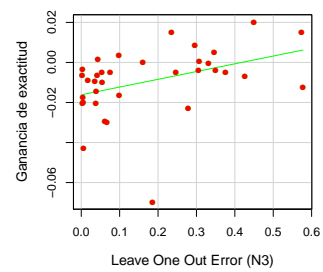
(c) SETRED (30%)



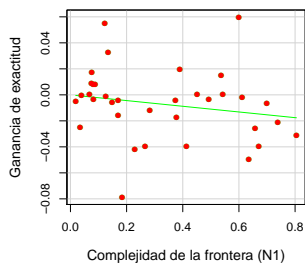
(d) SNNRCE (10%)



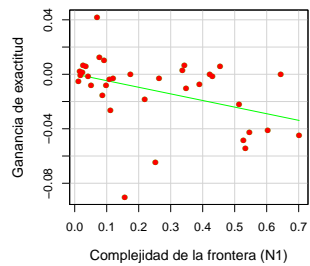
(e) SNNRCE (20%)



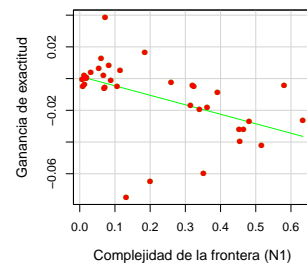
(f) SNNRCE (30%)



(g) Democratic (10%)



(h) Democratic (20%)



(i) Democratic (30%)

Figura C.6: Gráfico de dispersión entre la ganancia de exactitud obtenida con varios métodos semi-supervisados y la complejidad de mayor correlación para la medida DTW.



# Apéndice D

## Demostración de las propiedades de la curva estable

Este apéndice contiene la demostración de la siguiente afirmación:

**Proposición 1:** Sea  $I_{[b,\dots,p,\dots,v,\dots,e]}$  un intervalo formado por las tres curvas definidas (Figura D.1), los ejemplos seleccionados entre las iteraciones  $v$  y  $e$  se encuentran más cercanos a los ejemplos seleccionados después de la iteración  $p$  que a los ejemplos adicionados antes de dicha iteración.

Esto significa que un ejemplo no etiquetado  $x_h$ , seleccionado por el algoritmo P1NNPU durante la iteración  $h$  siendo ( $v < h < e$ ), satisface la siguiente expresión:

$$\text{Mindist}(h) = \text{dist}(x_h, x_m), m \notin [b, \dots, p - 1] \quad (\text{D.1})$$

En la Expresión D.1,  $x_m$  es un ejemplo no etiquetado seleccionado en la iteración  $m$  siendo  $m < h$ . Ahora se probará la Proposición 1 mediante reducción al absurdo. Primeramente, se asumirá lo contrario:

$$\text{Mindist}(h) = \text{dist}(x_h, x_m), m \in [b, \dots, p - 1] \quad (\text{D.2})$$

Como el algoritmo selecciona en cada iteración el ejemplo de  $U$  más cercano al conjunto  $P$  actual, la selección de  $x_h$  durante la iteración  $h$  implica que:

$$\text{Mindist}(t) < \text{Mindist}(h), \forall t = m + 1, \dots, h - 1 \quad (\text{D.3})$$

Esto significa que entre la iteración  $m + 1$  y la  $h - 1$ , deben ser seleccionados solamente ejemplos a una distancia mínima menor que  $\text{Mindist}(h)$ . Esta implicación es falsa pues  $\exists p$  tal que  $\text{Mindist}(p) > \text{Mindist}(h)$ . Esta condición no puede ser violada ya que forma parte de la definición de curva estable. La contradicción a la que se ha llegado prueba la Proposición 1 lo cual implica que  $m \in [p, \dots, h - 1]$ .

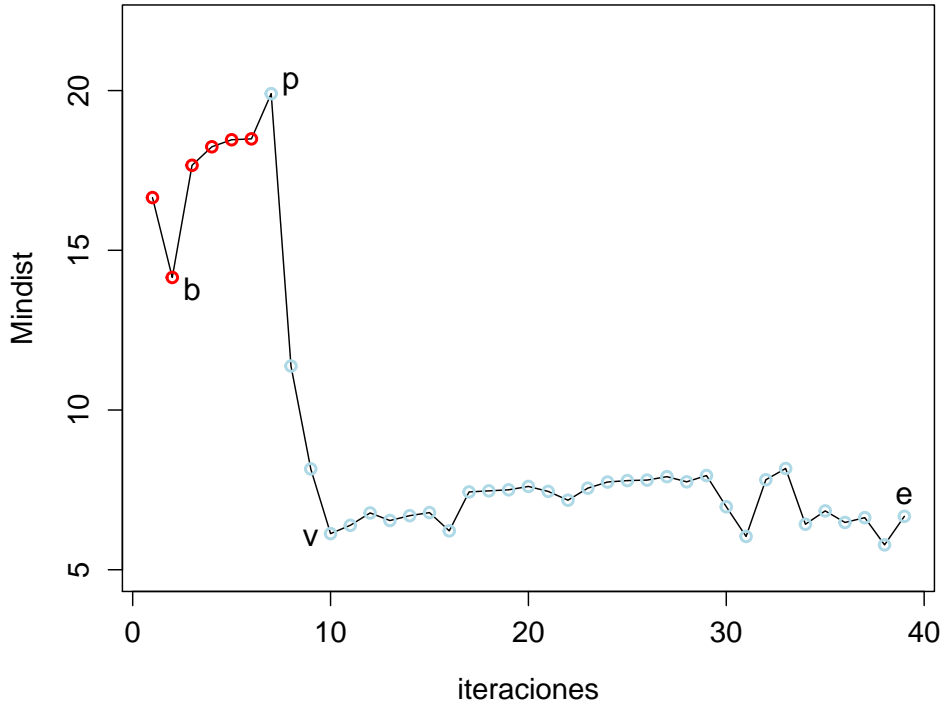


Figura D.1: Un típico intervalo  $I_{[b,e]}$  formado por la unión de una curva ascendente  $ac_{[b,p]}$ , una curva descendente  $dc_{[p,v]}$  y una curva estable  $sc_{[v,e]}$ .

# Apéndice E

## Detalles sobre la implementación del paquete `ssc`

Este apéndice contiene detalles relacionados con la implementación en lenguaje R del paquete `ssc`. A continuación se muestran algunas medidas básicas que permiten contabilizar aspectos relacionados con el monto de esfuerzo requerido durante el desarrollo de este software.

- **Número de funciones implementadas:** el paquete cuenta con un total de 47 funciones, de las cuales 17 son públicas y el resto son funciones privadas.
- **Cantidad de líneas de código:** este número asciende a un total de 1034 líneas de código fuente.
- **Cantidad de líneas de documentación:** en R es posible documentar el código dentro de los propios archivos fuente. Este número asciende a un total de 1021 líneas. Aquí se incluyen los comentarios que se han introducido en el código para facilitar su lectura y mantenimiento.
- **Cantidad de horas de programación:** este número asciende aproximadamente a un total de 920 horas.

La Tabla [E.1](#) muestra de forma desglosada el comportamiento de tres de las medidas evaluadas en cada uno de los archivos de código contenidos en el directorio `R/` del paquete.

Archivo	Número de funciones	Líneas de código	Líneas de documentación
BaseClassifier.r	3	22	61
BaseClassifierOneNN.r	5	58	54
CoBC.r	6	142	146
DataSets.r	0	2	38
Democratic.r	5	175	137
SETRED.r	3	113	128
SNNRCE.r	2	138	115
SelfTraining.r	2	59	109
SemiSupervised.r	7	77	39
Statistics.r	5	45	25
TriTraining.r	5	112	124
Utils.r	4	91	45

Tabla E.1: Evaluación de las medidas en cada archivo de código fuente.

# Bibliografía

- [1] *Languages for analytics/data mining*. <http://www.kdnuggets.com/2015/05/poll-r-rapidminer-python-big-data-spark.html>. Accessed: 2015-05.
- [2] Abraham, Z. y P. N. Tan: *A Semi-supervised Framework for Simultaneous Classification and Regression of Zero-Inflated Time Series Data with Application to Precipitation Prediction*. En *IEEE International Conference on Data Mining Workshops 2009*, págs. 644–649, Dec 2009.
- [3] Alcalá-Fdez, J., L. Sánchez, S. García, M. J. Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández y F. Herrera: *KEEL: a software tool to assess evolutionary algorithms for data mining problems*. *Soft Computing*, 13(3):307–318, 2008, ISSN 1433-7479.
- [4] Bagnall, A., J. Lines, J. Hills y A. Bostrom: *Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles*. *Knowledge and Data Engineering, IEEE Transactions on*, 27(9):2522–2535, Sept 2015, ISSN 1041-4347.
- [5] Bagnall, A. J. y G. J. Janacek: *Clustering Time Series from ARMA Models with Clipped Data*. En *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, págs. 49–58, New York, NY, USA, 2004. ACM, ISBN 1-58113-888-1.
- [6] Balakrishnan, S. y D. Madigan: *Decision Trees for Functional Variables*. En *Sixth International Conference on Data Mining, ICDM '06*, págs. 798–802, Dec 2006.

- 
- [7] Bartolini, I., P. Ciaccia y M. Patella: *WARP: accurate retrieval of shapes using phase of Fourier descriptors and time warping distance*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(1):142–147, Jan 2005, ISSN 0162-8828.
- [8] Basu, M. y T.K. Ho: *Data complexity in pattern recognition*. Springer London, 2006.
- [9] Batista, G., Y. Hao, E. Keogh y A. Mafra-Neto: *Towards automatic classification on flying insects using inexpensive sensors*. En *10th International Conference on Machine Learning and Applications (ICMLA)*, vol. 1, págs. 364–369. IEEE, 2011.
- [10] Batista, G.E., E.J. Keogh, A. MafraNeto y E. Rowton: *SIGKDD Demo: Sensors and Software to Allow Computational Entomology, an Emerging Application of Data Mining*. En *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, págs. 761–764, New York, NY, USA, 2011. ACM, ISBN 978-1-4503-0813-7.
- [11] Begum, N., B. Hu, T. Rakthanmanon y E. Keogh: *A Minimum Description Length Technique for Semi-Supervised Time Series Classification*. En *Integration of Reusable Systems*, vol. 263 de *Advances in Intelligent Systems and Computing*, págs. 171–192. Springer International Publishing, 2014.
- [12] Behera, H., P. Dash y B. Biswal: *Power quality time series data mining using S-transform and fuzzy expert system*. Applied Soft Computing, 10(3):945–955, 2010, ISSN 1568-4946.
- [13] Ben-David, A.: *A lot of randomness is hiding in accuracy*. Engineering Applications of Artificial Intelligence, 20(7):875–885, 2007, ISSN 0952-1976.
- [14] Ben-David, S., T. Lu y D. Pál: *Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning*. En *21st Annual Conference on Learning Theory, COLT' 2008*, págs. 33–44, Helsinki, Finland, 2008.



- [15] Bergmann, B. y G. Hommel: *Improvements of General Multiple Test Procedures for Redundant Systems of Hypotheses*. En *Multiple Hypotheses Testing*, vol. 70 de *Medizinische Informatik und Statistik*, págs. 100–115. Springer Berlin Heidelberg, 1988.
- [16] Biecek, P., E. Szcurek, M. Vingron y J. Tiuryn: *The R Package bgmm: Mixture Modeling with Uncertain Knowledge*. *Journal of Statistical Software*, 47(3):1–32, 2012.
- [17] Biswal, B., P. Dash y S. Mishra: *A hybrid ant colony optimization technique for power signal pattern classification*. *Expert Systems with Applications*, 38(5):6368–6375, 2011, ISSN 0957-4174.
- [18] Blum, A. y S. Chawla: *Learning from Labeled and Unlabeled Data Using Graph Mincuts*. En *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, págs. 19–26, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc., ISBN 1-55860-778-1.
- [19] Blum, A. y T. Mitchell: *Combining Labeled and Unlabeled Data with Co-training*. En *Eleventh Annual Conference on Computational Learning Theory, COLT' 98*, págs. 92–100, New York, NY, USA, 1998. ACM, ISBN 1-58113-057-0.
- [20] Blum, A. y T. Mitchell: *Combining labeled and unlabeled data with co-training*. En *Proceedings of the eleventh annual conference on computational learning theory*, págs. 92–100. ACM, 1998.
- [21] Box, G. E. y G. M. Jenkins: *Time series analysis: forecasting and control*. Holden-Day, San Francisco, 1976.
- [22] Breiman, L.: *Bagging predictors*. *Machine Learning*, 24(2):123–140, 1996, ISSN 1573-0565.
- [23] Breiman, L., J. Friedman, C. J. Stone y R. A. Olshen: *Classification and regression trees*. CRC press, 1984.
- [24] Brockwell, P. J. y R. A. Davis: *Introduction to time series and forecasting*. Springer Science & Business Media, second ed., 2006, ISBN 978-0-387-95351-9.

- [25] Bruno, G. y P. Garza: *Temporal Pattern Mining for Medical Applications*. Intelligent Systems Reference Library, 25:9–18, 2012.
- [26] Buza, K., A. Nanopoulos, L. Schmidt-Thieme y J. Koller: *Fast Classification of Electrocardiograph Signals via Instance Selection*. En *First IEEE International Conference on Healthcare Informatics, Imaging and Systems Biology (HISB), 2011*, págs. 9–16, July 2011.
- [27] Calvo, B., P. Larrañaga y J. A. Lozano: *Learning Bayesian classifiers from positive and unlabeled examples*. Pattern Recognition Letters, 28(16):2375–2384, 2007.
- [28] Carden, E. P. y J. M. Brownjohn: *ARMA modelled time-series classification for structural health monitoring of civil infrastructure*. Mechanical Systems and Signal Processing, 22(2):295–314, 2008, ISSN 0888-3270.
- [29] Chapelle, O., B. Schölkopf y A. Zien: *Semi-supervised learning*, vol. 2. MIT Press Cambridge, 2006.
- [30] Chatfield, C.: *The analysis of time series: an introduction*. CHAPMAN & HALL/CRC Texts in Statistical Science. CRC press, sixth edition ed., 2013.
- [31] Chen, L. y R. Ng: *On the Marriage of  $L_p$ -norms and Edit Distance*. En *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30, VLDB '04*, págs. 792–803. VLDB Endowment, 2004, ISBN 0-12-088469-0.
- [32] Chen, L., M. T. Özsu y V. Oria: *Robust and Fast Similarity Search for Moving Object Trajectories*. En *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD '05*, págs. 491–502, New York, NY, USA, 2005. ACM, ISBN 1-59593-060-4.
- [33] Chen, Y., B. Hu, E. Keogh y G.E. Batista: *DTW-D: Time Series Semi-supervised Learning from a Single Example*. En *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13*, págs. 383–391, New York, NY, USA, 2013. ACM, ISBN 978-1-4503-2174-7.

- [34] Chen, Y., E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen y G. Batista: *The UCR Time Series Classification Archive*, July 2015. [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/).
- [35] Cohen, I., F. G. Cozman, N. Sebe, M. C. Cirelo y T. S. Huang: *Semisupervised learning of classifiers: theory, algorithms, and their application to human-computer interaction*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(12):1553–1566, Dec 2004, ISSN 0162-8828.
- [36] Cohen, I., F. G. Cozman, N. Sebe, M. C. Cirelo y T. S. Huang: *Semisupervised learning of classifiers: theory, algorithms, and their application to human-computer interaction*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(12):1553–1566, Dec 2004, ISSN 0162-8828.
- [37] Corduas, M. y D. Piccolo: *Time series clustering and classification by the autoregressive metric*. Computational Statistics & Data Analysis, 52(4):1860–1872, 2008, ISSN 0167-9473.
- [38] Cover, T. y P. Hart: *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory, 13(1):21–27, January 1967, ISSN 0018-9448.
- [39] Cowpertwait, P. S. y A. V. Metcalfe: *Introductory time series with R*. Springer Science & Business Media, 2009.
- [40] Culp, M.: *spa: Semi-Supervised Semi-Parametric Graph-Based Estimation in R*. Journal of Statistical Software, 40(1):1–29, 2011, ISSN 1548-7660.
- [41] Dash, P., H. Behera y I. Lee: *Time sequence data mining using time-frequency analysis and soft computing techniques*. Applied Soft Computing, 8(1):202–215, 2008, ISSN 1568-4946.
- [42] De Comité, F., F. Denis, R. Gilleron y F. Letouzey: *Positive and Unlabeled Examples Help Learning*, págs. 219–230. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, ISBN 978-3-540-46769-4.
- [43] De Sousa, C. A. R., V. M. A. Souza y G. E. A. P. A. Batista: *Time Series Transductive Classification on Imbalanced Data Sets: An Experimental Study*. En *22nd International Conference on Pattern Recognition (ICPR)*, págs. 3780–3785, Aug 2014.

- [44] De Sousa, C. A. R., V. M. A. Souza y G. E. A. P. A. Batista: *An experimental analysis on time series transductive classification on graphs*. En *International Joint Conference on Neural Networks (IJCNN)*, págs. 1–8, July 2015.
- [45] Deng, H., G. Runger, E. Tuv y M. Vladimir: *A time series forest for classification and feature extraction*. *Information Sciences*, 239:142–153, 2013.
- [46] Denis, F.: *PAC learning from positive statistical queries*. En *Algorithmic Learning Theory*, págs. 112–126. Springer, 1998.
- [47] Denis, F., R. Gilleron, A. Laurent y M. Tommasi: *Text classification and co-training from positive and unlabeled examples*. En *Proceedings of the ICML 2003 workshop: the continuum from labeled to unlabeled data*, págs. 80–87, 2003.
- [48] Denis, F., R. Gilleron y M. Tommasi: *Text classification from positive and unlabeled examples*. En *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'02*, págs. 1927–1934, 2002.
- [49] Douzal-Chouakria, A. y C. Amblard: *Classification trees for time series*. *Pattern Recognition*, 45(3):1076–1091, 2012, ISSN 0031-3203.
- [50] Edwards, R. D., J. Magee y W. H. C. Bassetti: *Technical analysis of stock trends*. CRC Press Taylor & Francis Group, 10th ed., 2012.
- [51] Emerson, P.: *The original Borda count and partial voting*. *Social Choice and Welfare*, 40(2):353–358, 2013, ISSN 1432-217X.
- [52] Esling, P. y C. Agon: *Time-series Data Mining*. *ACM Comput. Surv.*, 45(1):12:1–12:34, Dic. 2012, ISSN 0360-0300.
- [53] Farouk Abdel Hady, M. y F. Schwenker: *Combining Committee-Based Semi-Supervised Learning and Active Learning*. *Journal of Computer Science and Technology*, 25(4):681–698, 2010.
- [54] Flesca, S., G. Manco, E. Masciari, L. Pontieri y A. Pugliese: *Exploiting structural similarity for effective Web information extraction*. *Data & Knowledge Engineering*, 60(1):222–234, 2007.

- [55] Frank, J., S. Mannor, J. Pineau y D. Precup: *Time Series Analysis Using Geometric Template Matching*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(3):740–754, March 2013, ISSN 0162-8828.
- [56] Frank, L. y E. Hubert: *Pretopological approach for supervised learning*. En *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 4, págs. 256–260, Aug 1996.
- [57] Freund, Y. y R. E. Schapire: *Experiments with a new boosting algorithm*. En *Thirteenth International Conference on Machine Learning (ICML'96)*, vol. 96, págs. 148–156, 1996.
- [58] Friedman, M.: *The use of ranks to avoid the assumption of normality implicit in the analysis of variance*. Journal of the American Statistical Association, 32(200):675–701, 1937.
- [59] Fu, T.: *A review on time series data mining*. Engineering Applications of Artificial Intelligence, 24(1):164–181, 2011.
- [60] Fujino, A., N. Ueda y K. Saito: *Semisupervised Learning for a Hybrid Generative/Discriminative Classifier based on the Maximum Entropy Principle*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(3):424–437, March 2008, ISSN 0162-8828.
- [61] Fulcher, B. D. y N. S. Jones: *Highly Comparative Feature-Based Time-Series Classification*. IEEE Transactions on Knowledge and Data Engineering, 26(12):3026–3037, Dec 2014, ISSN 1041-4347.
- [62] García, S., A. Fernández, J. Luengo y F. Herrera: *Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power*. Information Sciences, 180(10):2044–2064, 2010.
- [63] García, S. y F. Herrera: *An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons*. Journal of Machine Learning Research, 9:2695–2698, 2008.

- [64] Geler, Z., V. Kurbalija, M. Radovanović y M. Ivanović: *Comparison of different weighting schemes for the kNN classifier on time-series data*. Knowledge and Information Systems, págs. 1–48, 2015, ISSN 0219-3116.
- [65] Giorgino, T.: *Computing and visualizing dynamic time warping alignments in R: the dtw package*. Journal of statistical Software, 31(7):1–24, 2009.
- [66] Goldberger, A. L., L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. Peng y H. E. Stanley: *Physio-bank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals*. Circulation, 101(23):e215–e220, 2000.
- [67] Goldman, S. A. y Y. Zhou: *Enhancing Supervised Learning with Unlabeled Data*. En *Seventeenth International Conference on Machine Learning (ICML)*, págs. 327–334, June 29 - July 2 2000.
- [68] Grabocka, J., A. Nanopoulos y L. Schmidt-Thieme: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*, cap. Invariant Time-Series Classification, págs. 725–740. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, ISBN 978-3-642-33486-3.
- [69] Grünwald, P. D., I. J. Myung y M. A. Pitt: *Advances in minimum description length: Theory and applications*. MIT Press, 2005.
- [70] Hamooni, H., A. Mueen y A. Neel: *Phoneme sequence recognition via DTW-based classification*. Knowledge and Information Systems, págs. 1–23, 2015, ISSN 0219-3116.
- [71] Ho, T. K.: *Complexity of Classification Problems and Comparative Advantages of Combined Classifiers*. En *Multiple Classifier Systems*, vol. 1857 de *Lecture Notes in Computer Science*, págs. 97–106. Springer Berlin Heidelberg, 2000, ISBN 978-3-540-67704-8.
- [72] Ho, T. K.: *Data Complexity Analysis for Classifier Combination*. En *Multiple Classifier Systems*, vol. 2096 de *Lecture Notes in Computer Science*, págs. 53–67. Springer Berlin Heidelberg, 2001, ISBN 978-3-540-42284-6.

- [73] Ho, T. K. y M. Basu: *Complexity measures of supervised classification problems*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3):289–300, Mar 2002, ISSN 0162-8828.
- [74] Hochberg, Y. y D. Rom: *Extensions of multiple testing procedures based on Simes' test*. Journal of Statistical Planning and Inference, 48(2):141–152, 1995.
- [75] Hodges, J., E. L. Lehmann y cols.: *Rank methods for combination of independent experiments in analysis of variance*. The Annals of Mathematical Statistics, 33(2):482–497, 1962.
- [76] Hyndman, R. J. y G. Athanasopoulos: *Forecasting: principles and practice*. OTexts, 2012. <https://www.otexts.org/fpp>.
- [77] Ihaka, R. y R. Gentleman: *R: A Language for Data Analysis and Graphics*. Journal of Computational and Graphical Statistics, 5(3):299–314, 1996.
- [78] Itakura, F.: *Minimum prediction residual principle applied to speech recognition*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 23(1):67–72, Feb 1975, ISSN 0096-3518.
- [79] Java, A. y E. S. Perlman: *Predictive Mining of Time Series Data*. En *American Astronomical Society Meeting Abstracts #200*, vol. 34 de *Bulletin of the American Astronomical Society*, pág. 741, may 2002.
- [80] Jeong, Y. y R. Jayaraman: *Support vector-based algorithms with weighted dynamic time warping kernel function for time series classification*. Knowledge-Based Systems, 75:184–191, 2015, ISSN 0950-7051.
- [81] Jeong, Y., M. K. Jeong y O. A. Omitaomu: *Weighted dynamic time warping for time series classification*. Pattern Recognition, 44(9):2231–2240, 2011, ISSN 0031-3203.
- [82] Joachims, T.: *Transductive Inference for Text Classification using Support Vector Machines*. En *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, págs. 200–209, June 27 - 30 1999.
- [83] Kaya, H. y S. GunduzOguducu: *A distance based time series classification framework*. Information Systems, 51:27–42, 2015, ISSN 0306-4379.

- [84] Keogh, E. y S. Kasetty: *On the need for time series data mining benchmarks: a survey and empirical demonstration*. *Data Mining and knowledge discovery*, 7(4):349–371, 2003.
- [85] Keogh, E. y C. A. Ratanamahatana: *Exact indexing of dynamic time warping*. *Knowledge and Information Systems*, 7(3):358–386, 2005, ISSN 0219-1377.
- [86] Keogh, E., Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei y C. A. Ratanamahatana: *The UCR Time Series Classification/Clustering Homepage*, 2011. [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/).
- [87] Kim, M.: *Semi-supervised learning of hidden conditional random fields for time-series classification*. *Neurocomputing*, 119:339–349, 2013, ISSN 0925-2312. *Intelligent Processing Techniques for Semantic-based Image and Video Retrieval*.
- [88] Kirkpatrick, C. D. y J. R. Dahlquist: *Technical analysis: the complete resource for financial market technicians*. FT Press, 2010.
- [89] Kovacs-Vajna, Z. M.: *A fingerprint verification system based on triangular matching and dynamic time warping*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1266–1276, Nov 2000, ISSN 0162-8828.
- [90] Kurbalija, V., M. Radovanović, Z. Geler y M. Ivanović: *The influence of global constraints on similarity measures for time-series databases*. *Knowledge-Based Systems*, 56:49–67, 2014.
- [91] Lebrecht, R., S. Iovleff, F. Langrognnet, C. Biernacki, G. Celeux y G. Govaert: *Rmixmod: The R Package of the Model-Based Unsupervised, Supervised, and Semi-Supervised Classification Mixmod Library*. *Journal of Statistical Software*, 67(1):1–29, 2015, ISSN 1548-7660.
- [92] Lei, H. y B. Sun: *A Study on the Dynamic Time Warping in Kernel Machines*. En *Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (SITIS)*, SITIS '07, págs. 839–845, Dec 2007.



- [93] Leskes, B.: *The Value of Agreement, a New Boosting Algorithm*. En *Learning Theory*, vol. 3559 de *Lecture Notes in Computer Science*, págs. 95–110. Springer Berlin Heidelberg, 2005, ISBN 978-3-540-26556-6.
- [94] Letouzey, F., F. Denis y R. Gilleron: *Learning From Positive and Unlabeled Examples*, págs. 71–85. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, ISBN 978-3-540-40992-2.
- [95] Levenshtein, V.I.: *Binary codes capable of correcting deletions, insertions, and reversals*. En *Soviet physics doklady*, vol. 10, págs. 707–710, 1966.
- [96] Li, M. y Z. Zhou: *SETRED: Self-training with Editing*. En *Advances in Knowledge Discovery and Data Mining*, vol. 3518 de *Lecture Notes in Computer Science*, págs. 611–621. Springer Berlin Heidelberg, 2005, ISBN 978-3-540-26076-9.
- [97] Li, X., B. Liu y S. Ng: *Learning to Identify Unexpected Instances in the Test Set*. En *International Joint Conferences on Artificial Intelligence*, vol. 7, págs. 2802–2807, 2007.
- [98] Li, X., B. Liu y S. Ng: *Negative training data can be harmful to text classification*. En *Conference on Empirical Methods in Natural Language Processing*, págs. 218–228, October 2010.
- [99] Lin, W., M. A. Orgun y G. J. Williams: *An overview of temporal data mining*. En *Proceedings of the 1st Australian data mining workshop*, págs. 83–90, 2002.
- [100] Lines, J. y A. Bagnall: *Time series classification with ensembles of elastic distance measures*. *Data Mining and Knowledge Discovery*, 29(3):565–592, 2014, ISSN 1573-756X.
- [101] Liu, B., Y. Dai, X. Li, W.S. Lee y P.S. Yu: *Building text classifiers using positive and unlabeled examples*. En *Third IEEE International Conference on Data Mining (ICDM) 2003*, págs. 179–186, 2003.
- [102] Liu, H., H. Zhu, F. Xu y S. Wang: *Intensity Classification of the Image Quality Metric Series Using AdaBoost and Co-training Method*. En *Inter-*

- national Symposium on Computer Science and Society (ISCCS) 2011*, págs. 134–137, July 2011.
- [103] Loog, M.: *Constrained Parameter Estimation for Semi-supervised Learning: The Case of the Nearest Mean Classifier*. En *Machine Learning and Knowledge Discovery in Databases*, vol. 6322 de *Lecture Notes in Computer Science*, págs. 291–304. Springer Berlin Heidelberg, 2010, ISBN 978-3-642-15882-7.
- [104] Lu, T. T.: *Fundamental limitations of semi-supervised learning*. Tesis de Licenciatura, University of Waterloo, Ontario, Canada, 2009.
- [105] Marteau, P. y S. Gibet: *On Recursive Edit Distance Kernels With Application to Time Series Classification*. *IEEE Transactions on Neural Networks and Learning Systems*, 26(6):1121–1133, June 2015, ISSN 2162-237X.
- [106] Marteau, P. F.: *Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318, Feb 2009, ISSN 0162-8828.
- [107] Marussy, K. y K. Buza: *SUCCESS: A New Approach for Semi-supervised Classification of Time-Series*. En Rutkowski, L., M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh y J. Zurada (eds.): *Artificial Intelligence and Soft Computing*, vol. 7894 de *Lecture Notes in Computer Science*, págs. 437–447. Springer Berlin Heidelberg, 2013, ISBN 978-3-642-38657-2.
- [108] Meesrikamolkul, W., V. Niennattrakul y C. A. Ratanamahatana: *Advances in Knowledge Discovery and Data Mining: 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part I*, cap. Shape-Based Clustering for Time Series Data, págs. 530–541. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, ISBN 978-3-642-30217-6.
- [109] Meng, J., L. Wu, X. Wang y T. Lin: *Granulation-based symbolic representation of time series and semi-supervised classification*. *Computers & Mathematics with Applications*, 62(9):3581–3590, 2011, ISSN 0898-1221.

- [110] Mercer, J.: *Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations*. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 209:415–446, 1909, ISSN 02643952.
- [111] Merz, C. J., D. C. S. Clair y W. E. Bond: *Semi-supervised adaptive resonance theory (SMART2)*. En *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, vol. 3, págs. 851–856, Jun 1992.
- [112] Mitchell, T. M.: *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997, ISBN 9780071154673.
- [113] Mohd, S., R. Ashish y D. Mohd: *Temporal Data Mining: An Overview*. International Journal of Engineering and Advanced Technology, 1:20–24, 2011.
- [114] Mueen, A., E. Keogh, Q. Zhu, S. Cash y B. Westover: *Exact Discovery of Time Series Motifs*, cap. 40, págs. 473–484. 2009.
- [115] Mueen, A., E. Keogh, Q. Zhu, S. S. Cash, M. B. Westover y N. Bigdely-Shamlo: *A disk-aware algorithm for time series motif discovery*. Data Mining and Knowledge Discovery, 22(1):73–105, 2011, ISSN 1573-756X.
- [116] Nanopoulos, A., R. Alcock y Y. Manolopoulos: *Feature-based classification of time-series data*. International Journal of Computer Research, 10(3):49–61, 2001.
- [117] Nguyen, H., W. Ng y Y. Woon: *Closed motifs for streaming time series classification*. Knowledge and Information Systems, 41(1):101–125, 2014, ISSN 0219-3116.
- [118] Nguyen, M. N., X. Li y S. Ng: *Positive unlabeled learning for time series classification*. En *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Two*, págs. 1421–1426. AAAI Press, 2011.
- [119] Nguyen, M. N., X. Li y S. Ng: *Ensemble based positive unlabeled learning for time series classification*. En *Database Systems for Advanced Applications*, págs. 243–257. Springer, 2012.

- [120] Niennattrakul, V., D. Wanichsan y C.A. Ratanamahatana: *Knowledge-Based Intelligent Information and Engineering Systems: 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12-14, 2007. Proceedings, Part II*, cap. Hand Geometry Verification Using Time Series Representation, págs. 824–831. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, ISBN 978-3-540-74827-4.
- [121] Niennattrakul, V., D. Wanichsan y C.A. Ratanamahatana: *New Frontiers in Applied Data Mining: PAKDD 2009 International Workshops, Bangkok, Thailand, April 27-30, 2009. Revised Selected Papers*, cap. Accurate Subsequence Matching on Data Stream under Time Warping Distance, págs. 156–167. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, ISBN 978-3-642-14640-4.
- [122] Oates, T., L. Firoiu y P.R. Cohen: *Clustering time series with hidden markov models and dynamic time warping*. En *Proceedings of the IJCAI-99 workshop on neural, symbolic and reinforcement learning methods for sequence learning*, págs. 17–21, 1999.
- [123] Oppenheim, A. V., R. W. Schafer y J. R. Buck: *Discrete-time Signal Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, second ed., 1999, ISBN 0-13-754920-2.
- [124] Ordonez, P., T. Oates, M. Lombardi, G. Hernandez, K. W. Holmes, J. Fackler y C. U. Lehmann: *Visualization of multivariate time-series data in a neonatal ICU*. *IBM Journal of Research and Development*, 56(5):7:1–7:12, 2012.
- [125] Petitjean, F., G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen y E. Keogh: *Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm*. *Knowledge and Information Systems*, 47(1):1–26, 2016, ISSN 0219-3116.
- [126] Povinelli, R., M. Johnson, A. Lindgren y J. Ye: *Time series classification using Gaussian mixture models of reconstructed phase spaces*. *IEEE Transactions on Knowledge and Data Engineering*, 16(6):779–783, June 2004, ISSN 1041-4347.

- [127] Povinelli, R. J.: *Time series data mining: identifying temporal patterns for characterization and prediction of time series events*. Tesis de Doctorado, Faculty of the Graduate School, Marquette University, 1999.
- [128] Pree, H., B. Herwig, T. Gruber, B. Sick, K. David y P. Lukowicz: *On general purpose time series similarity measures and their use as kernel functions in support vector machines*. Information Sciences, 281:478–495, 2014, ISSN 0020-0255.
- [129] Quinlan, J. R.: *Induction of Decision Trees*. Machine Learning, 1(1):81–106, 1986, ISSN 1573-0565.
- [130] Quinlan, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, ISBN 1-55860-238-0.
- [131] R Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. <http://www.R-project.org/>.
- [132] R Development Core Team: *Writing R Extensions*. R Foundation for Statistical Computing, Vienna, Austria. Informe técnico., 2012. <http://CRAN.R-project.org/doc/manuals/R-exts.html>.
- [133] Rabiner, L.: *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2):257–286, Feb 1989, ISSN 0018-9219.
- [134] Rakthanmanon, T., B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria y E. Keogh: *Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping*. En *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, págs. 262–270, New York, NY, USA, 2012. ACM, ISBN 978-1-4503-1462-6.
- [135] Ratanamahatana, C. A. y E. Keogh: *Everything you know about dynamic time warping is wrong*. En *Third Workshop on Mining Temporal and Sequential Data*, 2004.

- [136] Ratanamahatana, C.A. y D. Wanichsan: *Stopping Criterion Selection for Efficient Semi-supervised Time Series Classification*. En Lee, R. (ed.): *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, vol. 149 de *Studies in Computational Intelligence*, págs. 1–14. Springer Berlin Heidelberg, 2008, ISBN 978-3-540-70559-8.
- [137] Rodpongpun, S., V. Niennattrakul y C. A. Ratanamahatana: *Selective Subsequence Time Series clustering*. *Knowledge-Based Systems*, 35:361–368, 2012, ISSN 0950-7051.
- [138] Rodríguez, J. J. y C. J. Alonso: *Interval and Dynamic Time Warping-based Decision Trees*. En *Proceedings of the 2004 ACM Symposium on Applied Computing, SAC '04*, págs. 548–552. ACM, 2004, ISBN 1-58113-812-1.
- [139] Rodríguez, J. J., C. J. Alonso y H. Boström: *Learning First Order Logic Time Series Classifiers: Rules and Boosting*. En *Principles of Data Mining and Knowledge Discovery*, vol. 1910 de *Lecture Notes in Computer Science*, págs. 299–308. Springer Berlin Heidelberg, 2000, ISBN 978-3-540-41066-9.
- [140] Russell, N., L. Cribbin y T. B. Murphy: *upclass: Updated Classification Methods using Unlabeled Data*, 2014. <http://CRAN.R-project.org/package=upclass>, R package version 2.0.
- [141] Sakoe, H. y S. Chiba: *Dynamic programming algorithm optimization for spoken word recognition*. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.
- [142] Serrà, J. y J. L. Arcos: *An empirical evaluation of similarity measures for time series classification*. *Knowledge-Based Systems*, 67(0):305–314, 2014, ISSN 0950-7051.
- [143] Shahnawaz, M., A. Ranjan y M. Danish: *Temporal data mining: an overview*. *International Journal of Engineering and Advanced Technology*, 1(1), 2011.
- [144] Shumway, R. H. y D. S. Stoffer: *Time series analysis and its applications: with R examples*. Springer Science & Business Media, third ed., 2010, ISBN 978-1-4419-7864-6.

- [145] Silva, S. da, M. Dias Júnior, V. Lopes Junior y M. J. Brennan: *Structural damage detection by fuzzy clustering*. Mechanical Systems and Signal Processing, 22(7):1636–1649, 2008, ISSN 0888-3270.
- [146] Tax, D. M. y R. P. Duin: *Uniform object generation for optimizing one-class classifiers*. The Journal of Machine Learning Research, 2:155–173, 2002.
- [147] Thanh Vinh, V. y D. Tuan Anh: *Some Novel Improvements for MDL-Based Semi-supervised Classification of Time Series*. En *Computational Collective Intelligence. Technologies and Applications*, vol. 8733 de *Lecture Notes in Computer Science*, págs. 483–493. Springer International Publishing, 2014.
- [148] Torgo, L.: *Data Mining with R, learning with case studies*. Chapman and Hall/CRC, 2010.
- [149] Tormene, P., T. Giorgino, S. Quaglini y M. Stefanelli: *Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation*. Artificial Intelligence in Medicine, 45(1):11–34, 2009, ISSN 0933-3657.
- [150] Triguero, I., S. García y F. Herrera: *Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study*. Knowledge and Information Systems, 42(2):245–284, 2015, ISSN 0219-1377.
- [151] Triguero, I., J. A. Sáez, J. Luengo, S. García y F. Herrera: *On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification*. Neurocomputing, 132:30–41, 2014, ISSN 0925-2312.
- [152] Tsai, C., C. Chen y C. Chien: *A time-interval sequence classification method*. Knowledge and Information Systems, 37(2):251–278, 2013, ISSN 0219-3116.
- [153] Vapnik, V.: *The nature of statistical learning theory*. Springer Science & Business Media, second ed., 2013, ISBN 978-1-4419-3160-3.
- [154] Wang, X., A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann y E. Keogh: *Experimental comparison of representation methods and distance measures for time series data*. Data Mining and Knowledge Discovery, 26(2):275–309, 2013.

- [155] Wang, Y., X. Xu, H. Zhao y Z. Hua: *Semi-supervised learning based on nearest neighbor rule and cut edges*. Knowledge-Based Systems, 23(6):547–554, 2010, ISSN 0950-7051.
- [156] Wei, L.: *Datasets used for experimental evaluation in the paper: Semi-Supervised Time Series Classification*, 2006. [www.cs.ucr.edu/~wli/selfTraining/](http://www.cs.ucr.edu/~wli/selfTraining/).
- [157] Wei, L. y E. Keogh: *Semi-supervised Time Series Classification*. En *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, págs. 748–753, 2006.
- [158] Weng, X. y J. Shen: *Classification of multivariate time series using two-dimensional singular value decomposition*. Knowledge-Based Systems, 21(7):535–539, 2008, ISSN 0950-7051.
- [159] Witten, I.H., E. Frank y M.A. Hall: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, third ed., 2011, ISBN 978-0-12-374856-0.
- [160] Witten, I.H., E. Frank y M.A. Hall: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, Boston, third edition ed., 2011, ISBN 978-0-12-374856-0.
- [161] Wu, X., V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.H. Zhou, M. Steinbach, D.J. Hand y D. Steinberg: *Top 10 algorithms in data mining*. Knowledge and Information Systems, 14(1):1–37, 2007, ISSN 0219-3116.
- [162] Xi, X., E. Keogh, C. Shelton, L. Wei y C.A. Ratanamahatana: *Fast Time Series Classification Using Numerosity Reduction*. En *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, págs. 1033–1040, New York, NY, USA, 2006. ACM, ISBN 1-59593-383-2.
- [163] Xing, Z., J. Pei y P.S. Yu: *Early classification on time series*. Knowledge and Information Systems, 31(1):105–127, 2012, ISSN 0219-3116.
- [164] Yamada, Y., E. Suzuki, H. Yokoi y K. Takabayashi: *Decision-tree induction from time-series data based on a standard-example split test*. En *Twentieth*



- International Conference on Machine Learning*, vol. 3 de *ICML '03*, págs. 840–847, Aug 2003.
- [165] Yarowsky, D.: *Unsupervised word sense disambiguation rivaling supervised methods*. En *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, págs. 189–196. Association for Computational Linguistics, 1995.
- [166] Yu, S., B. Krishnapuram, R. Rosales y R.B. Rao: *Bayesian Co-Training*. *Journal of Machine Learning Research*, 12:2649–2680, nov 2011, ISSN 1532-4435.
- [167] Yu, S. y C. Li: *PE-PUC: A graph based PU-learning approach for text classification*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4571 LNAI:574–584, 2007.
- [168] Zhang, B. y W. Zuo: *Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples*. *Journal of Computers*, 4(1):94–101, 2009.
- [169] Zhang, D., W. Zuo, D. Zhang y H. Zhang: *Time Series Classification Using Support Vector Machine with Gaussian Elastic Metric Kernel*. En *20th International Conference on Pattern Recognition (ICPR)*, ICPR '10, págs. 29–32, Aug 2010.
- [170] Zhong, S.: *Semi-supervised sequence classification with HMMs*. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02):165–182, 2005.
- [171] Zhou, Y. y S. Goldman: *Democratic co-learning*. En *IEEE 16th International Conference on Tools with Artificial Intelligence (ICTAI)*, págs. 594–602. IEEE, Nov 2004.
- [172] Zhou, Z. y M. Li: *Tri-training: exploiting unlabeled data using three classifiers*. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, Nov 2005, ISSN 1041-4347.

- [173] Zhu, X. y A. B. Goldberg: *Introduction to Semi-Supervised Learning*. Morgan & Claypool Publishers, 2009.
- [174] Zighed, D. A., S. Lallich y F. Muhlenbach: *Principles of Data Mining and Knowledge Discovery: 6th European Conference, PKDD 2002 Helsinki, Finland, August 19–23, 2002 Proceedings*, cap. Separability Index in Supervised Learning, págs. 475–487. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, ISBN 978-3-540-45681-0.