



Universidad de Valladolid

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Matemáticas

TÉCNICAS DE AGRUPACIÓN SOBRE SERIES TEMPORALES PARA LA DETECCIÓN DE ANOMALÍAS EN UN ENTORNO INDUSTRIAL

Autor: Adrián Manzano Santos

Tutor: Pedro César Álvarez Esteban

Índice general

1. Introducción y caso de estudio	10
1.1. Descripción del objeto de estudio	12
1.1.1. El trabajo diario	12
1.1.2. Los sensores	13
1.2. Objetivos y formulación del problema	14
2. Fundamentos sobre series temporales	15
2.1. El concepto de serie temporal	15
2.1.1. Introducción	15
2.1.2. Series temporales con modelo estacionario	16
2.1.3. Componentes de una serie temporal. El modelo aditivo	21
2.1.4. Técnicas para determinar la tendencia. El algoritmo de medias móviles	24
2.2. Métrica sobre series temporales. La distancia DTW	28
2.2.1. Introducción. El concepto de deformación en el tiempo	28
2.2.2. Caminos de deformación su coste	30
2.2.3. La distancia DTW	32
2.2.4. Algoritmo para el cálculo de la distancia DTW. La matriz DTW	36
2.2.5. Algoritmo para hallar la curva de deformación	40
2.2.6. Algoritmo para determinar las asociaciones de índices	42
2.2.7. Otras versiones del algoritmo DTW	44
2.3. Notas finales sobre la distancia DTW	50
2.3.1. Expansión por deformación de una serie temporal	51
2.3.2. La identificación por deformación	54
3. Fundamentos sobre <i>clustering</i>	57
3.1. Algoritmos de clasificación no supervisada (<i>clustering</i>)	57
3.1.1. El algoritmo de k-medias	60
3.2. El algoritmo de k-medias aplicado a series temporales	63
3.2.1. La media de Fréchet	63
3.2.2. Construcción del algoritmo de k-medias aplicable a series temporales	66
3.3. Principales problemas asociados al clustering de k-medias	68
3.4. Los diferentes enfoques en la clasificación de series temporales	70

4. Caso de estudio: Sistema de detección de anomalías de la máquina fresadora CNC	71
4.1. Análisis de la información	71
4.2. Arquitectura del Modelo	75
4.3. Estudio experimental de los parámetros	76
4.3.1. La configuración para la clasificación	77
4.4. Resultados de la clasificación	78
4.4.1. Evaluación del modelo	81
4.4.2. Ventajas del modelo híbrido	82
5. Conclusiones	84
A. Información extra sobre el caso de estudio	87
A.1. Relaciones de los materiales, operaciones y herramientas	87
A.1.1. La base de datos	88
B. Implementación en pseudocódigo de los algoritmos	92
B.1. Implementación del algoritmo de medias móviles	92
B.2. Implementación del algoritmo DTW	93
B.2.1. Implementación del algoritmo DTW versión primera	93
B.2.2. Implementación del algoritmo de cálculo de la curva de deformación	95
B.2.3. Implementación del algoritmo de asociaciones de índices	96
B.2.4. Implementación del algoritmo DTW en la versión segunda	96
B.3. Implementación del algoritmo DBA	98
B.4. Implementación del algoritmo de clustering sobre vectores	98
B.5. Implementación del algoritmo de clustering sobre series temporales	102
C. Ejemplo de realizaciones de ciertas actividades	105
D. Resultados del estudio experimental	111

Índice de figuras

2.1. Ejemplo de serie temporal con modelo estacionario de ruido blanco	18
2.2. Ejemplo de series temporales con modelo no estacionario de camino aleatorio . .	18
2.3. Ejemplo de la tendencia de una serie temporal	22
2.4. Serie temporal definida como $x_k = 2,75k + 10 \cdot \text{sen}(k) + 10 \quad k = 1, 2, \dots, 51$	24
2.5. Ejemplo de serie temporal con tendencia determinista (izquierda) y con tendencia evolutiva (derecha)	25
2.6. Comparativa de la serie X y de la tendencia T real con las series calculada por medias móviles de orden 7, Y , y de orden 3, Z	28
2.7. Ejemplo del efecto de deformación en el tiempo	29
2.8. Comparación de la asociación cronológica (A) con una asociación con deformación en el tiempo (B). Las líneas grises punteadas indican las asociaciones.	34
3.1. Ejemplo de clustering con la flor de iris: representación de los datos (izquierda) y resultado del algoritmo (derecha)	63
3.2. Representación de las series temporales a promediar (<i>a</i>) y del promedio obtenido mediante el algoritmo DBA (<i>b</i>).	64
3.3. Representación de las series temporales X , Y y Z con sus asociaciones de índices respecto de X (izquierda) y junto a su media C (derecha)	66
3.4. Ejemplo de agrupaciones complejas, cuya solución falla por k-medias.	69
3.5. Ejemplo de clustering incorrectos con la flor de iris: agrupación con dos clústeres (izquierda) y agrupación con los clústeres iniciales erróneos (derecha)	69
4.1. Representación de una operación de contorneado sobre el acero	72
4.2. Ocurrencias de cada tipo de operación con cada material	72
4.3. Histograma de las duraciones de las operaciones sobre el acero con duración inferior a una hora y 30 minutos (izquierda) y diagrama de barras con las duraciones agrupadas (derecha)	73
4.4. Diagrama de barras del número de operaciones resultantes: en azul sin imponer ningún tipo de restricción, en naranja tras exigir que la duración sea mayor de 5 minutos y menor de una hora y en verde tras exigir también que sean posteriores al 1 de o de 2019	74

4.5. Agrupaciones de los arranques según su duración (izquierda) y diagrama de barras del número de operaciones resultantes (derecha): en azul sin imponer ningún tipo de restricción, en naranja tras exigir que la duración sea mayor de 3 minutos y menor de una hora y en verde tras exigir también que sean posteriores al 1 de septiembre de 2019	74
4.6. Esquema del modelo de clasificación híbrido planteado	76
4.7. Ejemplo de aplicación del método del codo. La solución es de 5 clústeres.	77
4.8. Representación de la distancia intra-clúster para diferentes configuraciones de número de clústeres para las duraciones, para el tipo de operación de contorneado sobre acero.	78
4.9. Representación de la distancia intra-clúster según el modelo descrito, para la operación de contorneado sobre acero.	79
4.10. Representación del porcentaje de elementos asociados a cada clúster respecto del total clasificados para cada <i>opermat</i>	80
4.11. Representación del porcentaje de operaciones consideradas normales y anómalas para diferentes valores de tolerancia discriminando por el <i>opermat</i>	80
4.12. Representación de los resultados de evaluar las actividades realizadas con sus correspondientes modelos generados, para la totalidad de los datos, los datos de entrenamiento (septiembre-19 en adelante) y los datos de prueba (de mayo-19 a septiembre-19)	81
4.13. Representación de los tiempos requeridos en la construcción de modelos con diferentes configuraciones de la operación de contorneado sobre el acero. Cada representación se corresponde con un valor fijo de clústeres para la duración para cada número de clústeres para la clasificación bajo DTW.	82
A.1. Equipo con pantalla táctil con el sistema de control instalado	90
C.1. Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo planeado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente	106
C.2. Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo contorneado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente	107
C.3. Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo fresado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente	108
C.4. Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo taladrado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente	109
C.5. Representación de las mediciones realizadas por el sensor de corriente de fase 1 para el arranque de la máquina fresadora y resultado de clasificar dichas realizaciones por el modelo correspondiente	110

D.1. Representación de la distancia intra-clúster bajo la DTW como función objetivo para la operación de taladrado sobre acero	111
D.2. Representación de la distancia intra-clúster bajo la DTW como función objetivo para la operación de fresado sobre acero.	112
D.3. Representación de la distancia intra-clúster bajo la DTW como función objetivo para la operación de planeado sobre acero	113
D.4. Representación de la distancia intra-clúster bajo la DTW como función objetivo para los arranques de la máquina fresadora	114

Índice de tablas

1.1. Operaciones realizables en la fresadora y sus códigos	12
1.2. Materiales de uso en la fresadora y sus códigos	12
1.3. Unidades de medida y precisión de los sensores	14
2.1. Comparativa de la serie X y de la tendencia T real con las series calculada por medias móviles de orden 7, Y , y de orden 3, Z	27
A.1. Herramientas de la fresadora y sus códigos	87
A.2. Relación de las operaciones, herramientas y materiales	88
A.3. Campos e información en la tabla CNC_Funcionamiento	89
A.4. Campos e información de la tabla <i>tiempoOperaciones</i>	90
C.1. Resultados de evaluar las actividades seleccionadas como ejemplo por su correspondiente modelo según los identificadores dados en las <i>Figuras C.1, C.2, C.3, C.4 y C.5</i>	105

Agradecimientos

A mi familia, en especial a mis padres, por su apoyo mostrado estos años. A la Fundación Cidaut y a mis compañeros, por proponerme y ayudarme con este proyecto. Y, especialmente, a Pedro César, por aceptar la dirección de este trabajo y por toda su ayuda prestada.

Resumen

En plena revolución digital, ha cobrado una gran importancia la digitalización de procesos y sistemas a un mundo virtual, donde se busca extraer nueva información que los optimicen. En la fundación CIDAUT, dentro del proyecto IDIGIT4L, se está trabajando en la sensorización no intrusiva de determinadas infraestructuras y recursos con el objetivo de mejorar su rendimiento. Una vez extraída la información del mundo físico al mundo virtual, esta debe ser procesada y analizada para extraer conclusiones y, en última instancia, mejorar el mundo físico.

En este sentido, el presente documento tiene como objetivo principal resolver el problema de *la detección de anomalías sobre un conjunto de datos estructurados como series temporales*, para proponer una solución basada en la clasificación de información no supervisada o *clustering* (es decir, del dato a clasificar solo se dispone, como información, del propio dato) y la modelización de los procesos virtualizados mediante series temporales. El algoritmo de clasificación elegido es el algoritmo de k-medias, y en consecuencia, para adaptar y aplicar dicho algoritmo sobre series temporales son necesarios dos requisitos fundamentales: una medida de comparación y una función promedio.

Satisfaciendo estos requisitos, se estudia la distancia *Dynamic Time Warping* (DTW) como medida de comparación y la *media de Fréchet a través de la DTW* como función promedio, capaces de “deformar” la línea temporal de dos series para encontrar las mejores coincidencias entre los eventos registrados.

Todo lo anterior requiere de un profundo estudio teórico sobre series temporales y el algoritmo k-medias, para ser aplicado sobre una máquina Fresadora CNC modelo Nicolás Correa sensorizada de uso diario en el taller de la fundación CIDAUT en el parque tecnológico de Boecillo.

Palabras clave: serie temporal, proceso estocástico, *Dynamic Time Warping* (DTW), clasificación no supervisada (*clustering*), algoritmo, algoritmo de k-medias, algoritmo de medias móviles.

Capítulo 1

Introducción y caso de estudio

Actualmente estamos inmersos en la *Cuarta Revolución Industrial*, también conocida como *Industria 4.0*¹, que está considerada la etapa más importante desde la revolución industrial del siglo XVIII. Esta cuarta etapa está cambiando la forma en la que los negocios operan y los entornos en los que compiten. El concepto de Cuarta Revolución Industrial es bastante actual y fue acuñado por Klaus Schwab² en el Foro Económico Mundial de 2016, siendo las primeras referencias a dicha idea posteriores al año 2010.

Mientras que la Tercera Revolución Industrial está caracterizada por la *era digital*, la industria 4.0 busca combinar técnicas avanzadas de producción con las nuevas tecnologías tales como la *robótica*, la *Inteligencia Artificial* o el *Internet de las Cosas* entre otros, recogiendo la tendencia actual a la automatización y el intercambio de información. Para mejorar los procedimientos industriales, se comienza con una captura de la información del entorno físico traduciéndola a un entorno digital. En dicho entorno digital, la información es interpretada de forma analítica y analizada objetivamente para extraer unos resultados o conclusiones que finalmente se traducen en decisiones en el mundo real.

El presente trabajo de fin de grado se engloba dentro del proyecto **IDIGIT4L**³ de la *Fundación Cidaut* cuyo objetivo principal es la investigación de un ecosistema de digitalización en el cual intervienen personas, objetos y sistemas con el fin de transformar procesos industriales de forma inteligente y predictiva. Gracias a la convergencia del mundo físico y del mundo digital, se presenta el concepto de *Gemelo Digital distribuido*, basado en la virtualización de activos y el conocimiento adaptativo de las operaciones que permita mejorar los procesos y la calidad de los productos.

Con este objetivo, se han implementado soluciones (consúltese [16] y [17]) apoyadas en la

¹véase [18]

²véase [19]

³“I+D para la transformación de los sectores industriales e infraestructuras: Digitalización y su implementación inteligente y predictiva iDIGIT4L”, N.º de Exp.: CCTT1/17/VA/0007. Proyecto financiado por el Instituto para la Competitividad Empresarial de Castilla y León (Junta de Castilla y León), cofinanciado con Fondos FEDER, a través de la línea: “PROYECTOS I+D APLICADOS REALIZADOS POR CCTT. Convocatoria 2017”.

retroactualización y convergencia de activos industriales, independientemente de su nivel de digitalización, con el uso de sensores no intrusivos. Con ello, se consigue extraer y digitalizar cierta información de un entorno industrial sin afectar a su funcionamiento ni requerir modificaciones sobre los procesos industriales.

Una vez extraída la información y cargada en un entorno informático esta debe ser procesada y analizada, con el objetivo de extraer conclusiones. Este trabajo de fin de grado busca dar una solución a esta problemática, que a su vez se descompone en dos grandes cuestiones:

1. Elegir la técnica para la adecuada representación de la información extraída por los sensores.
2. Elegir las técnicas para el procesamiento de la información estructurada.

Para la primera de las cuestiones anteriores, se opta por modelizar la información mediante *series temporales* (véase *Definición 2.2*), cuyos fundamentos se presentan en el *capítulo 2* de esta memoria. Y, para la segunda cuestión, se suele recurrir a técnicas y algoritmos pertenecientes a la rama del *Machine Learning*, cuya elección depende de los objetivos perseguidos. En este proyecto, el objetivo final es la detección de procesos anómalos, y por ello se va a recurrir a técnicas de aprendizaje no supervisado o *clustering* (véase *Definición 3.1*), desarrollado en el *capítulo 3* de esta memoria.

Esta memoria ha sido estructurada como sigue:

- El *Capítulo 1* introduce el caso de estudio y los objetivos perseguidos.
- Los *Capítulos 2 y 3* se corresponden con el estudio teórico de las técnicas y algoritmos aplicados. El *Capítulo 2* se dedica al estudio de las series temporales (*Sección 2.1*) y de la medida de comparación *Dynamic Time Warping* (*Secciones 2.2 y 2.3*) y sus principales consecuencias asociadas.
- En el *Capítulo 3* se presentan las técnicas de clasificación de aprendizaje no supervisado o *clustering*, en particular el algoritmo de k-medias (*k-means*), en su versión tradicional (*Sección 3.1*) y las modificaciones necesarias para su aplicación sobre series temporales, bajo la distancia DTW (*Sección 3.2*).
- Finalmente, en el *Capítulo 4* se muestra un ejemplo de aplicación práctica de la teoría desarrollada sobre un caso de estudio real, la máquina fresadora CNC sensorizada de la Fundación CIDAUT. Por tanto, en este capítulo se muestran todos los detalles del estudio realizado para elegir el modelo de clasificación y los resultados obtenidos con el modelo elegido.

1.1. Descripción del objeto de estudio

En la fundación CIDAUT disponen de una máquina fresadora de control numérico por computadora (CNC) modelo Nicolás Correa CF20-18 de 1994 situada en las instalaciones de la fundación en el Parque Tecnológico de Boecillo.

Esta fresadora se encuentra en pleno funcionamiento y está dedicada a la fabricación de piezas prediseñadas bajo demanda. Ha sido elegida para el estudio por sus características: equipamiento de más de 20 años de servicio, no digitalizada y con órdenes de fabricación de diversa naturaleza donde cada pieza o producto es único y diferente al resto.

1.1.1. El trabajo diario

La máquina fresadora CNC permite realizar siete **operaciones** (véase *Tabla 1.1*) sobre cinco **materiales** (véase *Tabla 1.2*) diferentes. Una pieza es la realización de una o varias operaciones sobre un material concreto. Cada operación y cada material son identificadas por un código numérico único, reservando el 0 para el estado ocioso de la máquina (sin material ni operación).

Además, para cada operación y cada material se pueden utilizar una serie de herramientas (que dependen de ciertos detalles del diseño y estructura de la pieza que carecen de interés en el estudio a realizar). Para más información sobre las herramientas consúltese el *Sección A.1*.

Operación	Código
Toma de ceros	1
Planeado	2
Contorneado o copiado	3
Mandrinado	4
Fresado	5
Taladrado	6
Especiales	7

Tabla 1.1: Operaciones realizables en la fresadora y sus códigos

Material	Código
Plástico	1
Aluminio	2
Acero	3
Acero Inox o 316	4
Otros	5

Tabla 1.2: Materiales de uso en la fresadora y sus códigos

El diseño y producción de piezas en la máquina fresadora es bajo demanda, es decir, cada pieza diseñada y producida lo es para cubrir una necesidad concreta y es poco probable que se vuelva a producir otra pieza igual en un corto periodo de tiempo. Por tanto, a la hora de comparar los datos de dos operaciones en la máquina hay que tener en cuenta que no son sobre el mismo diseño (plano) aunque se trate del mismo material. Por ejemplo, en la operación de taladrado sobre acero un factor importante es el grosor del material bruto, que influye especialmente sobre la duración de la operación. Por tanto, dos operaciones de diferente duración pueden ser “iguales” en cuanto a comportamiento se refiere.

1.1.2. Los sensores

Para la recolección de información sobre el funcionamiento de la máquina se dispone de una instrumentación de sensores capaces de realizar mediciones no intrusivas (es decir, el propio sensor no altera el funcionamiento de la componente ni su medición). Estos sensores miden el consumo, posición de la fresa, temperatura y sonido producido.

- **Acelerómetro:** determina la posición en la que se encuentra el mandrino respecto de un origen preestablecido sobre tres ejes con datos de cada coordenada en el espacio. De forma que el acelerómetro registra la posición en tres ejes denominadas respectivamente acelerómetro-eje-X, acelerómetro-eje-Y y acelerómetro-eje-Z.
- **Sonda de temperatura:** mide la temperatura en la zona de trabajo interior de la fresadora.
- **Sonda trifásica :** determina el consumo de la máquina fresadora. La máquina fresadora dispone de tres *relés* que realizan un consumo de corriente. Cada relé dispone de su propio sensor de corriente, denominados respectivamente sensor de fase de corriente 1, 2 y 3.
- **Detector de sonido:** mide los niveles de presión sonora (ruido) producido por la máquina fresadora en la zona interior de trabajo.
- **Sensor de la puerta:** determina si la puerta que delimita la zona de trabajo se encuentra abierta o cerrada.

Todos los sensores realizan mediciones con una cadencia de un segundo, independientemente de la hora, operación realizada o estado de la máquina. Estas mediciones son almacenadas en una tabla de la base de datos SQL que se describe con detalle en la *Sección A.1.1*. En la *Tabla 1.3* se recoge la precisión de cada sensor.

Sensor	Unidad de medida	Precisión
Acelerómetro	Metros	0.005
Sensor de fase de corriente	Amperios	0.01
Sonda de temperatura	Grados centígrados	1
Detector de sonido	Decibelios	0.01

Tabla 1.3: Unidades de medida y precisión de los sensores

1.2. Objetivos y formulación del problema

El presente Trabajo de Fin de Grado está motivado por el proyecto IDIGIT4L, y busca cubrir la parte de virtualización y análisis de la información recogida por la instrumentación de sensores sobre la máquina fresadora CNC (*Sección 1.1*).

El objetivo principal es determinar si una operación realizada presenta alguna anomalía, bajo la hipótesis de que toda operación del mismo tipo y sobre el mismo material comparten un “modelo” común. Para alcanzar dicho objetivo, se hace necesario estructurar la información recolectada (como *series temporales*) y posteriormente compararla (mediante *clustering de k-medias sobre series temporales*).

Por tanto, el problema a resolver es *la detección de anomalías sobre un conjunto de datos estructurados como series temporales*. En primer lugar, cada operación realizada es representada como una serie temporal dada por las mediciones realizadas por el sensor de corriente de fase 1. Considerando que la serie temporal asociada es la única información conocida de la operación realizada (además de su tipo de operación y del material utilizado) entonces, para cada tipo de operación con cada material se aplica el algoritmo de clasificación no supervisada de k-medias para agrupar (en subconjuntos o clústeres) las diferentes realizaciones, considerando anómalas las operaciones cuyo clúster asociado tenga “pocas” operaciones respecto del total.

Para aplicar un algoritmo de k-medias son necesarios dos requisitos fundamentales: una medida de comparación y una función promedio. Además, deben tener en cuenta dos peculiaridades de las series temporales que representan las operaciones: la longitud de las series temporales es variable y un mismo evento puede ocurrir en momentos temporales diferentes con distinta duración (por ejemplo, la duración en la operación de taladrado depende del grosor del material). Satisfaciendo estos requisitos, se estudia la distancia *Dynamic Time Warping* (DTW) como medida de comparación y la *media de Fréchet* como función promedio, capaces de “deformar” la línea temporal de dos series para encontrar las mejores coincidencias entre los eventos registrados.

Por último, se debe definir un modelo de clasificación adecuado, que minimice los efectos negativos del trabajo con series temporales (como el elevado coste de tiempo consumido en su creación), y explote las bondades de las series temporales y la DTW. Por otro lado, todo modelo requiere de unos parámetros iniciales, que deben ser determinados adecuadamente, pues su influencia en los resultados obtenidos es notable.

Capítulo 2

Fundamentos sobre series temporales

En este capítulo se introducen los conceptos necesarios sobre series temporales y el modelo asociado, y las principales técnicas para la descomposición de una serie temporal. Posteriormente, se presenta una medida de comparación de similitud de series temporales denominada *Dynamic Time Warping*, capaz de identificar acciones iguales que suceden en momentos diferentes y/o se realizan a diferentes velocidades. Por último, se estudian las debilidades de dicha medida de comparación y los ajustes necesarios convertirla en una semi-métrica.

2.1. El concepto de serie temporal

En esta sección se introducen los conceptos fundamentales sobre series temporales (*Definición 2.1, 2.2 y 2.6*), así como las principales técnicas y modelos que se aplicaran en el análisis de series temporales. En particular, se centrará en el estudio de un modelo de descomposición de series temporales denominado *modelo aditivo* (*Definición 2.17* y en las técnicas para determinar la *tendencia* (*Definición 2.14*) de una serie temporal. Esta primera sección está basada en los capítulos primero y segundo de [2] y en los capítulos octavo y noveno de [3].

2.1.1. Introducción

En esta sección se definen los conceptos de serie temporal y su modelo (*de 2.2*) y se concluye con una primera clasificación de las series temporales.

Definición 2.1. Se define un **proceso estocástico** o **proceso aleatorio** como una familia de variables aleatorias $\{X_i : i \in T, X_i : \Omega \rightarrow \mathcal{S}\}$, con $T \subseteq \mathbb{R}$, que representan el espacio muestral Ω en el conjunto \mathcal{S} .

Al conjunto T se le denomina **conjunto de índices** y a \mathcal{S} se le denomina **espacio de estados**.

Generalmente, \mathcal{S} suele ser el conjunto de los reales \mathbb{R} y T se toma como un conjunto discreto (los naturales \mathbb{N}) o como un intervalo.

Definición 2.2. Sea $\{X_i : i \in T, X_i : \Omega \rightarrow \mathcal{S}\}$ un proceso estocástico, una **serie temporal** es una realización de cada una de las variables aleatorias que componen el proceso estocástico. Es decir, es una familia de la forma

$$\mathbf{X} = \{X_i(\omega_i) : i \in T, \omega_i \in \Omega\}.$$

En esta situación, se dice que el proceso estocástico $\{X_i : i \in T, X_i : \Omega \rightarrow \mathcal{S}\}$ es el **modelo de la serie temporal \mathbf{X}** .

Por tanto, una serie temporal es un conjunto de observaciones x_t recogidas en un tiempo específico t según un modelo que es un proceso estocástico. Entonces el conjunto de índices T indica los momentos en los que se realizan las mediciones. Dependiendo de cómo se tome dicho conjunto se tienen dos tipos de series temporales.

Definición 2.3. Sea $\mathbf{X} = \{X_i(\omega_i) : i \in T, \omega_i \in \Omega\}$ una serie temporal,

1. Se dice que \mathbf{X} es una serie temporal (o un proceso estocástico) de **tiempo discreto** si el conjunto de índices T es discreto.
2. Se dice que \mathbf{X} es una serie temporal (o un proceso estocástico) de **tiempo continuo** si el conjunto de índices T es un intervalo de la recta real.

El caso más común para series de tiempo discreto es $T = \mathbb{N}$ (ó $T = \mathbb{N} \cup \{0\}$) y para series de tiempo continuo es $T = [0, 1]$, donde, reescalando adecuadamente, 0 es el momento inicial en el que se realizan las observaciones y 1 es el momento final.

Notación. Sea $\mathbf{X} = \{X_i(\omega_i) : i \in T, \omega_i \in \Omega\}$ una serie temporal de tiempo discreto, con $T \subset \mathbb{N}$ finito de cardinal $n > 0$ natural, entonces si $x_i = X_i(\omega_i)$ para cada $i \in T$ se representa la serie temporal como $\mathbf{X} = [x_1, \dots, x_n]$.

Ejemplo 2.4. Proceso binario. Sea $T = \{1, \dots, n\}$ y sean X_1, \dots, X_n variables aleatorias independientes e igualmente distribuidas como Bernoulli de parámetro $\frac{1}{2}$. Para $n = 5$, una serie temporal con modelo $\{X_i : i = 1, \dots, 5\}$ es $[1, 0, 1, 0, 1]$. Según el modelo, la probabilidad de aparición de cada posible serie es de $\frac{1}{32}$.

2.1.2. Series temporales con modelo estacionario

En esta sección se va a estudiar un tipo de serie temporal cuyo modelo presenta unas características muy concretas que facilitarán su estudio.

Notación. Sean $X, Y : \Omega \rightarrow \mathbb{R}$ dos variables aleatorias reales, se denotan la esperanza de X por $\mathbb{E}(X)$, la varianza de X como $Var(X) = \mathbb{E}((X - \mathbb{E}X)^2)$ y la covarianza de X e Y por $Cov(X, Y) = \mathbb{E}((X - \mathbb{E}X)(Y - \mathbb{E}Y))$.

Definición 2.5. Sea $\{X_i : i \in T\}$ un modelo de serie temporal con $\mathbb{E}(X_i^2)$ finita para cada $i \in T$, se dice que el modelo es **fuertemente estacionario** si para cada subconjunto finito $\{s_1, \dots, s_n\} \subseteq T$ y para cada $h > 0$ tal que $\{s_1 + h, \dots, s_n + h\} \subseteq T$ se verifica que el vector aleatorio $(X_{s_1}, \dots, X_{s_n})$ esta igualmente distribuido que $(X_{s_1+h}, \dots, X_{s_n+h})$.

Definición 2.6. Sea $\{X_i : i \in T\}$ un modelo de serie temporal con $\mathbb{E}(X_i^2)$ finita para cada $i \in T$, se dice que el modelo es **débilmente estacionario** (o simplemente **estacionario**) si existen $\mu, \sigma \in \mathbb{R}$ y un subconjunto real $\{a_t : t \in T\}$ tales que verifican:

- *Media constante:* $\mathbb{E}(X_t) = \mu, \forall t \in T$
- *Varianza constante:* $Var(X_t) = \sigma, \forall t \in T$
- *Autocovarianza:* $Cov(X_t, X_{t+h}) = a_h, \forall t \in T, \forall h \neq 0$ tal que $t + h \in T$

En caso de que alguna de las anteriores condiciones no se verifique, se dice que la serie es **no estacionaria**.

Con abuso del lenguaje, se dice que una serie es estacionaria cuando así lo sea su modelo. La condición de ser estacionaria implica que la media y varianza no dependen de la variable aleatoria del modelo considerado. Por otro lado, la propiedad de la autocovarianza determina que la covarianza de dos variables aleatorias cualesquiera que forman parte del modelo depende exclusivamente de lo distantes que se encuentren en el tiempo.

Ejemplo 2.7. *Ruido blanco (white noise).* El ejemplo básico de serie estacionaria es el *ruido blanco*, que se caracteriza por tener modelo $\{X_i : i = 1, \dots, n\}$ con medias y covarianzas constantemente nulas y varianza constante. El caso más usual es cuando X_1, \dots, X_n son variables aleatorias independientes e igualmente distribuidas $N(0, \sigma^2)$.

Ejemplo 2.8. *Camino aleatorio (random walk).* El camino aleatorio $\{S_t : t = 0, 1, \dots\}$ se obtiene de la acumulación de variables aleatorias independientes e igualmente distribuidas. Es decir, sea $(X_n)_{n=1}^{\infty}$ una sucesión de variables aleatorias independientes e igualmente distribuidas entonces

$$S_n = \sum_{i=1}^n X_i$$

Supongamos que X_i está distribuida como una Bernoulli de parámetro $\frac{1}{2}$, esto es,

$$P(X_i = 0) = \frac{1}{2} \text{ y } P(X_i = 1) = \frac{1}{2}.$$

Entonces S_n es una binomial de parámetros n y $p = \frac{1}{2}$, luego $\mathbb{E}(S_n) = np = \frac{n}{2}$. Luego la esperanza no es constante y el modelo de la serie no es estacionario.

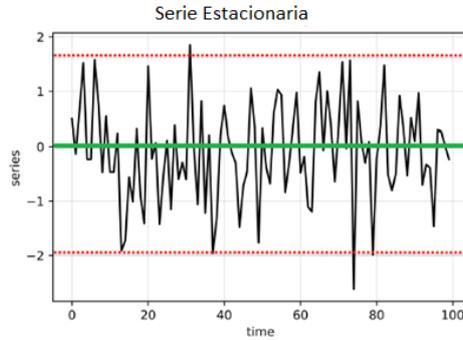


Figura 2.1: Ejemplo de serie temporal con modelo estacionario de ruido blanco

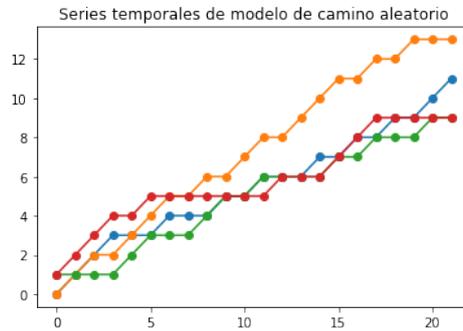


Figura 2.2: Ejemplo de series temporales con modelo no estacionario de camino aleatorio

En la siguiente definición y proposición, se presentan dos funciones asociadas al modelo de una serie temporal de interés que permiten determinar si dicho modelo es o no estacionario.

Definición 2.9. Sea $\{X_i : i \in T\}$ con $T \subseteq \mathbb{Z}$ un modelo de serie temporal con $\mathbb{E}(X_i^2)$ finita para cada $i \in T$.

1. Se define la **función de medias** $\mu : T \rightarrow \mathbb{R}$ como

$$\mu(t) = \mathbb{E}(X_t), \quad \forall t \in T$$

2. Se define la **función de covarianzas** como $\gamma : T \times T \rightarrow \mathbb{R}$ como

$$\gamma(t, s) = \text{Cov}(X_t, X_s), \quad \forall t, s \in T$$

Al índice $t_0 \in T$ elegido se le denomina **momento inicial**.

Proposición 2.10. *Un modelo $\{X_i : i \in T\}$ con función de medias μ y función de covarianzas γ es estacionario sí y solo sí se verifica que:*

1. $\mu(t)$ es constante.
2. $\gamma(t, t+h)$ no depende del valor de t para cada $h > 0$ tal que $t+h \in T$

Demostración. Basta observar que por definición para cada $t, s \in T$:

- $\mathbb{E}(X_t) = \mu(t)$
- $\text{Var}(X_t) = \gamma(t, t)$
- $\text{Cov}(X_t, X_s) = \gamma(t, s)$

Entonces $\mathbb{E}(X_t) = \nu$ ($\nu \in \mathbb{R}$) constante para todo $t \in T$ sí y solo sí μ es constantemente ν .

Por otro lado, si el modelo verifica las propiedades de varianza constante y autocovarianza entonces existen $\sigma \in \mathbb{R}$ y un subconjunto real $\{a_t : t \in T\}$ tal que $\gamma(t, t+h) = a_h$ y $\gamma(t, t) = \sigma$. Recíprocamente, si γ verifica la propiedad segunda entonces basta tomar $\sigma = \gamma(t_0, t_0)$ para cualquier $t_0 \in T$ y $a_h = \gamma(t_0, t_0+h)$ para que se verifiquen las condiciones de varianza constante y autocovarianza. Luego, la propiedad segunda se verifica sí y solo sí se verifican las propiedades de varianza constante y autocovarianza. □

En particular, a los modelos estacionarios se les puede caracterizar por las siguientes aplicaciones, que resultan de una simplificación de las aplicaciones anteriores.

Definición 2.11. *Sea $\{X_i : i \in T\}$ un modelo estacionario, sea $t_0 \in T$*

1. *Se define la **función de autocovarianzas** $\gamma : \{h \in \mathbb{R} \text{ con } t_0 + h \in T\} \rightarrow \mathbb{R}$ como*

$$\gamma(h) = \text{Cov}(X_{t_0}, X_{t_0+h})$$

2. *Se define la **función de autocorrelación** $\rho : \{h \in \mathbb{R} \text{ con } t_0 + h \in T\} \rightarrow \mathbb{R}$ como*

$$\rho(h) = \frac{\gamma(h)}{\gamma(t_0)}$$

Para el caso de estudio interesan las series temporales de tiempo discreto, esto es, $T \subseteq \mathbb{Z}$. Si dichas series comienzan en el momento $t_0 \in T$, mediante una traslación se puede considerar sin pérdida de generalidad que el momento inicial es el $0 \in T$. Con esta observación se enuncia el siguiente resultado.

Proposición 2.12. Propiedades de las funciones de autocovarianzas y autocorrelación. Sea $\{X_i : i \in T \subseteq \mathbb{Z}\}$ un modelo estacionario con función de autocovarianza γ y función de autocorrelación ρ con momento inicial $0 \in T$, sea entonces se verifican las siguientes propiedades:

1. $\gamma(0) \geq 0$
2. $|\gamma(h)| \leq \gamma(0)$ para cada h tal que $h \in T$
3. $\gamma(h) = \gamma(-h)$ para cada h tal que $h, -h \in T$
4. $|\rho(h)| \leq 1$ para cada h tal que $h \in T$
5. $\rho(h) = \rho(-h)$ para cada h tal que $h, -h \in T$

Demostración.

1. Por definición, $\gamma(0) = \text{Var}(0) \geq 0$.
2. Por definición, $\gamma(h) = \text{Cov}(X_0, X_h)$, entonces por la desigualdad de Cauchy-Schwarz y por ser constantes las varianzas se verifica que

$$\begin{aligned} \text{Cov}(X_0, X_h) &= \mathbb{E}[(X_0 - \mathbb{E}X_0)(X_h - \mathbb{E}X_h)] \\ &\leq (\mathbb{E}(X_0 - \mathbb{E}X_0)^2)^{1/2} \cdot (\mathbb{E}(X_h - \mathbb{E}X_h)^2)^{1/2} \\ &= \sqrt{\text{Var}(X_0)\text{Var}(X_h)} = \sqrt{2\text{Var}(X_0)^2} = \gamma(0) \end{aligned}$$

3. Basta aplicar que la covarianza es un operador simétrico y que por ser estacionaria se tiene la propiedad de la autocovarianza. Así, sumando h se tiene:

$$\gamma(-h) = \text{Cov}(X_0, X_{-h}) = \text{Cov}(X_h, X_0) = \text{Cov}(X_0, X_h) = \gamma(h)$$

4. Es resultado inmediato de aplicar la propiedad segunda.

$$|\rho(h)| = \frac{|\gamma(h)|}{\gamma(0)} \leq \frac{|\gamma(0)|}{\gamma(0)} = 1$$

5. Es resultado inmediato de aplicar la propiedad tercera

$$\rho(-h) = \frac{\gamma(-h)}{\gamma(0)} \leq \frac{\gamma(h)}{\gamma(0)} = \rho(h)$$

□

Nota. La suposición de que el momento inicial es el índice 0 se realiza con objeto de simplificar la notación. Si en vez de tomar $t_0 = 0$ para la construcción de las funciones se tomase un t_0 arbitrario entonces la condición que debe verificar h es la pertenencia al conjunto $\{h \in \mathbb{R} \text{ tales que } t_0 + h \in T\}$.

La principal ventaja de una serie estacionaria es que resulta bastante más sencillo realizar predicciones. Esto se debe a que, en este tipo de series, se puede suponer que el comportamiento pasado (una media y una varianza determinadas) se mantiene en el futuro. Es por ello que la mayoría de los modelos que intentan predecir el comportamiento sobre series temporales se basan en series temporales estacionarias. De estos modelos destaca el **modelo ARMA**.

Por último, dado que de una serie temporal no se conoce previamente su modelo, se deben estimar las funciones que la caracterizan mediante una muestra obtenida del modelo o del fenómeno observado. En particular, si suponemos que el modelo es estacionario se puede caracterizar como muestra la siguiente definición.

Definición 2.13. Sea $\mathbf{X} = [x_1, \dots, x_n]$ una serie temporal con modelo estacionario:

1. Se define la **media muestral** $\bar{\mu}$ como

$$\bar{\mu} = \sum_{i=1}^n x_i$$

2. Se define la **función de autocovarianzas muestral** $\hat{\gamma} : \{-n, \dots, n\} \rightarrow \mathbb{R}$ como

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{i=1}^{n-|h|} (x_{i+|h|} - \bar{\mu})(x_i - \bar{\mu}), \quad -n \leq h \leq n$$

3. Se define la **función de autocorrelación muestral** $\hat{\rho} : \{-n, \dots, n\} \rightarrow \mathbb{R}$ como

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}, \quad -n \leq h \leq n$$

2.1.3. Componentes de una serie temporal. El modelo aditivo

El modelo básico y más general, divide una serie temporal en tres componentes (que a su vez son una serie temporal cada una de ellas) y que son la tendencia, la estacionalidad y el residuo. En esta sección, se suponen series temporales de tiempo discreto cuyo conjunto de índices está contenido en \mathbb{N} y tiene cardinal finito.

Definición 2.14. Sea $\mathbf{X} = [x_1, \dots, x_n]$ una serie temporal:

1. Se define la **tendencia** como el comportamiento o movimiento regular (monotonía) de la serie.
2. Se define la **estacionalidad** como los movimientos de oscilación de la serie.

3. Se define el **residuo**, *residual* o *resto* como las variaciones aleatorias restantes que no se pueden englobar en la tendencia o en la estacionalidad.

Notación. Como notación general, $\mathbf{T} = [t_1, \dots, t_n]$, $\mathbf{S} = [s_1, \dots, s_n]$ y $\mathbf{R} = [r_1, \dots, r_n]$ denotan respectivamente la tendencia, estacionalidad y el residuo de la serie temporal $\mathbf{X} = [X_1, \dots, X_n]$.

Entonces, dada una serie temporal la tendencia comprende la evolución global de la serie y la estacionalidad captura los efectos de los eventos que ocurren con cierta repetición en el tiempo, tales como los eventos periódicos. Por último, el residuo recoge todo aquello que no puede ser capturado ni por la tendencia ni por la estacionalidad.

Ejemplo 2.15. El siguiente ejemplo ilustra el concepto de tendencia. Para ello se supone una serie temporal definida de la siguiente forma:

$$x_t = t + 2 \cdot \text{sen}(t), \quad t = 0, \dots, 50$$

Dado que $-2 \leq 2 \cdot \text{sen}(t) \leq 2, \quad \forall t$, entonces se tiene que $x_t \leq x_{t+3}$ luego la tendencia en su conjunto es creciente. La *Figura 2.3* ilustra el ejemplo.

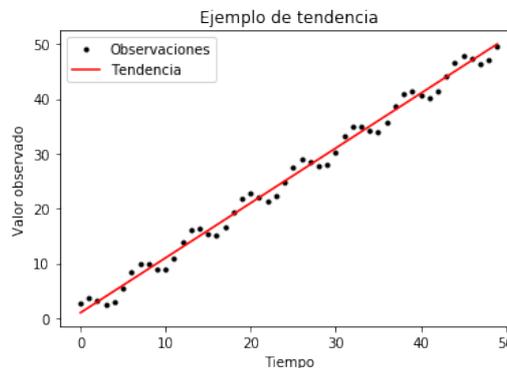


Figura 2.3: Ejemplo de la tendencia de una serie temporal

Dependiendo de la duración de las series temporales, generalmente cuando estas comprenden varios años, la estacionalidad se divide en variaciones estacionales y variaciones cíclicas. Las **variaciones estacionales** son oscilaciones a corto plazo mientras que las **variaciones cíclicas** son los movimientos a largo plazo.

Ejemplo 2.16. Como ejemplo, supongamos que se realizan mediciones de la temperatura en una estación meteorológica concreta cada hora del día durante diez años. Si se analiza cada año se observa una periodicidad de la temperatura media diaria que aumenta en los meses de primavera, alcanza su máximo en verano y disminuye en los meses de otoño hasta alcanzar su

mínimo en invierno. Este comportamiento que se verifica todos los años se correspondería con las variaciones cíclicas. Finalmente, si se observa un día concreto, se tiene que la temperatura en cada hora aumenta desde el amanecer a la puesta de sol y disminuye desde el anochecer hasta que amanece. Estas oscilaciones que se repiten de forma diaria se incluirían por las variaciones estacionales.

En el caso de estudio, este tipo de variaciones a largo plazo carece de sentido y por tanto se omitirán en lo que resta, considerando la estacionalidad como las variaciones estacionales.

El modelo básico de descomposición sobre series temporales es el **modelo aditivo**. Este modelo supone que la serie temporal es la suma de sus componentes.

Definición 2.17. sea $\mathbf{X} = [x_1, \dots, x_n]$ una serie temporal y sean $\mathbf{T} = [t_1, \dots, t_n]$, $\mathbf{S} = [s_1, \dots, s_n]$ y $\mathbf{R} = [r_1, \dots, r_n]$ la tendencia, la estacionalidad y el residuo respectivamente, se define el **modelo aditivo** como aquel que verifica las siguientes hipótesis:

1. La serie original es la suma de sus componentes. Esto es:

$$x_k = t_k + s_k + r_k, \quad \forall k = 1, \dots, n \quad (2.1)$$

2. Los modelos de las componentes de la serie temporal $\mathbf{T}, \mathbf{S}, \mathbf{R}$ son independientes entre sí para cada índice fijo, y el modelo de la serie \mathbf{R} es débilmente estacionario.

Ejemplo 2.18. Como ejemplo se presenta una serie temporal que cumple el modelo aditivo. Con las notaciones anteriores:

$$x_k = 2,75k + 10 \cdot \text{sen}(k) + 10 \quad k = 1, 2, \dots, 50$$

La serie temporal resultante se representa en la *Figura 2.4*. Claramente, por las definiciones dada en la *Definición 2.14* se tiene que la tendencia viene dada por $t_k = 2,75k + 10$ y la estacionalidad viene dada por $s_k = 10 \cdot \text{sen}(k)$. Con la construcción definida se cumple que no hay residual, puesto que no hay valores aleatorios que no sean parte ni de la tendencia ni de la estacionalidad, luego $r_k = 0$. En la *Figura 2.4* se puede observar la descomposición.

Existe un segundo modelo que supone que la serie es el producto de sus componentes. Este modelo se denomina **modelo multiplicativo**, y con las notaciones anteriores se expresa como:

$$x_k = t_k \cdot s_k \cdot r_k, \quad \forall k = 1, \dots, n \quad (2.2)$$

El modelo multiplicativo se puede transformar en un modelo aditivo, puesto que, es equivalente a aplicar las hipótesis de dicho modelo a la transformación por el logaritmo de la serie original. Sea $\mathbf{X} = [x_1, \dots, x_n]$ una serie temporal que verifica una descomposición multiplicativa (2.2) entonces:

$$\log(x_k) = \log(t_k) + \log(s_k) + \log(r_k), \quad \forall k = 1, \dots, n$$

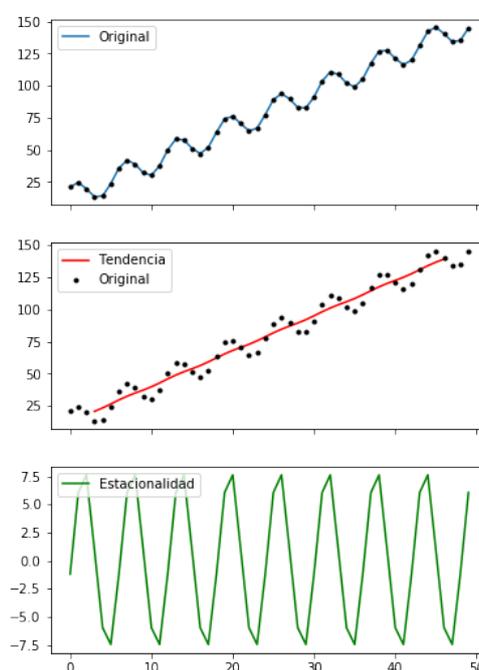


Figura 2.4: Serie temporal definida como $x_k = 2,75k + 10 \cdot \text{sen}(k) + 10 \quad k = 1, 2, \dots, 51$

2.1.4. Técnicas para determinar la tendencia. El algoritmo de medias móviles

Bajo la hipótesis del modelo aditivo, la tendencia es una componente clave que permite analizar el comportamiento de la serie temporal obviando los efectos estacionarios y las interferencias anómalas. Para la estimación se diferencian dos grupos de series temporales: series con tendencia determinista y series con tendencia evolutiva. Esta sección está basada en la Sección 9.2 de [3] y en [4].

Definición 2.19.

1. Se dice que una serie temporal tiene **tendencia determinista** si la tendencia está determinada por una función simple, continua y monótona en todo el conjunto de índices.
2. En caso contrario, se dice que la serie temporal tiene **tendencia evolutiva**.

Una tendencia evolutiva se caracteriza por variar su monotonía en el tiempo, y se puede aproximar localmente (en un intervalo pequeño de tiempo) por funciones simples y monótonas

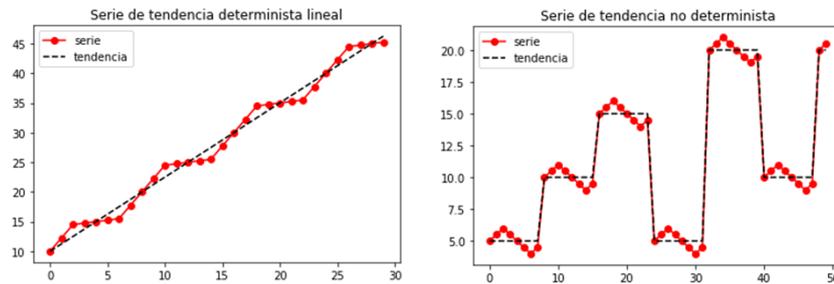


Figura 2.5: Ejemplo de serie temporal con tendencia determinista (izquierda) y con tendencia evolutiva (derecha)

1. Estimación de la tendencia determinista.

En las series temporales con tendencia determinista la principal técnica que se aplica en estos casos es el **análisis de regresión**. Este método supone que los parámetros que definen la tendencia permanecen inmutables en el tiempo, y busca ajustar la “nube de puntos” por una función continua y monótona. Si $\mathbf{T} = [t_1, \dots, t_n]$ denota la tendencia, las técnicas de regresión más comunes son:

- La **regresión lineal** supone que la dependencia de tendencia respecto al tiempo viene determinada por un polinomio lineal.

$$t_k = a + bk \quad a, b \in \mathbb{R}$$

- La **regresión polinómica** supone que la dependencia de la tendencia respecto al tiempo queda determinada por un polinomio de grado m . En especial, destaca el uso de la regresión cuadrática ($m = 2$) y cúbica ($m = 3$).

$$t_k = a_0 + \sum_{i=1}^m a_i k^i \quad a_0, \dots, a_m \in \mathbb{R}$$

- La **regresión logarítmica** supone que la tendencia tiene un comportamiento potencial o logarítmico.

$$t_k = a + b \cdot \ln(k) \quad a, b \in \mathbb{R}$$

- La **regresión exponencial** supone que la tendencia tiene un crecimiento exponencial respecto al tiempo.

$$t_k = a + b \cdot e^k \quad a, b \in \mathbb{R}$$

2. Estimación de la tendencia evolutiva.

En las series temporales con tendencia evolutiva la principal técnica que se aplica es el **algoritmo de medias móviles** que se basa en la suposición de que la observación en un momento determinado se ve “afectada” por las observaciones cercanas en el tiempo a dicho momento.

Sea $\mathbf{X} = [x_1, \dots, x_n]$ una serie temporal, el algoritmo de medias móviles es un algoritmo de suavizado que genera una nueva serie temporal $\mathbf{Y} = [y_1, \dots, y_m]$. Existen dos versiones dependiendo de si se consideran las observaciones futuras al momento actual (*versión centrada*) o solo las observaciones pasadas (*versión temporal*). En ambos casos se determina previamente un valor $v \in \mathbb{N}$ (denominado (“tamaño de ventana”) que marca el número de observaciones cercanas a contemplar para cada momento concreto.

Definición 2.20. Con las notaciones anteriores, \mathbf{Y} es la media móvil de la serie \mathbf{X} y se denomina **orden** de una media móvil \mathbf{Y} al valor v que representa el tamaño de ventana.

Algoritmo de Medias Móviles:

Sea $\mathbf{X} = [x_1, \dots, x_n]$ una serie temporal, se denota por $\mathbf{Y} = [y_1, \dots, y_m]$ la serie temporal resultante.

1. Se toma un valor de ventana $v \in \mathbb{N}$. Dicho valor debe ser impar si se considera la versión centrada del algoritmo.
2. Dependiendo de la versión del algoritmo elegida:
 - a) *Versión centrada.*

Sea por $u = \frac{v-1}{2}$, entonces para cada k natural en $[u, n-u]$ se toma y_k como la media de las observaciones en el periodo $[k-u, k+u]$:

$$y_k = \frac{1}{2u+1} \sum_{i=k-u}^{k+u} \omega_i \cdot x_i, \quad \text{donde} \quad \frac{1}{2u+1} \sum_{i=k-u}^{k+u} \omega_i = 1 \quad (2.3)$$

- b) *Versión temporal.*

Para cada k natural en $[v, n]$ se toma y_k como la media de las observaciones en el periodo $[k-v, k]$:

$$y_k = \frac{1}{v+1} \sum_{i=k-v}^k \omega_i \cdot x_i, \quad \text{donde} \quad \frac{1}{v+1} \sum_{i=k-v}^k \omega_i = 1 \quad (2.4)$$

En ambas versiones, la longitud de la media móvil \mathbf{Y} es exactamente $n - v$ siendo n la longitud de \mathbf{X} y v el orden.

El orden del algoritmo es clave en el resultado final. Así, cuando se toma un orden lo suficientemente grande se suavizan los efectos de la estacionalidad y se eliminan los efectos de los residuos (es decir, se mantiene la tendencia). Mientras que, si se toma un orden pequeño, entonces se suprimen los residuales manteniendo los efectos de la tendencia y la estacionalidad. Generalmente, los órdenes que no engloban una oscilación de la estacionalidad son considerados órdenes pequeños, mientras que un orden grande es considerado aquel que engloba al menos una oscilación de la estacionalidad.

La versión más sencilla, es el **algoritmo de medias móviles simples** en el cual el valor de los pesos $\omega_i = 1, \forall i = 1, \dots, n$.

Ejemplo 2.21. El siguiente ejemplo es una aplicación del algoritmo de medias móviles simple de orden 7 (denotada por \mathbf{Y}) y de orden 3 (denotada por \mathbf{Z}) sobre la serie temporal presentada en el *Ejemplo 2.18*. Esta serie venía determinada como

$$a_k = 2,75k + 10\text{sen}(k) + 10 \quad k = 1, 2, \dots, 50$$

cuya tendencia es $t_k = 2,75k + 10$. Introducimos una serie de valores residuales a la serie. En particular definimos el siguiente residual $\mathbf{R} = [r_1, \dots, r_{51}]$:

$$r_{10} = -10, \quad r_{20} = 10, \quad r_{30} = 20, \quad r_{40} = -10, \quad r_k = 0 \text{ si } k \notin \{10, 20, 30, 40\}$$

Y se construye la serie temporal

$$x_k = 2,75k + 10\text{sen}(k) + 10 + r_k \quad k = 1, 2, \dots, 50$$

Entonces, según (2.3) se tiene en la *Tabla 2.1* con las medias móviles correspondientes.

k	4	5	6	10	15	20	25	30	35	40	45
A	13,4	14,2	23,7	32,1	57,8	74,1	77,4	82,6	102	127,5	142,3
X	13,4	14,2	23,7	22,1	57,8	84,1	77,4	102,6	102	117,5	142,3
T	21	23,8	26,5	37,5	51,2	65	78,8	92,5	106,2	120	133,8
Y	21,8	24,8	26,8	36,6	50,6	65,5	78,9	96,4	106,7	117,8	132,9
Z	15,8	17,1	24,6	30,4	55,8	74,7	77,8	92,5	103,3	121,8	139,7

Tabla 2.1: Comparativa de la serie \mathbf{X} y de la tendencia \mathbf{T} real con las series calculada por medias móviles de orden 7, \mathbf{Y} , y de orden 3, \mathbf{Z} .

Como la función $\text{sen}(x)$ $x \in \mathbb{R}$ es 2π -periódica, entonces para un tamaño de ventana superior a 2π el algoritmo de medias móviles suprime tanto los efectos residuales como los estacionales. Por ello, la serie \mathbf{Y} de orden $7 > 2\pi$ aproxima la tendencia de la serie original, mientras que la serie \mathbf{Z} de orden $3 < 2\pi$ aproxima la serie omitiendo los residuales (serie \mathbf{A}). Véase la *Figura 2.6* donde se representan las diferentes series temporales consideradas.

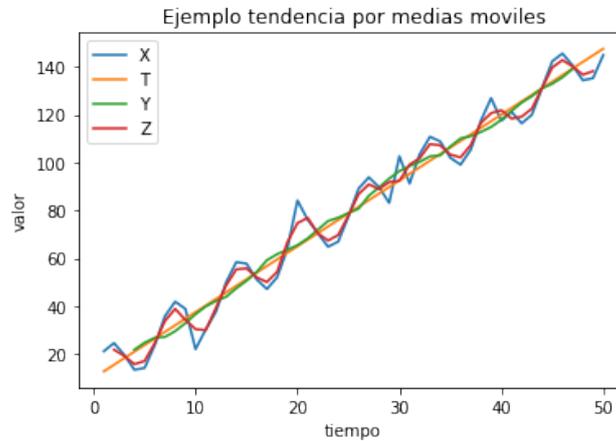


Figura 2.6: Comparativa de la serie \mathbf{X} y de la tendencia \mathbf{T} real con las series calculada por medias móviles de orden 7, \mathbf{Y} , y de orden 3, \mathbf{Z} .

2.2. Métrica sobre series temporales. La distancia DTW

En esta sección se pretende definir una distancia que permita comparar lo similares que son dos series temporales cualesquiera. Para definir una distancia adecuada en el caso de estudio se presentarán diferentes distancias que tienen en cuenta las peculiaridades que caracteriza al conjunto de las series temporales sobre otros tipos de datos. En particular, se introducirá el concepto de *deformación en el tiempo* como técnica comparativa de series temporales. Consúltese la *Sección 2.3* para ciertos detalles sobre la *DTW* y su adjetivo distancia.

Esta sección está estructurada según [7], siendo esta referencia bibliográfica introductoria completada mediante [5], [6], [8], [9]. Los resultados han sido adaptados a su aplicación sobre series temporales reales y las demostraciones han sido realizadas siguiendo las indicaciones dadas en las referencias anteriores u obtenidas adaptando ciertas pruebas de casos más genéricos. En particular, en [6] y [9] se generaliza a cadenas sobre un alfabeto cualquiera, pudiendo ser series de tipo categóricas. Por último, los algoritmos presentados se pueden encontrar en [7], [5] y [9], habiéndose homogeneizado la sintaxis y el pseudocódigo en la presente memoria.

2.2.1. Introducción. El concepto de deformación en el tiempo

En esta sección se introducen la problemática de la comparación de dos series temporales arbitrarias y sus peculiaridades. El enfoque dado en esta sección y la siguiente se basa en el apéndice de [12].

Definición 2.22. Sean $\mathbf{X} = (x_1, \dots, x_n)$ e $\mathbf{Y} = (y_1, \dots, y_n)$ dos vectores de \mathbb{R}^n , se define la *distancia euclídea* como:

$$d_E(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.5)$$

Con la distancia euclídea simplemente se consideran las series temporales como vectores. Es decir, sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_n]$ dos series temporales de la misma longitud entonces

$$d_E(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.6)$$

La distancia euclídea, aunque sencilla, resulta bastante incompleta pues no permite comparar series temporales de distinta longitud, ni es capaz de detectar el efecto *time warping* o **deformación en el tiempo**. El efecto de deformación en el tiempo se produce cuando se realiza una misma acción, pero a diferentes velocidades o en diferentes momentos de ocurrencia. El siguiente ejemplo lo ilustra.

Ejemplo 2.23. Supongamos dos personas que realizan el mismo recorrido y que cada minuto anotan la altitud a la que se encuentran. Además, en la parte más alta una de ellas se desplaza el doble de rápido que la otra persona. Las series temporales podrían ser de la forma:

$$\begin{aligned} \mathbf{X} &= [1, 2, 3, 3, 2, 1] && \text{(persona rápida)} \\ \mathbf{Y} &= [1, 2, 3, 3, 3, 3, 2, 1] && \text{(persona lenta)} \end{aligned}$$

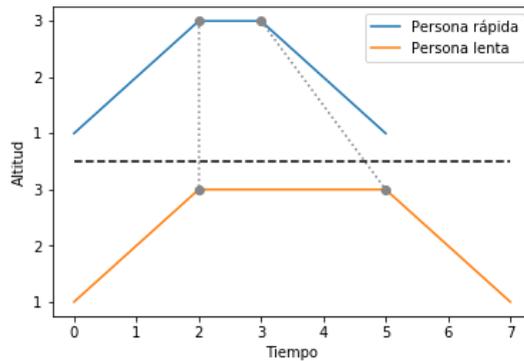


Figura 2.7: Ejemplo del efecto de deformación en el tiempo

El efecto de deformación en el tiempo se observa en la parte más alta (véase *Figura 2.7*), pues ambas personas realizan el mismo recorrido, pero como una de ellas es más rápida en ese tramo la altitud de 3 unidades aparece menos veces recogida.

La distancia euclídea es incapaz de comparar ambas series temporales. Se requiere que la medida de comparación sobre series temporales usada sea capaz de detectar ambas series como equivalentes. En la *Figura 2.7* se puede observar (en gris con línea punteada) como se relacionan los puntos claves de ambos recorridos, aunque no coincidan en el tiempo, pero se corresponden al mismo punto de longitud recorrida. En el ejemplo, el objetivo perseguido se entiende como “la persona rápida espera a la persona lenta antes de iniciar el descenso”.

Definición 2.24. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales, si se compara \mathbf{X} con respecto a \mathbf{Y} entonces a \mathbf{X} se le denomina serie **muestra** y a \mathbf{Y} se le denomina serie **prueba**.

2.2.2. Caminos de deformación su coste

En esta sección se va a presentar una medida de comparación que permita trabajar con series temporales de distinta longitud y sea capaz de detectar la deformación en el tiempo. Para ello se va a estudiar las posibles asociaciones entre los índices de dos series temporales buscando la asociación óptima en sentido que minimice la distancia de un elemento de la serie temporal muestra respecto a serie temporal prueba.

Definición 2.25. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales con conjuntos de índices $\{1, \dots, n\}$ y $\{1, \dots, m\}$ se define un **camino de deformación** (*Warping Path*) de orden $n \times m$ y cardinal P como una secuencia de P puntos bidimensionales

$$\{(i_1, j_1), \dots, (i_P, j_P)\} \subset \{1 \dots n\} \times \{1 \dots m\}$$

tales que satisfacen

1. *Condiciones en los bordes:* $(i_1, j_1) = (1, 1)$ y $(i_P, j_P) = (n, m)$

2. *Condiciones de monotonía:*

$$i_1 \leq i_2 \leq \dots \leq i_P$$

$$j_1 \leq j_2 \leq \dots \leq j_P$$

3. *Condición sobre el tamaño de paso:*

$$(i_s, j_s) - (i_{s-1}, j_{s-1}) \in \{(0, 1), (1, 0), (1, 1)\}, \quad \forall s = 2 \dots P$$

Las condiciones de monotonía y de tamaño de paso suponen que, con las notaciones de la definición, para cada $s = 2 \dots P$ se verifica

$$i_s = \begin{cases} i_{s-1} \\ i_{s-1} + 1 \end{cases}$$

$$j_s = \begin{cases} j_{s-1} \\ j_{s-1} + 1 \end{cases}$$

Notación. Un camino de deformación

$$\{(i_1, j_1), \dots, (i_P, j_P)\} \subset \{1 \dots n\} \times \{1 \dots m\}$$

se denota como una aplicación $\Psi : \{1 \dots P\} \longrightarrow \{1 \dots n\} \times \{1 \dots m\}$ tales que para cada $s \in \{1 \dots P\}$ se verifica que $(i_s, j_s) = \Psi(s)$. Además, con un abuso de notación, se denota al conjunto igual que a la aplicación definida, de forma que la situación $(i, j) = \Psi(s)$ para algún s se representa como $(i, j) \in \Psi$. Además, a cada componente de Ψ se les denota respectivamente como ψ_X y ψ_Y , de forma que $\Psi = (\psi_X, \psi_Y)$.

Un camino de deformación establece una asociación entre los índices de dos series temporales. La asociación básica, que es la que utiliza la distancia euclídea, es el momento de ocurrencia, pero con un camino de deformación se pueden establecer asociaciones de índices con momento de ocurrencia distinto. Esto es lo que le da el adjetivo “deformación”.

Así pues, se puede considerar una nueva línea temporal (conjunto de índices) $\{1 \dots P\}$ y dos series temporales de igual longitud $\mathbf{X}' = [x'_1, \dots, x'_p]$ e $\mathbf{Y}' = [y'_1, \dots, y'_p]$ tales que

$$\text{para cada } s = 1 \dots P \quad x'_s = x_{\psi_X(s)} \text{ e } y'_s = y_{\psi_Y(s)}$$

Y considerar la asociación dada por el momento de ocurrencia. Nótese que \mathbf{X}' e \mathbf{Y}' son el resultado de “deformar” las series originales.

Las asociaciones de índices dadas por un camino de deformación satisfacen por definición una restricción temporal lógica, que se enuncia como *el tiempo no puede retroceder, solo detenerse* y es lo que indica la condición segunda de la *Definición 2.25*. Además, las condiciones segunda y tercera garantizan que Ψ , como aplicación, es inyectivo, lo que se enuncia como *el tiempo no puede detenerse en ambas series*. Finalmente, las condiciones en los bordes permiten garantizar el siguiente resultado.

Proposición 2.26. *Sea Ψ un camino por deformación de orden $n \times m$ y cardinal P entonces:*

1. *Para cada $i \in \{1, \dots, n\}$ existen $j \in \{1, \dots, m\}$ y $s \in \{1, \dots, P\}$ tales que $\Psi(s) = (i, j)$.*
2. *Para cada $j \in \{1, \dots, m\}$ existen $i \in \{1, \dots, n\}$ y $s \in \{1, \dots, P\}$ tales que $\Psi(s) = (i, j)$.*

Demostración. Se realiza el razonamiento para la primera afirmación, siendo análogo el razonamiento para la segunda. Sea $\Psi = (\psi_X, \psi_Y)$ el camino de deformación. En primer lugar, por las condiciones primera y segunda de la *Definición 2.25* los índices $1, n \in \{1, \dots, n\}$ y $1, m \in \{1, \dots, m\}$ verifican la proposición anterior.

- **Existencia de s .** Se razona por reducción al absurdo. Se supone que existe $i \in \{1 \dots n\}$ con $i \neq n$ e $i \neq 1$ tal que $\psi_X(s) \neq i$ para todo $s = 1 \dots P$. Entonces por la condición tercera, $\psi_X(s) < i < n$ para todo $s = 1, \dots, P$. Luego $\psi(P) < n$ en contra de la segunda condición de la definición.
- **Existencia de j .** Por lo anterior, para $i \in \{2, \dots, (n-1)\}$ existe $s \in \{1, \dots, P\}$ tal que $\psi_X(s) = i$, luego basta tomar $j = \psi_Y(s)$.

□

Notación. Se denota al conjunto de los caminos de deformación orden $n \times m$ de cardinal no fijo como $\mathcal{P}_{n,m}$.

Si P es el cardinal de un camino de $\mathcal{P}_{n,m}$ entonces se verifica que

$$\text{máx}\{n, m\} \leq P \leq n + m$$

Definición 2.27. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales con conjuntos de índices $\{1, \dots, n\}$ y $\{1, \dots, m\}$ respectivamente, sea \mathbb{V} un espacio métrico que contiene a todos los coeficientes de las series temporales dotado de una distancia d , y sea Ψ un camino por deformación de orden $n \times m$ y cardinal P , se define el coste del camino para las series \mathbf{X} e \mathbf{Y} como

$$C_{\mathbf{X}, \mathbf{Y}}(\Psi) = \sum_{(i,j) \in \Psi} d(x_i, y_j) = \sum_{i=1}^P d(x_{\Psi_X(i)}, y_{\Psi_Y(j)})$$

2.2.3. La distancia DTW

En esta sección se define la distancia DTW (*Dynamic Time Warping*) de dos series temporales y se estudian sus principales propiedades, así como el camino de deformación óptimo en el sentido de coste mínimo (véanse [9] y [12]).

Se supone que \mathbb{V} es un espacio métrico (generalmente \mathbb{R}) con una distancia denotada por d y que contiene a todos los coeficientes de las series temporales que se presentan. Estas series son de tiempo discreto con su conjunto de índices contenido en los naturales y de cardinal finito.

Definición 2.28. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales con \mathbf{X} la serie muestra e \mathbf{Y} la serie prueba:

1. Se define la **distancia DTW** (Dynamic Time Warping) de las series temporales \mathbf{X} frente a \mathbf{Y} como

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = \min_{\Psi \in \mathcal{P}_{n,m}} \left(\sum_{(i,j) \in \Psi} d(x_i, y_j) \right) = \min_{\Psi \in \mathcal{P}_{n,m}} C_{\mathbf{X}, \mathbf{Y}}(\Psi) \quad (2.7)$$

2. Se define la **curva de deformación** Φ como el camino de deformación que hace mínimo la expresión (2.7). Es decir,

$$\Phi = \arg \min_{\Psi \in \mathcal{P}_{n,m}} \left(\sum_{(i,j) \in \Psi} d(x_i, y_j) \right) = \arg \min_{\Psi \in \mathcal{P}_{n,m}} C_{\mathbf{X}, \mathbf{Y}}(\Psi) \quad (2.8)$$

El concepto de curva de deformación permite formalizar la definición de deformación en el tiempo.

Definición 2.29. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales y sea $\Phi = (\phi_X, \phi_Y)$ la curva de deformación, se dice que se ha producido una **deformación en el tiempo** en la comparación de \mathbf{X} frente a \mathbf{Y} en el momento $k+1$ si se da alguno de los siguientes casos:

- (a) $\phi_X(k+1) = \phi_X(k)$ y $\phi_Y(k+1) = \phi_Y(k) + 1$
- (b) $\phi_X(k+1) = \phi_X(k) + 1$ y $\phi_Y(k+1) = \phi_Y(k)$

En el caso (a) se dice que se ha producido una **inserción** y en el caso (b) se dice que se ha producido una **eliminación**.

Además, si se verifica que

- (c) $\phi_X(k+1) = \phi_X(k) + 1$ y $\phi_Y(k+1) = \phi_Y(k) + 1$

entonces se dice que se ha producido una **coincidencia temporal** en $k+1$.

Ejemplo 2.30. Consideramos las series temporales $\mathbf{X} = [3, 3, 6, 0, 3, 3]$ e $\mathbf{Y} = [3, 3, 3, 6, 0, 3]$. Su distancia euclídea es de $\sqrt{54}$, y su distancia como suma de las distancias siguiendo el orden cronológico es de 12 (fig. 2.8 A). Sin embargo, si consideramos la asociación de puntos presente en la fig. 2.8 B) dada por el camino de deformación

$$\Psi = \{(1, 1), (2, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 6)\}$$

El coste de dicho camino para la comparación de las series temporales es cero.

El siguiente resultado muestra que la distancia DTW es capaz de detectar y suprimir los efectos de deformaciones en el tiempo. En particular, basta probarlo para dos series temporales tales que la serie prueba tenga duplicado un único término respecto de la serie muestra.

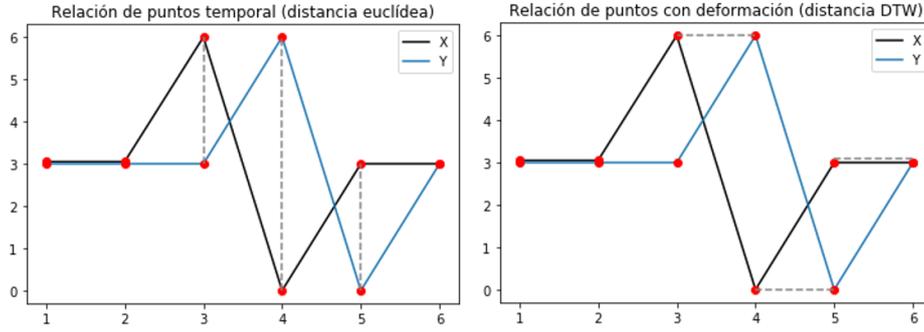


Figura 2.8: Comparación de la asociación cronológica (A) con una asociación con deformación en el tiempo (B). Las líneas grises punteadas indican las asociaciones.

Proposición 2.31. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_{n+1}]$ dos series temporales tales que $\mathbf{Y} = [x_1, \dots, x_{p-1}, x_p, x_p, x_{p+1}, \dots, x_n]$ entonces

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = 0$$

Demostración. Basta definir adecuadamente un camino de deformación. Como la distancia d_{DTW} es suma de términos no nulos (pues d es una distancia) entonces basta hallar un camino de deformación de coste cero para concluir que la distancia DTW de las series es cero. Dicho camino por deformación es:

$$\Gamma = \{(1, 1), \dots, (p-1, p-1), (p, p), (p, p+1), (p+1, p+2), (p+2, p+3), \dots, (n, n+1)\}$$

Trivialmente, Γ es un camino por deformación (pues cumple todas las propiedades) y su coste es

$$\begin{aligned} \sum_{(i,j) \in \Gamma} d(x_i, y_j) &= \sum_{i=1}^p d(x_i, y_i) + d(x_p, y_{p+1}) + \sum_{i=p+1}^n d(x_i, y_{i+1}) \\ &= \sum_{i=1}^p d(x_i, x_i) + d(x_p, x_p) + \sum_{i=p+1}^n d(x_i, x_i) = 0 \end{aligned}$$

□

Cuando una serie temporal \mathbf{Y} es el resultado de duplicar algunos términos respetando el orden cronológico de otra serie \mathbf{X} , se dice que \mathbf{Y} es una extensión de \mathbf{X} . En la *Sección 2.3* se profundiza en este concepto y se muestra que efectivamente la distancia DTW toma valor nulo frente cualquier deformación temporal.

Mediante la curva de deformación, es posible obtener a partir de dos series temporales \mathbf{X} e \mathbf{Y} otras dos series de la misma longitud \mathbf{X}' e \mathbf{Y}' que conservan la distancia DTW . El siguiente resultado ilustra dicho comportamiento.

Proposición 2.32. Sean \mathbf{X} e \mathbf{Y} dos series temporales entonces existen $\mathbf{X}' = [x'_1, \dots, x'_k]$ e $\mathbf{Y}' = [y'_1, \dots, y'_k]$ series temporales de la misma longitud tales que

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = d_{DTW}(\mathbf{X}', \mathbf{Y}') = \sum_{i=1}^k d(x'_i, y'_i)$$

Además, si $\Phi = (\phi_X, \phi_Y)$ denota la curva de deformación con cardinal P entonces se verifica que para cada $i \in \{1, \dots, k\}$

$$x'_i = x_{\phi_X(i)} \quad e \quad y'_i = y_{\phi_Y(i)}$$

Demostración. Por la construcción dada, la curva de deformación de \mathbf{X}' e \mathbf{Y}' se corresponderá con

$$\Psi = \{(i, i) : i = 1, \dots, P\} = \{(\phi_X(i), \phi_Y(i)) : i = 1, \dots, P\}$$

pues en \mathbf{X}' e \mathbf{Y}' aparecen los mismos coeficientes de \mathbf{X} e \mathbf{Y} a lo sumo repetidos respetando el orden cronológico de aparición. Por tanto:

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^P d(x_{\phi_X(k)}, y_{\phi_Y(k)}) = \sum_{i=1}^P d(x'_i, y'_i) = d_{DTW}(\mathbf{X}', \mathbf{Y}')$$

□

Corolario 2.33. Con las notaciones de la proposición anterior (Proposición 2.32) si $d_{DTW}(\mathbf{X}, \mathbf{Y}) = 0$ entonces $\mathbf{X}' = \mathbf{Y}'$

Demostración. Se tiene que $0 = \sum_{i=1}^k d(x'_i, y'_i)$ y como d es una distancia entonces $d(x'_i, y'_i) = 0$ y por tanto $x'_i = y'_i$ para cada i .

□

Ejemplo 2.34. Sean $\mathbf{X} = [1, 2, 3]$ e $\mathbf{Y} = [1, 3, 3, 4, 5]$ dos series temporales. La curva de deformación es:

$$\Phi(1) = (1, 1), \quad \Phi(2) = (2, 1), \quad \Phi(3) = (3, 2), \quad \Phi(4) = (3, 3), \quad \Phi(5) = (3, 4)$$

En $k = 2$ se produce una eliminación, mientras que en $k = 3$ se produce una coincidencia y en $k = 4$ y $k = 5$ se produce una inserción. Además, según la *Proposición 2.32*, las extensiones de las series temporales son:

$$\begin{aligned} \mathbf{X}' &= [x_1, x_2, x_3, x_3, x_3] = [1, 2, 3, 3, 3] \\ \mathbf{Y}' &= [y_1, y_1, y_2, y_3, y_4] = [1, 1, 3, 3, 4] \end{aligned}$$

El ejemplo anterior ilustra las deformaciones producidas en las extensiones de las series muestra y prueba. Cuando se produce una eliminación en k entonces se duplica un término en la extensión de la serie prueba (caso $k = 2$ del ejemplo anterior) y supone que el tiempo en la serie muestra se detiene. Por otro lado, cuando se produce una inserción en k se duplica el término correspondiente en la extensión de la serie muestra (caso $k = 4$ en el ejemplo anterior) y supone que el tiempo en la serie prueba no transcurre.

2.2.4. Algoritmo para el cálculo de la distancia DTW. La matriz DTW

El cálculo de la distancia DTW mediante la expresión (2.7) resulta complejo y muy costoso, pues habría que analizar todos los posibles caminos por deformación para cada cardinal P con $\max(n, m) \leq P \leq n + m$. Para ello, se presenta un algoritmo capaz de calcular la distancia DTW de dos series con complejidad cuadrática ([7]).

El siguiente algoritmo aplica las restricciones temporales sobre los caminos de deformación para reducir la complejidad del problema. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ las series muestra y prueba, siguiendo las restricciones temporales el par de índices (i, j) de un camino de deformación puede ir precedido exclusivamente por los pares de índices $(i-1, j)$, $(i, j-1)$ o $(i-1, j-1)$. Si se denota por $\psi_{i,j}$ el camino de coste mínimo aplicado a las series $[x_1, \dots, x_i]$ y $[y_1, \dots, y_j]$ entonces

$$C(\psi_{i,j}) = d(x_i, y_j) + \min\{C(\psi_{i-1,j-1}), C(\psi_{i-1,j}), C(\psi_{i,j-1})\}$$

Esta idea es la que subyace en la construcción del algoritmo.

El algoritmo *Dynamic Time Warping (DTW)* (versión primera)

Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales cualesquiera de longitud n y m respectivamente y sea d una distancia. Se denota por $A_{DTW} = (a_{i,j})_{i=1, \dots, n; j=1, \dots, m}$ la matriz del algoritmo DTW.

1. Se construye la matriz DTW.

1.1. El primer elemento de la matriz DTW es la distancia entre x_1 e y_1 .

$$a_{1,1} = d(x_1, y_1) \quad (2.9)$$

1.2. Se calcula la primera fila y columna de la matriz DTW. Para cada $j = 2 \dots m$

$$a_{1,j} = d(x_1, y_j) + a_{1,j-1} \quad (2.10)$$

$$a_{j,1} = d(x_j, y_1) + a_{j-1,1} \quad (2.11)$$

1.3. Para cada $i = 2, \dots, n$ y para cada $j = 2, \dots, m$ se toma el elemento $a_{i,j}$ como:

$$a_{i,j} = d(x_i, y_j) + \min(a_{i,j-1}, a_{i-1,j}, a_{i-1,j-1}) \quad (2.12)$$

2. La distancia DTW entre ambas series es el término $a_{n,m}$.

La afirmación de que el término (n, m) –ésimo de la matriz DTW es el valor de la distancia DTW está justificada por el *Teorema 2.37*. Para probar dicho teorema se deben introducir previamente unos conceptos sobre la relación de dos índices según su matriz DTW.

Definición 2.35. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales, y sea $A_{DTW} = (a_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$ la matriz de distancias del algoritmo DTW que compara \mathbf{X} frente a \mathbf{Y} .

1. Se dice que dos índices $i \in \{1, \dots, n\}$ y $j \in \{1, \dots, m\}$ son **coincidentes** si el elemento $a_{i,j}$ está definido ($a_{i,j} < \infty$).
2. Sean $i_0 \in \{1, \dots, n\}$ y sea $j_0 \in \{1, \dots, m\}$ índices, se define la **coincidencia óptima** respecto de (i_0, j_0) como el par de índices (i, j) tales que

$$(i, j) = \begin{cases} \arg \min_{(\alpha, \beta)} (a_{\alpha, \beta} / \alpha \in \{i_0, i_0 + 1\} \text{ y } \beta \in \{j_0, j_0 + 1\}) & \text{si } i_0 + 1 \leq n \text{ y } j_0 + 1 \leq m \\ (i_0, j_0 + 1) & \text{si } i_0 = n \text{ y } j_0 < m \\ (i_0 + 1, j_0) & \text{si } i_0 < n \text{ y } j_0 = m \end{cases}$$

En estas condiciones, se dice que j es **coincidencia óptima** de i respecto de (i_0, j_0) .

Por construcción, se verifica que el valor $\min(a_{i,j-1}, a_{i-1,j}, a_{i-1,j-1}) + d(x_i, y_j)$ indica el coste del camino óptimo que une $(1, 1)$ con (i, j) teniendo en cuenta las restricciones temporales. En particular se tiene el siguiente resultado.

Lema 2.36. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales, sea Φ la curva de deformación y sea $A_{DTW} = (a_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$ la matriz del algoritmo DTW entonces se verifica que para cada $s = 1, \dots, P$

$$a_{\phi_X(s), \phi_Y(s)} = \sum_{i=1}^s d(x_{\phi_X(i)}, y_{\phi_Y(i)})$$

Demostración. Claramente, como $\Phi(1) = (1, 1)$ entonces $a_{1,1} = d(x_1, y_1)$. Por las restricciones temporales sobre la curva de deformación, se cumple que

$$\Phi(s) = \begin{cases} \Phi(s-1) + (1, 0) \\ \Phi(s-1) + (0, 1) \\ \Phi(s-1) + (1, 1) \end{cases}$$

Razonamos por el principio de inducción matemática sobre s . Para $s = 1$ el resultado es trivialmente cierto, luego se supone que

$$a_{\phi_X(s), \phi_Y(s)} = \sum_{i=1}^s d(x_{\phi_X(i)}, y_{\phi_Y(i)})$$

y se prueba para $s + 1$.

$$a_{\phi_X(s+1), \phi_Y(s+1)} = d(x_{\phi_X(s+1)}, y_{\phi_X(s+1)}) + \min(a_{\phi_X(s+1)-1, \phi_Y(s+1)}, a_{\phi_X(s+1), \phi_Y(s+1)-1}, a_{\phi_X(s+1)-1, \phi_Y(s+1)-1})$$

Por definición de curva de deformación como camino de deformación de coste mínimo, y dadas las restricciones temporales impuestas sobre los caminos de deformación, se verifica que el

$$\min(a_{\phi_X(s+1)-1, \phi_Y(s+1)}, a_{\phi_X(s+1), \phi_Y(s+1)-1}, a_{\phi_X(s+1)-1, \phi_Y(s+1)-1})$$

se alcanza en $a_{\phi_X(s), \phi_Y(s)}$. Por tanto, aplicando la hipótesis de inducción se tiene que

$$\begin{aligned} a_{\phi_X(s+1), \phi_Y(s+1)} &= d(x_{\phi_X(s+1)}, y_{\phi_X(s+1)}) + a_{\phi_X(s), \phi_Y(s)} \\ &= d(x_{\phi_X(s+1)}, y_{\phi_X(s+1)}) + \sum_{i=1}^s d(x_{\phi_X(i)}, y_{\phi_Y(i)}) \end{aligned}$$

Luego se concluye por el principio de inducción matemática. □

Aplicando el lema anterior, el siguiente teorema garantiza que el resultado del algoritmo DTW expuesto en la *pág.* 36 es efectivamente la distancia DTW.

Teorema 2.37. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales, sea $A_{DTW} = (a_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$ la matriz DTW de comparación de \mathbf{X} respecto de \mathbf{Y} y sea Φ la curva de deformación con P pares de índices, entonces:

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = a_{n,m} = \sum_{k=1}^P d(x_{\phi_X(k)}, y_{\phi_Y(k)}) \quad (2.13)$$

Demostración. Basta aplicar el lema anterior (*Lema 2.36*) para $s = P$ pues $\Phi(P) = (n, m)$ y, por tanto:

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^P d(x_{\phi_X(k)}, y_{\phi_Y(k)}) = a_{n,m}$$

□

El algoritmo *DTW* expuesto anteriormente reduce la complejidad del problema, pues basta con construir adecuadamente la matriz DTW. Así, si la longitud de las series temporales muestra y prueba son n y m respectivamente, la complejidad del algoritmo anterior es de $\mathcal{O}(nm)$. Por tanto, suponiendo $n \geq m$ la complejidad es de $\mathcal{O}(n^2)$.

Restricciones supuestas sobre las series temporales y su relación.

Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales con matriz DTW dada por $A_{DTW} = (a_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$, para que el algoritmo DTW sea completo (en cualquiera de las versiones que se presenten) y tenga sentido como tal se han supuesto las siguientes hipótesis sobre las series temporales y su relación.

1. Cada índice de la serie muestra debe coincidir con al menos un índice de la serie prueba.
Es decir,

para cada $i \in \{1, \dots, n\}$ existe un $j \in \{1, \dots, m\}$ tal que $a_{i,j}$ está definido

2. Cada índice de la serie prueba debe coincidir con al menos un índice de la serie muestra.
Es decir,

para cada $j \in \{1, \dots, m\}$ existe un $i \in \{1, \dots, n\}$ tal que $a_{i,j}$ está definido

En general, las coincidencias descritas en 1 y 2 se suelen dar en más de un par de índices. En la versión primera del algoritmo DTW (pág. 36) cada índice de la serie prueba coincide con todos los índices de la serie muestra y viceversa.

3. El primer índice de la serie muestra debe coincidir con el primer índice de la serie prueba.
El último índice de la serie muestra debe coincidir con el último índice de la serie prueba.
Esto es:

$$a_{1,1} < \infty \quad \text{y} \quad a_{n,m} < \infty$$

Para finalizar la sección, se presenta un ejemplo de cálculo de la matriz DTW.

Ejemplo 2.38. Consideramos las series temporales $\mathbf{X} = [3, 3, 6, 0, 3, 3]$ e $\mathbf{Y} = [3, 3, 3, 6, 0, 3]$ (véase *ej. 2.30*). Denotamos por $E = (e_{i,j})_{1 \leq i,j \leq 6}$ la matriz de distancias para la distancia dada por $d(a, b) = |a - b|$ (es decir, $e_{i,j} = |x_i - y_j|$) y por A_{DTW} la matriz del algoritmo. Aplicando las ecuaciones (2.10), (2.11) y (2.12) se tiene que:

$$E = \begin{pmatrix} 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 \\ 3 & 3 & 3 & 6 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad A_{DTW} = \begin{pmatrix} 0 & 0 & 0 & 3 & 6 & 6 \\ 0 & 0 & 0 & 3 & 6 & 6 \\ 3 & 3 & 3 & 0 & 6 & 9 \\ 6 & 6 & 6 & 6 & 0 & 3 \\ 6 & 6 & 6 & 9 & 3 & 0 \\ 6 & 6 & 6 & 9 & 6 & 0 \end{pmatrix}$$

2.2.5. Algoritmo para hallar la curva de deformación

Además del valor de la distancia DTW interesa especialmente la curva de deformación, pues aporta gran cantidad de información útil y necesaria en la comparación de las series temporales. En esta sección se presenta un algoritmo capaz de hallar la curva de deformación a partir de la matriz DTW ([7]).

Teorema 2.39. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales, el conjunto

$$\Gamma = \{(i_s, j_s) \mid (i_s, j_s) \text{ es coincidencia óptima respecto de } (i_{s-1}, j_{s-1}), s = 2, \dots, P-1\} \\ \text{con } (i_1, j_1) = (1, 1) \text{ y } (i_P, j_P) = (n, m)$$

es una curva de deformación de la comparación de \mathbf{X} frente a \mathbf{Y} .

Demostración. En primer lugar, veamos qué Γ es camino de deformación. Γ verifica que:

1. $\Gamma(1) = (1, 1)$, $\Gamma(s) = (n, m)$
2. $\Gamma(s) = (i_s, j_s) = \begin{cases} (i_{s-1} + 1, j_{s-1} + 1) = \Gamma(s-1) + (1, 1) \\ (i_{s-1} + 1, j_{s-1}) = \Gamma(s-1) + (1, 0) \\ (i_{s-1}, j_{s-1} + 1) = \Gamma(s-1) + (0, 1) \end{cases}$

Luego verifica las tres condiciones de los caminos de deformación.

Aplicando la definición de coincidencia óptima sobre la ecuación de recursividad del algoritmo DTW, se verifica que para cada $s = 2, \dots, P$ se cumple

$$a_{i_s, j_s} = d(x_{i_s}, y_{j_s}) + a_{i_{s-1}, j_{s-1}}$$

Luego

$$a_{n, m} = a_{i_P, j_P} = d(x_{i_P}, y_{j_P}) + a_{i_{P-1}, j_{P-1}}$$

Razonando de forma recursiva se tiene que

$$a_{n, m} = \sum_{s=1}^P d(x_{i_s}, y_{j_s})$$

Además, como Γ es el camino de deformación de coste mínimo, pues por el *Teorema 2.37* ($a_{n, m}$ es la distancia DTW), entonces Γ es curva de deformación. □

En ningún momento se afirma que el camino Γ del teorema anterior sea la única curva de deformación. Es más, la curva de deformación puede no ser única (véase el *Ejemplo 2.40*) y depende de algunas consideraciones de la implementación del algoritmo.

El algoritmo que permite el cálculo de la curva de deformación, como consecuencia del teorema anterior viene definido como sigue. Este se basa en recorrer adecuadamente la matriz DTW.

Algoritmo para hallar la curva de deformación

Sea $A_{DTW} = (a_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$ la matriz obtenida por el algoritmo DTW.

1. Se comienza con el par de índices $\delta(0) = (n, m)$.
2. Se define $\delta(1)$ como

$$\delta(1) = \arg \min_{(\alpha, \beta)} (a_{\alpha, \beta} : \alpha \in \{n, n-1\} \text{ y } \beta \in \{m, m-1\}) \quad (2.14)$$

3. Recurrentemente, para cada $k = 1, \dots, (P-1)$ se define

$$\delta(k) = \arg \min_{(\alpha, \beta)} (a_{\alpha, \beta} : \alpha \in \{n-k-1, n-k\} \text{ y } \beta \in \{m-k-1, m-k\}) \quad (2.15)$$

donde se supone que $\delta(P-1) = (1, 1)$

4. Entonces $\Phi(k) = \delta(P-k)$, $k = 1, \dots, P$

Ejemplo 2.40. Consideramos las series temporales $\mathbf{X} = [3, 3, 6, 0, 3, 3]$ e $\mathbf{Y} = [3, 3, 3, 6, 0, 3]$ (véanse *ej. 2.30* y *ej. 2.38*) cuya matriz DTW es

$$A_{DTW} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & 3 & 6 & 6 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 3 & 6 & 6 \\ 3 & 3 & 3 & \mathbf{0} & 6 & 9 \\ 6 & 6 & 6 & 6 & \mathbf{0} & 3 \\ 6 & 6 & 6 & 9 & 3 & \mathbf{0} \\ 6 & 6 & 6 & 9 & 6 & \mathbf{0} \end{pmatrix}$$

La curva de deformación (marcada en negrita en la matriz anterior), según el algoritmo presentado para tal fin, viene dada por:

$$\begin{aligned} \Phi(1) &= (1, 1), & \Phi(2) &= (2, 1), & \Phi(3) &= (2, 2), & \Phi(4) &= (2, 3) \\ \Phi(5) &= (3, 4), & \Phi(6) &= (4, 5), & \Phi(7) &= (5, 6), & \Phi(8) &= (6, 6) \end{aligned}$$

En el ejemplo se puede observar como la definición dada para la curva de deformación no implica su unicidad. Es más, es también curva de deformación la definida como:

$$\begin{aligned} \Psi(1) &= (1, 1), & \Psi(2) &= (1, 2), & \Psi(3) &= (1, 3), & \Psi(4) &= (2, 3), \\ \Psi(5) &= (3, 4), & \Psi(6) &= (4, 5), & \Psi(7) &= (5, 6), & \Psi(8) &= (6, 7). \end{aligned}$$

La elección de una u otra curva de deformación depende exclusivamente de los detalles en la implementación del algoritmo a la hora de resolver (2.15). La curva Φ es la obtenida por la implementación presentada en el *Sección B.2.2*.

Significado geométrico de la curva de deformación.

Con la notación de la curva de deformación $\Phi(k) = (\phi_X(k), \phi_Y(k))$, según la *Definición 2.25* se tienen tres posibles relaciones entre $\Phi(k)$ y $\Phi(k+1)$. Geométricamente, estas relaciones están enlazadas con un “movimiento” sobre la matriz A_{DTW} , y suponen un comportamiento en el transcurso del tiempo entre ambas series.

- Una *deformación en el tiempo por eliminación* en $k+1$ se produce cuando

$$\phi_X(k+1) = \phi_X(k) + 1 \quad \text{y} \quad \phi_Y(k+1) = \phi_Y(k)$$

y representa un *movimiento en horizontal* en la matriz A_{DTW} . Esto implica que el tiempo en la secuencia de prueba \mathbf{Y} se *detuvo*. Cuando se produce una sucesión de movimientos en horizontal se dice que la secuencia prueba se desaceleró.

- Una *deformación en el tiempo por inserción* en $k+1$ se produce cuando

$$\phi_X(k+1) = \phi_X(k) \quad \text{y} \quad \phi_Y(k+1) = \phi_Y(k) + 1$$

y representa un *movimiento en vertical* en la matriz A_{DTW} . Esto implica que el tiempo de la secuencia muestra \mathbf{X} se *detuvo*. Cuando se produce una sucesión de movimientos en vertical se dice que que la secuencia prueba se aceleró.

- Una *coincidencia temporal* en $k+1$ entre ambas series se produce cuando

$$\phi_X(k+1) = \phi_X(k) + 1 \quad \text{y} \quad \phi_Y(k+1) = \phi_Y(k) + 1$$

y representa un *movimiento en diagonal* en la matriz A_{DTW} . Cuando se produce una sucesión de movimientos diagonales se dice que ambas mantienen la misma velocidad.

2.2.6. Algoritmo para determinar las asociaciones de índices

Las asociaciones entre índices de dos series temporales $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ (véase *Definición 2.35*) establecen cuando un mismo “evento” sucede en un momento temporal diferente y/o a diferentes velocidades. En el ejemplo introductorio de la *Figura 2.7* el evento “comienza a descender” se produce en el momento 4 de la primera serie y en el 6 de la segunda serie ([7] y [8]).

A partir de la curva de deformación se establece y define la relación de asociación entre los índices de la serie muestra con la serie prueba. En la siguiente definición se desarrolla el concepto.

Definición 2.41. Sean \mathbf{X} e \mathbf{Y} dos series temporales muestra y prueba con conjuntos de índices $\{1, \dots, n\}$ y $\{1, \dots, m\}$ respectivamente, sea $\Phi = (\phi_x, \phi_y)$ la curva de deformación de cardinal P , se dice que un índice $i \in \{1, \dots, n\}$ de la serie muestra se asocia con un índice $j \in \{1, \dots, m\}$ de la serie prueba si existe $k \in \{1, \dots, P\}$ tal que $(i, j) = \Phi(k)$.

En particular, interesa dado un índice i de la serie muestra conocer los índices j de la serie prueba que se asocian con i . En principio, se pueden dar tres situaciones que dependen de la relación temporal (deformación o coincidencia) en dicho momento.

- Si se ha producido una eliminación entonces existen al menos dos índices de la serie muestra que se asocian con un mismo índice de la serie prueba.
- Si se ha producido una inserción entonces para un índice de la serie muestra existen al menos dos índices de la serie prueba que se asocian con dicho índice.

Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales arbitrarias y sea $\Phi = (\phi_X, \phi_Y)$ la curva de deformación de cardinal P . Se denota por $\gamma(s)$, $s = 1, \dots, n$, al conjunto de los índices que se asocian con s . Es decir,

$$\gamma(s) = \{\phi_Y(t) : t \in \{1, \dots, P\} \text{ tales que } \phi_X(t) = s\}$$

O escrito de otra forma

$$\gamma(s) = \{j \in \{1, \dots, m\} \text{ tales que } (s, j) \in \Phi\}$$

La técnica para determinar γ para cada s es recorrer de forma ordenada la curva de deformación observando la primera componente de cada par de índices.

Ejemplo 2.42. Consideramos las series temporales $\mathbf{X} = [3, 3, 6, 0, 3, 3]$ e $\mathbf{Y} = [3, 3, 3, 6, 0, 3]$ (véanse *ej. 2.30*, *ej. 2.38* y *ej. 2.40*) cuya curva de deformación viene dada por:

$$\begin{aligned} \Phi(1) &= (1, 1), & \Phi(2) &= (2, 1), & \Phi(3) &= (2, 2), & \Phi(4) &= (2, 3) \\ \Phi(5) &= (3, 4), & \Phi(6) &= (4, 5), & \Phi(7) &= (5, 6), & \Phi(8) &= (6, 6) \end{aligned}$$

Las asociaciones de índices son:

$$\begin{aligned} \gamma(1) &= 1 & \gamma(2) &= \{1, 2, 3\} & \gamma(3) &= 4 \\ \gamma(4) &= 5 & \gamma(5) &= 6 & \gamma(6) &= 6 \end{aligned}$$

Finalmente, se puede observar el significado de las deformaciones temporales en función de las identificaciones de índices. Recuperando la primera curva de deformación del ejemplo anterior:

1. Si se produce una **inserción** en k (caso de $k = 3$, $k = 4$ en Φ) entonces el conjunto $\gamma(2)$ tiene al menos dos elementos (en este caso tres por haberse producido dos inserciones).
2. Si se produce una **eliminación** en k (caso de $k = 2$ o $k = 8$ en Φ) entonces la intersección $\gamma(1) \cap \gamma(2) = \{1\} \neq \emptyset$.

2.2.7. Otras versiones del algoritmo DTW

Existen diferentes versiones del algoritmo DTW que suponen una mejora respecto de la *versión primera* que se presenta en la *Sección 2.2.4*. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales cuyos coeficientes quedan definidos en el espacio métrico (\mathbb{V}, d) , y sea $A_{DTW} = (a_{i,j})_{i=1, \dots, n; j=1, \dots, m}$ la matriz resultante del algoritmo *DTW* correspondiente. En [7] aparecen otras modificaciones diferentes a las aquí consideradas, con mayor detalle y su justificación.

1. Cambios en el patrón de recursividad.

El **patrón de recursividad** hace referencia a la *relación recursiva* que permite calcular los coeficientes de la matriz A_{DTW} . En la primera versión del algoritmo (*página 36*) este patrón está definido por (2.12). Para más detalles sobre los patrones de recursividad, y para conocer otro tipo de parones diferentes a los simétricos aquí desarrollados, consúltese [9].

Cambiando ligeramente el problema que define la distancia DTW (véase (2.7)) este se puede plantear con pesos. En la versión más general, dadas dos series temporales $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ y dado Ψ un camino de deformación de cardinal P se define la distorsión acumulada para $\Psi = (\psi_X, \psi_Y)$ respecto de \mathbf{X} e \mathbf{Y} como

$$dist_{\Psi}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^P d(x_{\psi_X(i)}, y_{\psi_Y(i)}) \cdot m_{\Psi}(i) M_{\Psi}$$

donde $m_{\Psi}(i)$ es el peso para el paso i –ésimo y M_{Ψ} es una constante de normalización que permite comparar la distorsión acumulada de dos series. En este nuevo contexto, la distancia DTW se define como

$$d_{DTW}^*(\mathbf{X}, \mathbf{Y}) = \min_{\Psi \in \mathcal{P}_{n,m}} dist_{\Psi}(\mathbf{X}, \mathbf{Y})$$

Generalmente, los pesos para el paso suelen ser constantes que no dependen ni de la serie temporal ni del valor de los índices, sino que dependen del tipo de deformación o coincidencia producida (es decir, de la relación entre $\Psi(s)$ y $\Psi(s+1)$ en las tres posibles opciones dadas en la *Definición 2.25*). En el segundo patrón, se define

$$m_{\Psi}(s) = \begin{cases} 1 & \text{si } \Psi(s) \text{ no es coincidencia} \\ 2 & \text{si } \Psi(s) \text{ es coincidencia} \end{cases}$$

Y el valor de normalización es $n + m$.

El cambio propuesto en esta variación viene de sustituir (2.12) por

$$a_{i,j} = \min\{a_{i-1,j-1} + 2 \cdot d(x_i, y_j), a_{i-1,j} + d(x_i, y_j), a_{i,j-1} + d(x_i, y_j)\} \quad (2.16)$$

Esta modificación busca “favorecer” que se produzcan deformaciones temporales frente a las coincidencias, penalizando la situación en la cual dos componentes, una de cada serie, son próximas entre sí, pero no iguales.

A los patrones de recursividad dados en (2.12) y (2.16) se les conoce como **patrones simétricos**, pues su construcción trata por igual al coeficiente $a_{i,j-1}$ y al coeficiente $a_{i-1,j}$.

2. Aplicación de restricciones de localidad.

El principal problema que presentan ambos patrones de recursividad en la versión primera del algoritmo DTW es que permite una **deformación temporal total**. Ésta se produce cuando en la curva de deformación se establece la identificación de todos los índices de una de las series temporales respecto al primer o último índice. Esta deformación supone que el total del tiempo de una de las series transcurre mientras el tiempo permanece detenido en la otra serie. En la curva de deformación se observaría que:

$$\begin{aligned} \Phi &= \{(1, 1), \dots, (n, 1), (n, 2), \dots, (n, m)\} \\ &\quad \text{ó} \\ \Phi &= \{(1, 1), \dots, (1, m), (2, m), \dots, (n, m)\} \end{aligned}$$

La solución a este problema es agregar una *restricción de localidad sobre la deformación en el tiempo*. Existen diferentes restricciones de localidad, desde funciones simples, funciones dependientes del tiempo ¹, etc.

La restricción más sencilla es la **limitación constante**. Se toma un valor $\omega \in \mathbb{N}$ de **tolerancia**, y se impiden deformaciones en el tiempo superiores a ω . Es decir, si \mathcal{L} denota el cardinal del camino de deformación, se considera el subconjunto de $\mathcal{P}_{n \times m}$ dado por:

$$\mathcal{P}_{n \times m}^* := \{\Psi = (\psi_X, \psi_Y) \in \mathcal{P}_{n \times m} \text{ tales que } \forall s = 1, \dots, \mathcal{L}(\Psi) \text{ cumplen } |\psi_X(s) - \psi_Y(s)| \leq \omega\}$$

Y se redefine la distancia DTW como

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = \min_{\Psi \in \mathcal{P}_{n,m}^*} \left(\sum_{(i,j) \in \Psi} d(x_i, y_j) \right) \quad (2.17)$$

En términos de la matriz del algoritmo DTW, $A_{DTW} = (a_{i,j})_{i,j}$, se aplica la siguiente restricción: $a_{i,j}$ estará definido ($a_{i,j} \neq \infty$) si y solo si $|i - j| \leq \omega$. Para ello se modifican las ecuaciones (2.10), (2.11) y (2.12).

¹Restricciones de Itakura y Sakoe-Chiba. Consúltese [10] para más detalle.

Algoritmo DTW versión segunda

Sea $\mathbf{X} = [x_1, \dots, x_n]$ la serie muestra e $\mathbf{Y} = [y_1, \dots, y_m]$ la serie prueba, cuyos coeficientes pertenecen al espacio métrico (\mathbb{V}, d) .

1. Dado $\omega \in \mathbb{N}$, se calculan los coeficientes de la matriz $A_{DTW} = (a_{i,j})_{i,j}$

1.1. Se define el primer elemento de la matriz DTW:

$$a_{1,1} = d(x_1, y_1) \quad (2.18)$$

1.2. Se calcula la primera fila y columna de la matriz DTW.

Para cada $j = 2, \dots, m$ se define

$$a_{1,j} = \begin{cases} d(x_1, y_j) + a_{1,j-1} & \text{Si } |i-j| \leq \omega \\ \infty & \text{Si } |i-j| > \omega \end{cases} \quad (2.19)$$

Y para cada $i = 2, \dots, m$ se define

$$a_{i,1} = \begin{cases} d(x_i, y_1) + a_{i,j-1} & \text{Si } |i-j| \leq \omega \\ \infty & \text{Si } |i-j| > \omega \end{cases} \quad (2.20)$$

1.3. Para cada $i = 2, \dots, n$ y para cada $j = 2, \dots, m$ se define el elemento $a_{i,j}$ como:

$$a_{i,j} = \begin{cases} d(x_i, y_j) + \min(a_{i,j-1}, a_{i-1,j}, a_{i-1,j-1}) & \text{Si } |i-j| \leq \omega \\ \infty & \text{Si } |i-j| > \omega \end{cases} \quad (2.21)$$

2. La distancia entre ambas series es el término $a_{n,m}$

Con esta restricción sobre los índices se impiden deformaciones temporales de una longitud superior a ω . En principio, ω puede ser un natural arbitrario. Sin embargo, para que exista una solución y el algoritmo sea completo, además de que se cumplan todas las restricciones sobre las series prueba y muestra expuestas en la *Sección 2.2.4*, es necesario que $\omega \geq |n - m|$. Así, esta versión del algoritmo justifica la *Definición 2.35* y hace necesarias las restricciones primera y segunda expuestas en la *página 39*, superfluas en la versión primera.

El siguiente resultado justifica la importancia de que $\omega \geq |n - m|$ para la completitud del algoritmo y existencia de solución.

Proposición 2.43. *Es condición necesaria y suficiente para que exista una solución a la comparación de dos series temporales de longitud n y m , y dicha solución sea hallada por el algoritmo DTW en la versión segunda que el valor ω de tolerancia verifique que $\omega \geq |m - n|$.*

Demostración.

- Veamos que es condición suficiente. Si $\omega \geq |m - n|$ entonces el elemento $a_{n,m} \neq \infty$ por definición. Además, suponiendo $n \geq m$ se puede considerar el camino (que no tiene por qué ser la curva de deformación) dado por

$$\{(1, 1), \dots, (m, m), (m + 1, m), \dots, (n, m)\}$$

En consecuencia, existe solución y es hallada.

- Veamos que es condición necesaria. Supongamos que $\omega < |n - m|$, entonces para el coeficiente $a_{n,m}$ este no quedaría definido ($a_{n,m} = \infty$) y no existe solución a la comparación de las series.

□

3. Distancia p -DTW

Dados los vectores $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{y} = (y_1, \dots, y_m) \in \mathbb{R}^n$ se define la distancia p (también conocida como L_p) entre ambos vectores como

$$\left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}.$$

De manera análoga, se puede definir la distancia p -DTW.

Definición 2.44. Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales arbitrarias y sea Ψ un camino de deformación de orden $n \times m$ y cardinal S , se define la **función de coste de grado p** como

$$C_{\mathbf{X}, \mathbf{Y}}^p(\Psi) = \left(\sum_{(i,j) \in \Psi} d(x_i, y_j)^p \right)^{1/p} = \left(\sum_{i=1}^S d(x_{\Psi_X(i)}, y_{\Psi_Y(j)})^p \right)^{1/p} \quad (2.22)$$

Y se define la **distancia p -DTW** de dos series temporales queda como:

$$d_{p\text{-DTW}}(\mathbf{X}, \mathbf{Y}) = \min_{\Psi \in \mathcal{P}_{n,m}} C_{\mathbf{X}, \mathbf{Y}}^p(\Psi) = \left(\min_{\Psi \in \mathcal{P}_{n,m}} \sum_{(i,j) \in \Psi} d(x_i, y_j)^p \right)^{1/p} \quad (2.23)$$

Respecto a los algoritmos presentados basta realizar los cambios oportunos en los patrones de recursividad. La distancia DTW dada en la la *Definición 2.28* se corresponde con el caso de $p = 1$. El siguiente lema recoge la relación fundamental entre la dicha distancia 1-DTW (dada por (2.7) y denotada por d_{DTW}) y la distancia p -DTW.

Lema 2.45. sea \mathbb{V} el espacio de los coeficientes de las series temporales dotado de la distancia d^* , dado $p \in \mathbb{N}$, se toma $d^*(a, b) = d(a, b)^p$ y se considera la distancia DTW según la Definición 2.28 (versión primera) inducida para d^* denotada por d_{DTW}^* . Entonces para cualquier par de series temporales \mathbf{X} e \mathbf{Y} se cumple que

$$d_{p-DTW}(\mathbf{X}, \mathbf{Y}) = d_{DTW}^*(\mathbf{X}, \mathbf{Y})^{1/p}$$

Donde d_{p-DTW} está inducida por la distancia d según 2.44.

Demostración. Consideramos la distancia $\hat{d}(a, b) = (a - b)^2$, y se calcula la distancia DTW para \hat{d} denotada por \hat{d}_{DTW} . Entonces, por el Lema 2.45 $d_{2-DTW}(\mathbf{X}', \mathbf{Y}')^2 = \hat{d}_{DTW}(\mathbf{X}, \mathbf{Y})$ y por la Proposición 2.32 existen \mathbf{X}' e \mathbf{Y}' tales que:

$$d_{2-DTW}(\mathbf{X}', \mathbf{Y}')^2 = \hat{d}_{DTW}(\mathbf{X}, \mathbf{Y}) = \hat{d}_{DTW}(\mathbf{X}', \mathbf{Y}') = \sum_{i=1}^k (x_i' - y_i')^2 = d_E(\mathbf{X}', \mathbf{Y}')^2$$

□

El caso más usual, que es el implementado en la librería de Python `tslearn` utilizada para el caso de estudio, es la distancia 2-DTW². Como consecuencia inmediata de la Proposición 2.32 se verifica la siguiente relación entre la distancia 2-DTW y la distancia euclídea.

Corolario 2.46. Sean \mathbf{X} e \mathbf{Y} dos series temporales con coeficientes reales, entonces existen $\mathbf{X}' = [x_1', \dots, x_k']$ e $\mathbf{Y}' = [y_1', \dots, y_k']$ series temporales de la misma longitud tales que

$$d_{DTW}(\mathbf{X}, \mathbf{Y}) = d_{DTW}(\mathbf{X}', \mathbf{Y}') = \sqrt{\sum_{i=1}^k (x_i' - y_i')^2} = d_E(\mathbf{X}', \mathbf{Y}')$$

Demostración. El resultado es una cuestión de aplicar la definición.

$$d_{DTW}^*(\mathbf{X}, \mathbf{Y}) = \min_{\Psi \in \mathcal{P}_{n,m}} \sum_{(i,j) \in \Psi} d^*(x_i, y_j) = \min_{\Psi \in \mathcal{P}_{n,m}} \sum_{(i,j) \in \Psi} d(x_i, y_j)^p = d_{p-DTW}^p$$

□

La versión tercera del algoritmo DTW registra los cambios necesarios a realizar basados en el resultado anterior.

²Consúltense [1], sección “Time Series Clustering” subsección “k-means and Dynamic Time Warping”

El algoritmo DTW (versión tercera, p -DTW)

Sean $\mathbf{X} = [x_1, \dots, x_n]$ e $\mathbf{Y} = [y_1, \dots, y_m]$ dos series temporales cualesquiera de longitud n y m respectivamente. Sea d una distancia sobre espacio en el cual están definidos los valores de x_i e y_i , $\forall i$, dado $p \in \mathbb{N}$.

1. Se construye la matriz $A_{DTW} = (a_{i,j})_{i,j}$
 - 1.1. Se define el primer elemento de la matriz DTW (A_{DTW}) como la distancia entre x_1 e y_1 .

$$a_{1,1} = d(x_1, y_1)^p \quad (2.24)$$

- 1.2. Se calcula la primera fila y columna de la matriz DTW.
Para cada $j = 2, \dots, m$ se define

$$a_{1,j} = d(x_1, y_j)^p + a_{1,j-1} \quad (2.25)$$

Y para cada $i = 2, \dots, m$ se define

$$a_{i,1} = d(x_i, y_1)^p + a_{i-1,1} \quad (2.26)$$

- 1.3. Para cada $i = 2, \dots, n$ y para cada $j = 2, \dots, m$ se define el elemento $a_{i,j}$ como:

$$a_{i,j} = d(x_i, y_j)^p + \min(a_{i,j-1}, a_{i-1,j}, a_{i-1,j-1}) \quad (2.27)$$

2. La distancia DTW entre ambas series es el término $(a_{n,m})^{1/p}$.

4. Comentario final sobre las distintas variaciones.

En lo que resta, salvo que se especifique lo contrario se considera una distancia DTW genérica, pues en cualquiera de sus versiones (siempre que se cumplan las restricciones que la hacen completa) los algoritmos pueden ser aplicados. En particular, la biyectividad de la aplicación $f : [0, \infty) \rightarrow [0, \infty)$ dada por $f(t) = t^{1/p}$ hace que todos los resultados estudiados para d_{DTW} (Definición 2.28) sean aplicables a d_{p-DTW} . Además, cuando la restricción de localidad impuestas en (2.17) (versión segunda) cumplan las restricciones necesarias recogidas en la pág. 39 los resultados también son aplicables, pues el requisito necesario y suficiente es la existencia de curva de deformación.

Sin embargo, no se puede afirmar que las diferentes versiones sean equivalentes entre sí, en cuanto a la medida de comparación obtenida.

2.3. Notas finales sobre la distancia DTW

El adjetivo distancia o métrica aplicado a la medida de comparación dada por el algoritmo DTW (en cualquiera de sus versiones) responde motivos tradicionales, pues en ningún momento se probó que d_{DTW} define una verdadera distancia. Esta sección está basada en [6], cuyos resultados y demostraciones han sido adaptados al estudio de series temporales desarrollado en esta memoria.

Definición 2.47. Sea \mathbb{V} un espacio vectorial sobre \mathbb{R} , una **distancia** es una aplicación $d : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$ que verifica las siguientes propiedades:

- (d1) $d(x, y) = 0 \Leftrightarrow x = y, \quad \forall x, y \in \mathbb{V}.$
- (d2) $d(x, y) = d(y, x), \quad \forall x, y \in \mathbb{V}.$
- (d3) $d(x, z) = d(x, y) + d(y, z), \quad \forall x, y, z \in \mathbb{V}.$
- (d4) $d(x, y) \geq 0, \quad \forall x, y \in \mathbb{V}.$

Si verifica todos los axiomas salvo la desigualdad triangular (d3) entonces se dice que d es una **semi-métrica**.

Sea \mathbb{V} un espacio métrico con distancia d , sea \mathcal{T} el conjunto de todas las series temporales de longitud finita con coeficientes en \mathbb{V} . Se define la aplicación $\delta : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$ dada por $\delta(\mathbf{X}, \mathbf{Y}) = d_{DTW}(\mathbf{X}, \mathbf{Y})$.

En primer lugar, δ verifica (d2) y (d4).

- (d4) $\delta(\mathbf{X}, \mathbf{Y}) = \sum_{(i,j) \in \Phi} d(x_i, y_j) \geq 0$ pues $d(x_i, y_j) \geq 0$ para todo $i = 1, \dots, n$ y $j = 1, \dots, m$ por ser d una distancia.

- (d2) $\delta(\mathbf{X}, \mathbf{Y}) = \delta(\mathbf{Y}, \mathbf{X})$, pues basta observar que si se construye la matriz DTW asociada a cada una de las distancias, éstas resultan ser transpuestas, consecuencia del algoritmo de construcción y por ser la distancia d simétrica.

Sin embargo, δ no verifica (d1). Dada una serie temporal $\mathbf{X} = [x_1, \dots, x_n]$ si se define la serie $\mathbf{Y} = [y_1, \dots, y_{n+5}]$ como:

$$y_k = \begin{cases} x_k & \text{si } k \leq n \\ x_n & \text{si } k > n \end{cases}$$

Por tanto, como \mathbf{Y} es una deformación temporal de \mathbf{X} en su último término se tiene que $\delta(\mathbf{X}, \mathbf{Y}) = 0$ siendo $\mathbf{X} \neq \mathbf{Y}$. En cuanto a la desigualdad triangular, el siguiente ejemplo muestra que no necesariamente ha de verificarse.

Ejemplo 2.48. Se consideran las series temporales $\mathbf{X} = [0, 1]$, $\mathbf{Y} = [0, 1, 1]$ y $\mathbf{Z} = [0, 2]$. Entonces:

$$\begin{aligned}\delta(\mathbf{Y}, \mathbf{X}) &= 0 \\ \delta(\mathbf{X}, \mathbf{Z}) &= 1 \\ \delta(\mathbf{Y}, \mathbf{Z}) &= 2 \\ \delta(\mathbf{Y}, \mathbf{Z}) &= 2 > \delta(\mathbf{Y}, \mathbf{X}) + \delta(\mathbf{X}, \mathbf{Z}) = 0 + 1\end{aligned}$$

Por lo que no se verifica la desigualdad triangular en \mathcal{T} .

2.3.1. Expansión por deformación de una serie temporal

La relación fundamental entre dos series temporales es el efecto de deformación en el tiempo. Analíticamente este efecto se observa cuando aparecen repetidos de forma consecutiva algún término de la serie temporal.

En la *Proposición 2.32* se trató de forma implícita este concepto, pues la serie \mathbf{X}' construida a partir de \mathbf{X} es una extensión de dicha serie. Dicha proposición, junto a su corolario, presentan una gran relevancia en las caracterizaciones de esta sección.

Definición 2.49. Se dice que una serie $\mathbf{Y} = [y_1, \dots, y_m]$ es una *extensión por deformación* (o simplemente *extensión*) de la serie $\mathbf{X} = [x_1, \dots, x_n]$ si:

1. Para cada $i \in \{1, \dots, n\}$ existe $j \in \{1, \dots, m\}$ tal que $x_i = y_j$.
2. Se puede establecer una identificación entre los índices $\gamma : \{1, \dots, m\} \rightarrow \{1, \dots, n\}$ tal que:
 - a) $\gamma(t) \leq \gamma(t+1)$, $\forall t = 1, \dots, m$
 - b) $\gamma(1) = 1$
 - c) $\gamma(m) = n$
 - d) $y_t = x_{\gamma(t)}$

Se denota por $\mathbf{Y} \gg \mathbf{X}$ el caso en que \mathbf{Y} es extensión de \mathbf{X} .

La definición anterior viene a destacar que en \mathbf{Y} aparecen todos los términos de \mathbf{X} en el mismo orden cronológico y algunos de ellos pueden aparecer repetidos de forma consecutiva. Es claro que las condiciones de extensión implican que $m \geq n$.

Ejemplo 2.50. Sea $\mathbf{X} = [1, 2, 3, 4]$, son extensiones las series temporales $[1, 1, 1, 2, 3, 4]$, $[1, 1, 2, 2, 3, 3, 4, 4]$ y $[1, 2, 2, 3, 4]$.

De forma inherente al concepto de extensión, se buscan aquellas series temporales que no pueden ser extensión de ninguna serie temporal.

Definición 2.51. *Se dice que una serie temporal \mathbf{X} está en **forma condensada** si no es extensión de ninguna otra serie temporal distinta a sí misma.*

Es claro que si \mathbf{X} está en forma condensada entonces la única serie de la que \mathbf{X} es extensión es ella misma.

Teorema 2.52. $\mathbf{X} = [x_1, \dots, x_n]$ *está en forma condensada si y solo si $x_i \neq x_{i+1}$, $i = 1, \dots, n-1$*

Demostración. Manteniendo la notación de la *Definición 2.49*. Sea $\gamma : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$ la identificación entre índices de ambas series.

\Leftarrow Supongamos que se verifica que para cada i se tiene que $x_i \neq x_{i+1}$. Razonamos por reducción al absurdo. Suponemos que $\mathbf{X} = [x_1, \dots, x_n]$ es una extensión de $\mathbf{Y} = [y_1, \dots, y_m]$ con $\mathbf{Y} \neq \mathbf{X}$.

Entonces por hipótesis se cumple que $\gamma(t) < \gamma(t+1)$ (pues si $\gamma(t) = \gamma(t+1)$ entonces $x_t = x_{t+1}$ contra hipótesis). Luego como $x_{\gamma(t)} = y_t$ se tiene que $m \leq n$, y por ser \mathbf{X} extensión de \mathbf{Y} se tiene que $m \geq n$. En conclusión, $m = n$ y $\mathbf{X} = \mathbf{Y}$, absurdo.

\Rightarrow Supongamos que $\mathbf{X} = [x_1, \dots, x_n]$ está en forma condensada. Razonamos por reducción al absurdo. Suponemos que existe un $t \in \{1, \dots, n-1\}$ tal que $x_t = x_{t+1}$. Entonces es claro que para la serie temporal $\mathbf{Y} = [x_1, \dots, x_t, x_{t+2}, \dots, x_n]$ la serie \mathbf{X} es una extensión, contradiciendo que \mathbf{X} está en forma condensada.

□

En consecuencia, para hallar la forma condensada de una serie temporal basta con “suprimir” los términos que aparecen repetidos de forma consecutiva.

A la acción de “*suprimir términos repetidos*” se le denomina **condensación** y a la acción de “*duplicar términos consecutivos*” se le denomina **extensión**. Las deformaciones temporales que se producen en la comparación de dos series (eliminación e inserción) se reducen a extensiones de ambas series (ver *Proposición 2.32*) o al mantenimiento invariante de la serie muestra y la realización de extensiones (inserción) y condensaciones (eliminación) sobre la serie prueba.

Proposición 2.53. *La forma condensada de una serie temporal arbitraria $\mathbf{Y} \in \mathcal{T}$ existe y es única.*

Demostración. Es consecuencia inmediata de la técnica para hallar la forma condensada. □

Es claro que si la serie temporal \mathbf{Y} es una extensión de la serie \mathbf{X} entonces $\delta(\mathbf{X}, \mathbf{Y}) = 0$. El recíproco a dicha afirmación también es cierto con algunos matices, y permitirá la aplicación de la distancia DTW al algoritmo de k-medias. De ello se tiene la importancia del siguiente teorema, que caracteriza el concepto de extensión mediante la distancia DTW, y justifica la afirmación “la distancia DTW es nula frente deformaciones en el tiempo”.

Teorema 2.54. Sean $\mathbf{X}, \mathbf{Y} \in \mathcal{T}$, $\mathbf{X} = [x_1, \dots, x_n]$, $\mathbf{Y} = [y_1, \dots, y_m]$.

1. Si $\mathbf{X} \gg \mathbf{Y}$ o $\mathbf{Y} \gg \mathbf{X}$ entonces $\delta(\mathbf{X}, \mathbf{Y}) = 0$.
2. Si $\delta(\mathbf{X}, \mathbf{Y}) = 0$ entonces existe $\mathbf{X}^* \in \mathcal{T}$ tal que $\mathbf{X} \gg \mathbf{X}^*$ e $\mathbf{Y} \gg \mathbf{X}^*$.
3. Si $\mathbf{Y} \gg \mathbf{X}$ entonces $\delta(\mathbf{Y}, \mathbf{Z}) \geq \delta(\mathbf{X}, \mathbf{Z})$ para cualquier serie $\mathbf{Z} \in \mathcal{T}$.
4. Sean \mathbf{X} e \mathbf{Y} tales que $\mathbf{X} \gg \mathbf{Z}$ e $\mathbf{Y} \gg \mathbf{Z}$ entonces $\delta(\mathbf{X}, \mathbf{Y}) = 0$.

Demostración.

1. La primera afirmación es trivial, pues la distancia DTW es capaz de suprimir los efectos de las extensiones (véase *Proposición 2.31*)
2. Respecto a la segunda afirmación basta recurrir al *corolario 2.33*, pues existen extensiones \mathbf{X}' e \mathbf{Y}' de las series originales \mathbf{X} e \mathbf{Y} respectivamente tales que

$$0 = \delta(\mathbf{X}, \mathbf{Y}) = \delta(\mathbf{X}', \mathbf{Y}') = \sum_{i=1}^k d(x_i', y_i') \Rightarrow x_i' = y_i', \forall i \Rightarrow \mathbf{X}' = \mathbf{Y}'$$

y basta considerar la forma condensada de \mathbf{X}' .

3. Por último, para la tercera afirmación basta observar que sean Φ y Ψ las curvas de deformación respectivas obtenidas de $\delta(\mathbf{Y}, \mathbf{Z})$ y $\delta(\mathbf{X}, \mathbf{Z})$ entonces $\Psi \subseteq \Phi$ (pues para cada $i \in \{1, \dots, n\}$ existe $j \in \{i, \dots, m\}$ tal que $x_i = y_j$) lo que concluye por el *Teorema 2.37*.
4. Las series \mathbf{X} e \mathbf{Y} son el resultado de realizar extensiones sobre $\mathbf{Z} = [z_1, \dots, z_p]$. Supongamos que \mathbf{Z} tiene longitud p y definimos $\eta_X : \{1, \dots, p\} \rightarrow \mathbb{N}$ y $\eta_Y : \{1, \dots, p\} \rightarrow \mathbb{N}$ tales que $\eta_X(i)$ es el número de veces que se repite el coeficiente z_i en la serie \mathbf{X} y $\eta_Y(i)$ es el número de veces que se repite el coeficiente z_i en la serie \mathbf{Y} . Construimos \mathbf{U} tal que el número de veces consecutivas que se repite z_i en \mathbf{U} es $\eta_X(i) + \eta_Y(i)$. (Es decir, $\eta_U = \eta_X + \eta_Y$).

Con ello se ha construido \mathbf{U} que es extensión tanto de \mathbf{X} como de \mathbf{Y} . Aplicando el enunciado tercero del propio problema con $\mathbf{U} \gg \mathbf{X} \gg \mathbf{Z}$ se tiene

$$\delta(\mathbf{U}, \mathbf{Y}) \geq \delta(\mathbf{X}, \mathbf{Y}) \geq \delta(\mathbf{Z}, \mathbf{Y})$$

Y por el primer enunciado, como $\mathbf{U} \gg \mathbf{Y} \gg \mathbf{Z}$

$$0 = \delta(\mathbf{U}, \mathbf{Y}) \geq \delta(\mathbf{X}, \mathbf{Y}) \geq \delta(\mathbf{Z}, \mathbf{Y}) = 0$$

□

2.3.2. La identificación por deformación

En esta sección se busca caracterizar a la distancia DTW para que cumpla la propiedad (d1) de la *Definición 2.47* extendiéndola a un espacio “adecuado”.

Definición 2.55. Se define la *identificación por deformación* en \mathcal{T} como la relación

$$\mathbf{X} \sim \mathbf{Y} \iff \delta(\mathbf{X}, \mathbf{Y}) = 0$$

Proposición 2.56. La *identificación por deformación* establece una relación de equivalencia en \mathcal{T}

Demostración. Veamos que verifica las propiedades necesarias:

1. Reflexividad: $\delta(\mathbf{X}, \mathbf{X}) = 0$ (la diagonal principal de la matriz DTW es de ceros) luego $\mathbf{X} \sim \mathbf{X}$.
2. Simetría: Como las matrices correspondientes al intercambiar los papeles a la serie muestra y prueba son transpuestas (por la propia naturaleza del algoritmo DTW) entonces δ es simétrica. Por tanto, si $\mathbf{X} \sim \mathbf{Y}$ entonces $0 = \delta(\mathbf{X}, \mathbf{Y}) = \delta(\mathbf{Y}, \mathbf{X})$ e $\mathbf{Y} \sim \mathbf{X}$.
3. Transitividad: si $\mathbf{X} \sim \mathbf{Y}$ e $\mathbf{Y} \sim \mathbf{Z}$ veamos que $\mathbf{X} \sim \mathbf{Z}$.
 - Como $\delta(\mathbf{X}, \mathbf{Y}) = 0$ entonces existe \mathbf{X}^* forma condensada tal que $\mathbf{X} \gg \mathbf{X}^*$ e $\mathbf{Y} \gg \mathbf{X}^*$
 - Como $\delta(\mathbf{Z}, \mathbf{Y}) = 0$ entonces existe \mathbf{Z}^* forma condensada tal que $\mathbf{Z} \gg \mathbf{Z}^*$ e $\mathbf{Y} \gg \mathbf{Z}^*$

Entonces \mathbf{X}^* y \mathbf{Z}^* son formas condensadas de \mathbf{Y} , y por la unicidad de la forma condensada $\mathbf{X}^* = \mathbf{Z}^*$. Así \mathbf{X} y \mathbf{Z} son extensiones (deformaciones temporales) de la misma serie y dado que la distancia DTW detecta e ignora las deformaciones en el tiempo se tiene que $\delta(\mathbf{X}, \mathbf{Z}) = 0$

□

A la relación de equivalencia se le denota por \mathcal{R} , y al conjunto de todas las clases de equivalencia se lo denota por \mathcal{T}/\mathcal{R} . El *Ejemplo 2.48* muestra que, aunque dos series temporales $\mathbf{X} = [0, 1]$ e $\mathbf{Y} = [0, 1, 1]$ pertenezcan a la misma clase de equivalencia ($\delta(\mathbf{X}, \mathbf{Y}) = 0$), la distancia de estas series a otra serie temporal arbitraria $\mathbf{Z} = [0, 2]$ no tiene por qué coincidir.

$$\delta(\mathbf{X}, \mathbf{Z}) = 1 \neq 2 = \delta(\mathbf{Y}, \mathbf{Z})$$

Es decir, δ no se puede extender directamente a \mathcal{T}/\mathcal{R} . En consecuencia, se define $\delta^* : \mathcal{T}/\mathcal{R} \times \mathcal{T}/\mathcal{R} \rightarrow \mathbb{R}$ como

$$\delta^*([\mathbf{X}], [\mathbf{Y}]) = \inf_{\mathbf{X} \in [\mathbf{X}]} \inf_{\mathbf{Y} \in [\mathbf{Y}]} \delta(\mathbf{X}, \mathbf{Y})$$

El siguiente resultado da una caracterización de las clases de equivalencia consecuencia inmediata del *Teorema 2.54* a partir de un representante de cada clase en forma condensada.

Proposición 2.57. *Para cada clase de equivalencia $[\mathbf{X}]$ de la identificación por deformación existe una única serie en forma condensada \mathbf{X}^* tal que:*

$$[\mathbf{X}] = \{ \mathbf{Y} \in \mathcal{T} \text{ tales que } \delta(\mathbf{X}^*, \mathbf{Y}) = 0 \} = [\mathbf{X}^*]$$

Demostración. Basta con aplicar adecuadamente el *Teorema 2.54*.

- Existencia: si $\mathbf{X}, \mathbf{Y} \in [\mathbf{Z}]$ entonces $\delta(\mathbf{X}, \mathbf{Y}) = 0$. Por tanto, existe $\mathbf{X}^* \in \mathcal{T}$ tal que $\mathbf{X} \gg \mathbf{X}^*$ e $\mathbf{Y} \gg \mathbf{X}^*$. Basta con tomar la forma condensada de \mathbf{X}^* , denotada por \mathbf{X}^+ , y se tiene que $\mathbf{X} \gg \mathbf{X}^+ \gg \mathbf{X}^*$ e $\mathbf{Y} \gg \mathbf{X}^+ \gg \mathbf{X}^*$. Así pues, $\delta(\mathbf{X}^+, \mathbf{Y}) = 0$ y $\delta(\mathbf{X}^+, \mathbf{X}) = 0$.
- Unicidad: veamos qué $[\mathbf{X}]$ solo contiene una única serie en forma condensada. Supongamos que existen dos series en forma condensada, $\mathbf{X}^*, \mathbf{Y}^* \in [\mathbf{X}]$. Entonces, existe \mathbf{Z}^* en forma condensada tal que $\mathbf{X}^* \gg \mathbf{Z}^*$ e $\mathbf{Y}^* \gg \mathbf{Z}^*$. Como una serie en forma condensada solo puede ser extensión de sí misma se tiene que $\mathbf{X}^* = \mathbf{Z}^*$ e $\mathbf{Y}^* = \mathbf{Z}^*$, y por tanto $\mathbf{X}^* = \mathbf{Y}^*$.

□

Como conclusión final de esta sección, los siguientes resultados justifican que δ^* es una semi-métrica.

Teorema 2.58. *Sea la δ^* la aplicación inducida por la distancia DTW δ sobre la identificación por deformación entonces δ^* define una semi-métrica sobre \mathcal{T}/\mathcal{R} y además si \mathbf{X}^* e \mathbf{Y}^* denotan el único elemento en forma condensada de las clases $[\mathbf{X}]$ e $[\mathbf{Y}]$ respectivamente, se verifica que*

$$\delta^*([\mathbf{X}], [\mathbf{Y}]) = \delta(\mathbf{X}^*, \mathbf{Y}^*)$$

Demostración.

1. En primer lugar veamos igualdad dada. Para ello basta observar que por el *Teorema 2.54* si \mathbf{X} es una extensión de \mathbf{X}' entonces para cualquier $\mathbf{Z} \in \mathcal{T}$ se tiene que $\delta(\mathbf{X}, \mathbf{Z}) \geq \delta(\mathbf{X}', \mathbf{Z})$. Por tanto, si \mathbf{X}^* es la forma condensada se tiene que para todo $\mathbf{X}' \in [\mathbf{X}]$ se cumple que $\delta(\mathbf{X}^*, \mathbf{Z}) \leq \delta(\mathbf{X}', \mathbf{Z})$. Por tanto,

$$\inf_{\mathbf{X} \in [\mathbf{X}]} \delta(\mathbf{X}, \mathbf{Z}) = \delta(\mathbf{X}^*, \mathbf{Z})$$

Así pues, aplicando el razonamiento anterior a cada una de las clases

$$\delta^*([\mathbf{X}], [\mathbf{Y}]) = \inf_{\mathbf{X} \in [\mathbf{X}]} \inf_{\mathbf{Y} \in [\mathbf{Y}]} \delta(\mathbf{X}, \mathbf{Y}) = \delta(\mathbf{X}^*, \mathbf{Y}^*)$$

2. En segundo lugar veamos que δ^* define una semi-métrica. Por la igualdad anterior, como δ es simétrica (d2) y no negativa (d4) entonces también lo es δ^* . Por tanto, basta ver que se verifica (d1). Esto es:

$$\delta^*([\mathbf{X}], [\mathbf{Y}]) = 0 \Leftrightarrow [\mathbf{X}] = [\mathbf{Y}] \quad \text{para todo } [\mathbf{X}], [\mathbf{Y}] \in \mathcal{T}/\mathcal{R}$$

\Leftarrow Si $[\mathbf{X}] = [\mathbf{Y}]$ entonces existe una serie X^* en forma condensada que pertenece a ambas clases de equivalencia y $\delta^*([\mathbf{X}], [\mathbf{Y}]) = \delta(\mathbf{X}^*, \mathbf{X}^*) = 0$

\Rightarrow Si $\delta^*([\mathbf{X}], [\mathbf{Y}]) = 0$ entonces $\delta(\mathbf{X}^*, \mathbf{Y}^*) = 0$ donde \mathbf{X}^* e \mathbf{Y}^* son las formas condensadas respectivas de cada clase de equivalencia. Por tanto $\mathbf{Y}^* \in [\mathbf{X}^*]$ y por la unicidad dada en la *Proposición 2.57* se tiene que $\mathbf{X}^* = \mathbf{Y}^*$ luego $[\mathbf{X}] = [\mathbf{X}^*] = [\mathbf{Y}^*] = [\mathbf{Y}]$.

□

Se considera la extensión canónica $\delta^+ : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$ dada por

$$\delta^+(\mathbf{X}, \mathbf{Y}) = \delta^*([\mathbf{X}], [\mathbf{Y}]) = \delta(\mathbf{X}^*, \mathbf{Y}^*)$$

donde \mathbf{X}^* e \mathbf{Y}^* son las formas condensadas de \mathbf{X} e \mathbf{Y} respectivamente. Entonces, entendiendo que dos series son iguales si una es deformación de la otra, δ^+ es una semi-métrica que permanece invariante ante los efectos de deformación (extensiones o condensaciones). Es decir, si $\mathbf{X}, \mathbf{Y} \in \mathcal{T}$ son tales que $\mathbf{X} \gg \mathbf{Y}$ entonces para cada $\mathbf{Z} \in \mathcal{T}$ se cumple que $\delta^+(\mathbf{X}, \mathbf{Z}) = \delta^+(\mathbf{Y}, \mathbf{Z})$.

Capítulo 3

Fundamentos sobre *clustering*

El objetivo principal es clasificar diferentes series temporales para encontrar patrones comunes y posibles anomalías. En este capítulo se presenta el algoritmo de clasificación no supervisada (*clustering*) por excelencia, el algoritmo de k-medias, y se estudian las modificaciones necesarias para poder aplicar dicho algoritmo, que inicialmente se desarrolló para conjuntos de datos cuyos miembros tienen la misma dimensión, sobre un conjunto de series temporales donde la longitud (dimensión) de cada serie es finita pero variable entre sus miembros.

3.1. Algoritmos de clasificación no supervisada (*clustering*)

En esta sección se introducen las técnicas necesarias para encontrar patrones o “agrupamientos naturales” en un conjunto de datos de los cuales no se dispone de más información que el propio dato. Así pues, se trata de un caso de *aprendizaje no supervisado*, en el que se busca dividir un conjunto de datos en diferentes subconjuntos, tales que, los elementos de cada subconjunto compartan características, desconocidas a priori, y que cada subconjunto difiera al máximo posible del resto. Esta sección se basa en [13], donde se presenta la versión tradicional del algoritmo de k-medias entre otros.

El *clustering* es una técnica que consiste en agrupar un conjunto de objetos (datos) no etiquetados (es decir, se desconocen sus características fuera del propio dato) en subconjuntos denominados **clústeres**. Cada *clúster* está formado por una colección de objetos considerados similares entre sí y diferenciado del resto, de forma que la pertenencia a un *clúster* es excluyente.

A cada clúster se lo asigna un **centroide** (centro geométrico), también llamado *baricentro*, que actúa como representante de su clúster y viene determinado por una “aplicación” sobre los puntos que pertenecen a cada clúster.

Para poder realizar esta clasificación se requiere disponer de una medida de similitud (denominada **distancia**) que permita comparar los objetos. Por tanto, un algoritmo de *clustering* busca minimizar las distancias entre elementos del mismo clúster y maximizar las distancias

entre diferentes clústeres. La medida de similitud no tiene que ser necesariamente una distancia en el sentido de la definición formal (*Definición 2.47*), basta con que cumpla ciertas propiedades.

Definición 3.1. Sea \mathbb{V} un espacio vectorial sobre \mathbb{R} de dimensión finita $r \in \mathbb{N}$ y sea $d : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$ tal que para cada $x, y \in \mathbb{V}$

- (a) $d(x, y) \geq 0$
- (b) $d(x, y) = d(y, x)$

Dado $\mathcal{D} \subset \mathbb{V}$ un conjunto finito de cardinal $m \in \mathbb{N}$, $\mathcal{D} = \{x_1, \dots, x_m\}$, que representan los datos a clasificar, si se denotan por K_1, \dots, K_n las n agrupaciones.

1. Se dice que K_1, \dots, K_n forman un **sistema de clústeres** si la pertenencia a cada K_i es excluyente del resto y su unión es el conjunto total de datos. Esto es:

$$\mathcal{D} = \bigsqcup_{i=1}^n K_i \text{ (unión disjunta).}$$

2. A cada K_i se le denomina **clúster**.
3. Se define la **matriz de pertenencia** como $W = (\omega_{i,j})_{1 \leq i \leq m; 1 \leq j \leq n}$ tal que:

$$\omega_{i,j} = \begin{cases} 1 & \text{si } x_i \in K_j \\ 0 & \text{si } x_i \notin K_j \end{cases}$$

4. Sea $F : \mathbb{V} \times \mathbb{V} \times \dots \times \mathbb{V} \rightarrow \mathbb{V}$ la función que determina los centroides de cada clúster, entonces, para el clúster K_i se define su **centroide** denotado por c_i como:

$$c_i = F(\omega_{1,i} \cdot x_1, \omega_{2,i} \cdot x_2, \dots, \omega_{m,i} \cdot x_m)$$

5. Se define la **distancia intra-clúster** para el clúster $K_t = \{\alpha_1, \dots, \alpha_s\}$ como la suma de las distancias entre cada par de elementos del clúster. Esto es:

$$\sum_{j=1}^s \sum_{i=j+1}^s d(\alpha_j, \alpha_i) = \sum_{j=1}^m \sum_{i=j+1}^m \omega_{i,t} \cdot \omega_{j,t} \cdot d(x_j, x_i)$$

6. Se define la **distancia inter-clúster** entre K_i y K_j como:

$$\text{mín}\{d(x, y) : x \in K_i, y \in K_j\} = d(K_i, K_j)$$

7. Se define la **variación intra-clúster** de K_t , o simplemente **variación** de K_t , y se denota por $S^t(W)$, como la suma de las distancias entre cada elemento del clúster con su centroide. Esto es:

$$S^t(W) = \sum_{x \in K_t} d(x, c_t) = \sum_{i=1}^m \omega_{i,t} \cdot d(x_i, c_t)$$

8. Se define la **variación total intra-clúster**, y se denota por $\mathcal{S}(W)$ como la suma de las variaciones de cada clúster. Esto es:

$$\mathcal{S}(W) = \sum_{t=1}^n \mathcal{S}^t(W) = \sum_{t=1}^n \sum_{i=1}^m \omega_{i,t} \cdot d(x_i, c_t)$$

Generalmente, \mathbb{V} es un espacio métrico y d es una distancia. El objetivo del clustering es equivalente a encontrar la matriz de pertenencia $W^* = (\omega_{i,j}^*)_{1 \leq i \leq m; 1 \leq j \leq n}$ que haga mínimo la variación total intra-clúster $\mathcal{S}(W^*)$ entre todas las posibles matrices de pertenencia.

Proposición 3.2. Propiedades de la matriz de pertenencia. Sea $W = (\omega_{i,j})_{1 \leq i \leq m; 1 \leq j \leq n}$ la matriz de pertenencia, esta verifica que:

1) $\omega_{i,j} \in \{0, 1\} \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, n.$

2) $\sum_{j=1}^n \omega_{i,j} = 1, \quad \forall i = 1, \dots, m.$

3) $\sum_{i=1}^m \sum_{j=1}^n \omega_{i,j} = m.$

Además, sea $\mathcal{D} = \{x_1, \dots, x_m\}$ un conjunto de datos, toda matriz $A = (a_{i,j})_{1 \leq i \leq m; 1 \leq j \leq n}$ de dimensiones $m \times n$ que verifique las propiedades anteriores es matriz de pertenencia.

Demostración.

1. La propiedad 1 es consecuencia de cómo se definió la matriz W en la *Definición 3.1-1*.
2. La propiedad segunda se prueba razonando por reducción al absurdo. Supongamos que para cierto i se cumple $\sum_{j=1}^n \omega_{i,j} \neq 1$. Por la propiedad primera se tiene que

$$\sum_{j=1}^n \omega_{i,j} = 0 \quad \text{ó} \quad \sum_{j=1}^n \omega_{i,j} > 1$$

En el primer caso esto supone que $\omega_{i,j} = 0, \quad j = 1, \dots, n$, luego por definición, el elemento $x_i \notin K_j \quad \forall j = 1, \dots, n$ y se tiene que $\mathcal{D} \not\subseteq \bigsqcup_{i=1}^n K_i$, absurdo.

En el segundo caso, se verifica que existe un $i \in \{1, \dots, m\}$ para el cual existen $j_1, j_2 \in \{1, \dots, n\}$ tal que $j_1 \neq j_2$ y $\omega_{i,j_1} = \omega_{i,j_2} = 1$. Es decir, por la definición de matriz de pertenencia se cumple que $x_i \in K_{j_1}$ y $x_i \in K_{j_2}$, y en consecuencia K_{j_1} y K_{j_2} no son disjuntos. Absurdo.

Por tanto, la suposición inicial era incorrecta y se concluye que $\forall i = 1, \dots, m \quad \sum_{j=1}^n \omega_{i,j} = 1$

3. La propiedad tercera es consecuencia inmediata de la propiedad segunda, dado que:

$$\sum_{i=1}^m \sum_{j=1}^n \omega_{i,j} = \sum_{i=1}^m 1 = m$$

Por último, veamos el recíproco. Sean K_1, \dots, K_n las agrupaciones, veamos que es un sistema de clústeres. Por la propiedad primera, $a_{i,j} \in \{0, 1\}$, luego se puede considerar que:

$$x_i \in K_j \Leftrightarrow a_{i,j} = 1 \quad \text{y} \quad x_i \notin K_j \Leftrightarrow a_{i,j} = 0$$

La propiedad segunda garantiza que para cada i existe un j tal que $x_i \in K_j$ y para todo $p \neq j$, $x_i \notin K_p$. En consecuencia \mathcal{D} es unión disjunta de los clústeres y se tiene que K_1, \dots, K_n es un sistema de clústeres. \square

3.1.1. El algoritmo de k-medias

El algoritmo de *clustering* por excelencia es el **algoritmo de k-medias** (*k-means*), también denominado *algoritmo de Lloyd*, pues fue propuesto por primera vez por Stuart Lloyd en 1957.

En esta sección se considera \mathbb{V} un espacio métrico de dimensión finita $n \in \mathbb{N}$ dotado de la distancia d , sea $\mathcal{D} = \{x_1, \dots, x_m\} \subset \mathbb{V}$ un subconjunto finito de cardinal m (representa los datos a clasificar). Supongamos que se desea una división de los datos en $k \in \mathbb{N}$ clústeres. Se denotan por K_1, \dots, K_k cada uno de los k clústeres, y por c_1, \dots, c_k los centroides asignados respectivamente a cada clúster.

El algoritmo de k-medias asigna un dato $a \in \mathcal{D}$ al clúster K_i sí y solo sí:

$$d(a, c_i) \leq d(a, c_j) \quad \forall j = 1 \dots n \text{ con } i \neq j$$

Una vez asignados los datos a un único clúster, se recalculan los centroides como la media de los datos que pertenecen a dicho clúster. El procedimiento anterior es repetido hasta que se cumple cierto criterio de parada.

El algoritmo de k-medias requiere que se determine previamente el número de clústeres en los que separar el espacio. Para ello se suelen emplear técnicas heurísticas, pues la elección adecuada del número de clústeres puede afectar notablemente al resultado. También es necesario inicializar los centroides en la primera iteración del algoritmo, para lo que generalmente se recurre a una asignación aleatoria.

Definición 3.3. Con las notaciones anteriores, sea $W = (\omega_{i,j})_{1 \leq i \leq m; 1 \leq j \leq k}$ la matriz de pertenencia, en el algoritmo de k-medias:

1. Se dice que un dato $a \in \mathcal{D}$ se **asigna al clúster** K_i y se denota por $a \in K_i$ si se cumple:

$$d(a, c_i) = \min\{d(a, c_j) : j = 1, \dots, k\}$$

2. Se define el **centroide** $c_i = (c_i^1, \dots, c_i^n)$ asociado a clúster K_i como la media del conjunto de puntos agrupados en dicho clúster.

Algoritmo de k-medias (versión tradicional)

Sea k el número de clústeres predefinidos, sean K_1, \dots, K_k los clústeres y c_1, \dots, c_k los centroides asociados a cada clúster. Sea $\mathcal{D} = \{x_1, \dots, x_m\}$ el conjunto de datos.

1. Inicialización: se inicializa aleatoriamente cada centroide.
2. Asignación: se realiza la asignación de datos a cada grupo o clúster. Para cada $x \in \mathcal{D}$, de forma que x pertenecerá al clúster cuyo centroide minimice su distancia respecto al conjunto de centroides.

$$x \in K_i \Leftrightarrow i = \arg \min_{j=1..k} (d(x, c_j)) \quad (3.1)$$

3. Cálculo de centroides: se reajustan los centroides como la media de los datos de cada clúster.

$$c_i = \frac{1}{|K_i|} \sum_{x \in K_i} x \quad (3.2)$$

4. Evaluación: Se comprueba el criterio de parada, que se verifica cuando la asignación de clústeres permanece constante entre dos iteraciones consecutivas. Si no se cumple este criterio, se repiten los pasos 2, 3, y 4 con los nuevos centroides.

El algoritmo de k-medias es un problema de complejidad *np-hard* siempre que la dimensionalidad de los datos sea mayor o igual que dos, es decir, el coste computacional (y por tanto el tiempo necesario) crecen de forma polinómica respecto a un crecimiento lineal del número de datos ¹.

Como consecuencia de la *Proposición 3.2* se verifica que dado un conjunto de datos es equivalente conocer la asignación de clúster a conocer la matriz de pertenencia. En particular, en la demostración de dicha proposición se probó que dada una matriz que cumpliera las propiedades enunciadas para las matrices de pertenencia, a partir de esta matriz se podía reconstruir la asignación de clústeres. Así pues, el algoritmo anterior puede enunciarse en términos de la matriz de pertenencia.

La versión de la matriz de pertenencia presenta una ventaja de sencillez en su implementación, pues contempla conceptos más abstractos tales como “un dato pertenece a un clúster” como un valor concreto de un coeficiente de la matriz de pertenencia.

¹Consúltese [15]

Algoritmo de k-medias (versión de matriz de pertenencia)

Sean k el número de clústeres predefinidos, c_1, \dots, c_k los centroides asociados a cada clúster y $\mathcal{D} = \{x_1, \dots, x_m\}$ el conjunto de datos a clasificar.

1. Inicialización: se inicializa aleatoriamente cada centroide.
2. Asignación: sea $W^p = (\omega_{i,j}^p)_{1 \leq i \leq m; 1 \leq j \leq k}$ la matriz de pertenencia en la p -ésima iteración. Se inicializa $W^p = \mathbf{0}$ (matriz nula) y para cada $i \in \{1, \dots, m\}$ se toma

$$j = \arg \min_{t=1, \dots, k} \{d(x_i, c_t)\} \quad (3.3)$$

Y se actualiza el coeficiente correspondiente de la matriz de pertenencia

$$\omega_{i,j}^p = 1 \quad (3.4)$$

3. Cálculo de centroides: se reajustan los centroides como la media de los datos de cada clúster. Con la matriz de pertenencia, si $c_i = (c_i^1, \dots, c_i^n)$ y $x_j = (x_j^1, \dots, x_j^n)$ entonces:

$$c_i^t = \frac{\sum_{j=1}^m \omega_{j,i} \cdot x_j^t}{\sum_{j=1}^m \omega_{j,i}} \quad t = 1 \dots n \quad (3.5)$$

4. Evaluación: Se comprueba el criterio de parada.
 - a) Matriz de pertenencia permanece constante en una iteración, $W^p = W^{p-1}$.
 - b) La variación total intra-clúster alcanza un valor de tolerancia mínimo. Es decir, fijado $T \in \mathbb{R}$ tolerancia, se finaliza si $\mathcal{S}(W^p) < T$

Si no se cumple el criterio de parada entonces se repiten los pasos 2, 3 y 4

La implementación en pseudocódigo se recoge en el *Apéndice B.4* y se realiza para la versión de la matriz de pertenencia del algoritmo.

Ejemplo 3.4. Uno de los conjuntos de datos más conocidos en el contexto de la búsqueda de patrones es la base de datos de los diferentes tipos de flor de iris. En ella se recogen cuatro atributos de tres tipos de flores (iris-Setosa, iris-Versicolor e iris-Virginica) de los cuales se van a considerar solamente dos, que son la longitud del pétalo y el ancho del pétalo. La siguiente figura (3.1) ilustra los datos y el resultado de aplicar el algoritmo de k-medias con 3 clústeres.

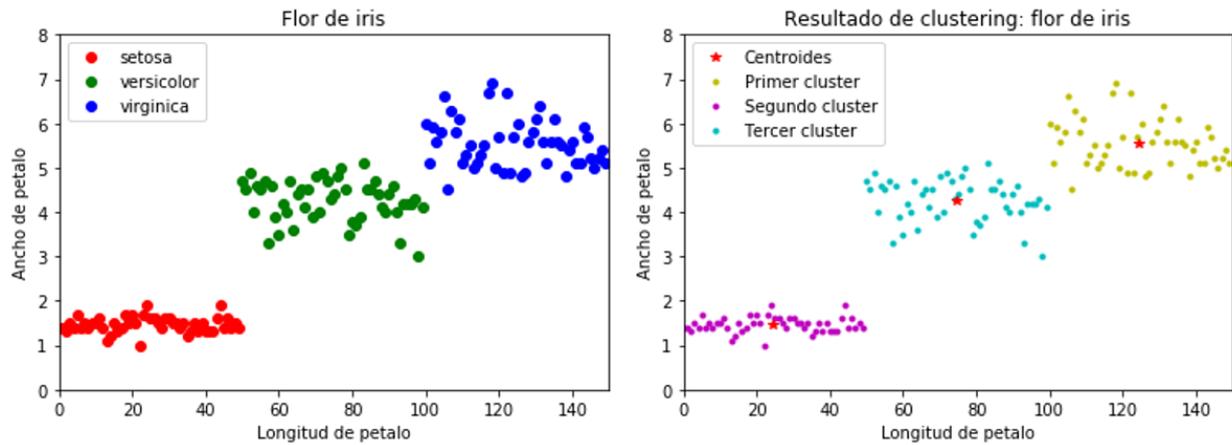


Figura 3.1: Ejemplo de clustering con la flor de iris: representación de los datos (izquierda) y resultado del algoritmo (derecha)

3.2. El algoritmo de k-medias aplicado a series temporales

El algoritmo de k-medias como se presenta en la *sección 3.1.1* es aplicable a conjuntos de series temporales de la misma longitud. Sin embargo, en el estudio de las series temporales es poco usual que estas presenten la misma longitud. En esta sección se van a tratar las modificaciones a realizar sobre el algoritmo de k-medias expuesto anteriormente para cubrir el conjunto de las series temporales sin importar su dimensión. Esta sección se basa en [11] y [12] y [15].

3.2.1. La media de Fréchet

Dado que se comparan series temporales de diferente longitud, la media de las series como media aritmética de cada componente carece de sentido. En esta sección se busca definir una media (en el sentido de una medida de tendencia central perteneciente al conjunto de las series temporales) para lo que se recurre a la *media de Fréchet*.

Manteniendo la notación, \mathcal{T} denota al conjunto de las series temporales de longitud finita y variable menor que n y d_{DTW} denota la distancia DTW (en cualquiera de sus versiones).

Definición 3.5. Sea $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_m\} \subset \mathcal{T}$.

1. Se define la **función de Fréchet** como la aplicación $F: \mathcal{T} \rightarrow \mathbb{R}$ tal que para cada $\mathbf{Z} \in \mathcal{T}$

$$F_{\mathcal{D}}(\mathbf{Z}) = \sum_{i=1}^m d_{DTW}(\mathbf{Z}, \mathbf{X}_i)$$

2. Se define la **media de Fréchet** del conjunto de datos \mathcal{D} como la serie temporal μ que hace mínimo el valor de la función de Fréchet². Esto es:

$$\mu = \arg \min_{M \in \mathcal{T}} F_{\mathcal{D}}(M) \quad (3.6)$$

La siguiente figura ilustra (fig. 3.2) un ejemplo de aplicación de media de Fréchet para promediar un conjunto de series temporales. Para ello se ha utilizado el algoritmo DBA que se presenta posteriormente.

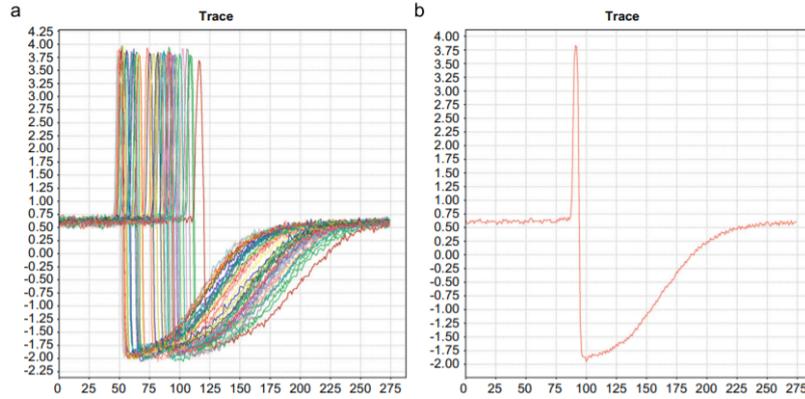


Figura 3.2: Representación de las series temporales a promediar (a) y del promedio obtenido mediante el algoritmo DBA (b).

Ejemplo 3.6. En dimensión 1, la media de Fréchet de un conjunto de datos (unidimensional) se corresponde con la mediana.

El algoritmo que permite resolver el problema definido por (3.6) bajo la distancia DTW se denomina *DTW barycenter averaging* (DBA). Dicho algoritmo utiliza las asociaciones de índices dadas por la distancia DTW para calcular la media simple componente a componente.

²La existencia de la media de Fréchet está garantizada en \mathcal{T} como un mínimo. Los detalles de esta afirmación se pueden consultar en [12].

Algoritmo DBA

Sea $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_m\}$ el conjunto con las secuencias a promediar, con n_i la longitud de la serie $\mathbf{X}_i = [x_1^i, \dots, x_{n_i}^i]$.

1. Se supone que existe una serie temporal $\mathbf{C} = (c_1, \dots, c_t)$ sobre la que se realiza la comparación.
2. Se toma T como la longitud de \mathbf{C} .
3. Sea $\gamma_i(s)$ el conjunto de los índices de la serie \mathbf{X}_i asociados al índice s de la serie \mathbf{C} por la distancia DTW. Sea $\mathbf{C}' = [c'_1, \dots, c'_t]$ la media de Fréchet, entonces dado $s \in \{1, \dots, t\}$ se define

$$c'_s = \frac{1}{\sum_{i=1}^n |\gamma_i(s)|} \left(\sum_{i=1}^n \sum_{k \in \gamma_i(s)} x_k^i \right) \quad (3.7)$$

Donde $|\gamma_i(s)|$ denota el número de elementos del conjunto $\gamma_i(s)$

En consecuencia, el algoritmo DBA define una “media” entre series temporales respecto de \mathbf{C} como la serie temporal cuya componente s –ésima es dada por la media de las componentes de las series temporales a promediar cuyos índices se asocian con s en la deformación *DTW*.

Dado un conjunto de series temporales, su media de Fréchet (sin especificar respecto a que serie) se halla generando aleatoriamente series de comparación y repitiendo el procedimiento anterior se selecciona la media de menor distancia DTW. En la generación aleatoria caben estudiar dos atributos de cada serie de comparación:

- El tamaño de la serie
- La generación de las componentes

En cuanto a la primera de ellas, se suele tomar el máximo de las longitudes de las series temporales que se quieren promediar. Respecto a los valores de las componentes se recurre a técnicas heurísticas.

Ejemplo 3.7. Se consideran las series temporales dadas por $\mathbf{Y} = [4, 4, 7, 4, 1]$, $\mathbf{Z} = [5, 8, 8, 5, 5, 2]$ y Dada $\mathbf{X} = [3, 6, 3, 0]$. Sean Φ y Ψ las curvas de deformación de las comparaciones de \mathbf{X} frente a \mathbf{Y} y \mathbf{Z} respectivamente y sean $\gamma_Y(s)$ y $\gamma_Z(s)$ el conjunto de índices de las series \mathbf{Y} y \mathbf{Z} asociados al índice s de la serie \mathbf{X} . Entonces:

$$\begin{array}{llllll} \Phi(1) = (1, 1) & \Phi(2) = (1, 2) & \Phi(3) = (2, 3) & \Phi(4) = (3, 4) & \Phi(5) = (4, 5) & \\ \Psi(1) = (1, 1) & \Psi(2) = (2, 2) & \Psi(3) = (2, 3) & \Psi(4) = (2, 4) & \Psi(5) = (3, 5) & \Psi(6) = (4, 6) \\ \gamma_Y(1) = \{1, 2\} & \gamma_Y(2) = \{3\} & \gamma_Y(3) = \{4\} & \gamma_Y(4) = \{5\} & & \\ \gamma_Z(1) = \{1\} & \gamma_Z(2) = \{2, 3, 4\} & \gamma_Z(3) = \{5\} & \gamma_Z(4) = \{6\} & & \end{array}$$

La media de Fréchet para las series \mathbf{Y} , \mathbf{Z} frente a \mathbf{X} es \mathbf{C} dada por:

$$c_1 = \frac{y_1 + y_2 + z_1}{3} = \frac{13}{3} \quad c_2 = \frac{y_3 + z_2 + z_3 + z_4}{4} = 7$$

$$c_3 = \frac{y_4 + z_5}{2} = 4,5 \quad c_4 = \frac{y_5 + z_6}{2} = 1,5$$

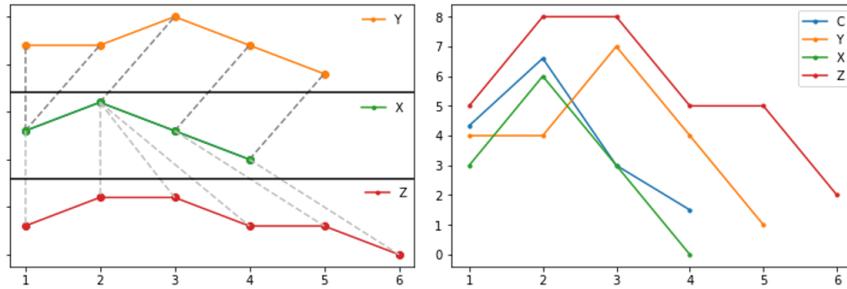


Figura 3.3: Representación de las series temporales \mathbf{X} , \mathbf{Y} y \mathbf{Z} con sus asociaciones de índices respecto de \mathbf{X} (izquierda) y junto a su media \mathbf{C} (derecha)

El algoritmo DBA como solución a la media de Fréchet.

El algoritmo DBA como se ha presentado no resuelve exactamente el problema (3.6), sino que resuelve una variante en la que las series candidatas a promedio tienen longitud fija. Es decir, sea $\mathcal{T}_T \subset \mathcal{T}$ el conjunto de las series temporales de longitud exactamente T , el algoritmo DBA encuentra la solución al problema

$$\mu = \arg \min_{\mathbf{M} \in \mathcal{T}_T} F_{\mathcal{D}}(\mathbf{M}) \quad (3.8)$$

Sin embargo, el algoritmo dado es perfecto para el clustering de series temporales al cumplir las condiciones específicas.

3.2.2. Construcción del algoritmo de k-medias aplicable a series temporales

En esta sección, se recogen las modificaciones pertinentes sobre el algoritmo de k-medias general de la sección 3.1.1 para ser aplicable a series temporales de diferente longitud con misma escala temporal (es decir, el conjunto de índices de mayor cardinal es finito y contiene al resto).

Como se permiten series de diferente longitud, la medida de comparación a usar es la distancia DTW (en cualquiera de sus versiones) y los centroides se calculan mediante el algoritmo DBA.

Algoritmo de k-medias (versión aplicable a series temporales)

Sean k el número de clústeres predefinidos, $\mathbf{C}_1, \dots, \mathbf{C}_k$ los centroides asociados a cada clúster K_1, \dots, K_k y $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_m\}$ el conjunto de datos a clasificar. Sea n_i la longitud de la serie $\mathbf{X}_i = [x_1^i, \dots, x_{n_i}^i]$

1. Inicialización: Se inicializa aleatoriamente cada centroide como series temporales de longitud T . Generalmente, se toma $T = \max\{n_1, \dots, n_m\}$
2. Asignación: sea $W^p = (\omega_{i,j}^p)_{1 \leq i \leq m; 1 \leq j \leq k}$ la matriz de pertenencia en la p -ésima iteración. Se inicializa $W^p = \mathbf{0}$ (matriz nula) y para cada $i \in \{1, \dots, m\}$ se toma

$$j = \arg \min_{t=1..k} \{d_{DTW}(x_i, c_t)\} \quad (3.9)$$

Y se actualiza el coeficiente correspondiente de la matriz de pertenencia

$$\omega_{i,j}^p = 1 \quad (3.10)$$

3. Cálculo de la función de identificación: realizada la asignación de X_i al cluster j -ésimo, se calcula γ_i función de identificación de índices de \mathbf{C}_j respecto de \mathbf{X}_i .
4. Cálculo de centroides: se reajustan los centroides Se define \mathbf{C}_j como la media de Fréchet (*algoritmo DBA*) sobre el conjunto de las series temporales asociadas al clúster j

$$\{X_i : i = 1..m, \omega_{i,j}^p = 1\}$$

5. Evaluación: Se comprueba el criterio de parada.
 - a) La matriz de pertenencia permanece constante en una iteración. Es decir, $W^p = W^{p-1}$.
 - b) La variación total intra-clúster, definida como $\sum_{t=1}^n \sum_{i=1}^m \omega_{i,t} \cdot d_{DTW}(\mathbf{X}_i, \mathbf{C}_t)$ alcanza un valor de tolerancia mínimo.

Si no se cumple el criterio de parada entonces se repiten los pasos 2, 3, 4 y 5

Para su implementación, dado el elevado coste computacional de la distancia DTW (cuya complejidad algorítmica es de $\mathcal{O}(n^2)$), se debe minimizar el número de veces que se calcula dicha distancia. Por ello, en la implementación presentada en el *Apéndice B.5*, se “guarda” el valor de la distancia de cada dato a su centroide (en el vector *distancias*) necesaria para el cálculo de la variación. Análogamente, se procede al cálculo de las asociaciones de índices de cada centroide respecto de cada dato asignado a su mismo clúster (implementado en la lista *gammas*) aprovechando el momento de cálculo de la matriz DTW correspondiente, y que resultan necesarias para el cálculo de los centroides según el algoritmo DBA. Por último, cabe destacar que los centroides deben ser de la misma longitud y que esta se mantiene en cada iteración.

3.3. Principales problemas asociados al clustering de k-medias

En esta sección se presentan los principales problemas y dificultades asociadas al algoritmo de clasificación por k-medias de series temporales. Estas dificultades pueden ser consecuencia del propio algoritmo y/o de su aplicación sobre series temporales.

En el contexto de análisis de datos se vienen realizando numerosos estudios para aplicar la clasificación de clústeres de forma efectiva y eficiente en grandes conjuntos de información. En la actualidad, las líneas de investigación se centran en la escalabilidad, en la efectividad para agrupar formas y datos complejos y en los métodos dirigidos a gestionar datos mixtos o no numéricos.

1. **Escalabilidad.** El algoritmo de k-medias en general funciona bien (respecto al tiempo) sobre conjunto de datos pequeños, pero al tratarse de un problema *np-hard*, el tiempo requerido cuando se aumenta el conjunto de datos crece enormemente. Es más, aplicado a series temporales con la distancia DTW, supuestos k centroides de longitud T y m series temporales de longitud máxima n , para tan solo una única iteración del algoritmo se tiene una complejidad de $\mathcal{O}(h \cdot T \cdot n \cdot m)$. Esto se debe a que para cada serie temporal se calcula su distancia DTW a cada centroide cuya complejidad es $\mathcal{O}(T \cdot n)$. Además, se requieren varias iteraciones hasta llegar a la condición de parada dada.
2. **Capacidad para tratar con diferentes tipos de datos.** El trabajar con series temporales de diferente longitud implica un problema, pues a priori carecen de sentido operaciones matemáticas básicas como la suma y el producto. Esta misma dificultad se puede extender a otro tipo de datos no numéricos en la versión genérica del algoritmo de clustering.
3. **Descubrir clústeres con formas arbitrarias.** Dependiendo de la distancia usada se predefine la forma de los clústeres. Por ejemplo, en el plano real, la distancia euclídea fuerza a que los clústeres tengan forma de bolas mientras que la distancia del supremo fuerza a que tengan forma de cuadrados. Un caso similar se produce entre las diferentes distancias p -DTW al restringirse a espacios de dimensión n y series de dicha longitud. En la *Figura 3.4* se puede observar geoméricamente como cualquier bola que se considere es incapaz de separar ambos conjuntos de datos.
4. **Parámetros iniciales predefinidos.** El algoritmo de k-medias requiere que se determinen previamente el número de clústeres y los centroides iniciales para la primera iteración. En particular, una mala elección de los centroides iniciales puede suponer un importante error en la clasificación obtenida. En la siguiente figura (*3.5*) se puede observar los efectos sobre el ejemplo de la flor de iris (véase *ej: 3.4*) de elegir erróneamente tanto el número de clústeres como los centroides iniciales. En el primer caso se observa como la clasificación obtenida difiere de la real al agrupar dos tipos de flores distintos en un mismo clúster, pues solo se permitieron dos

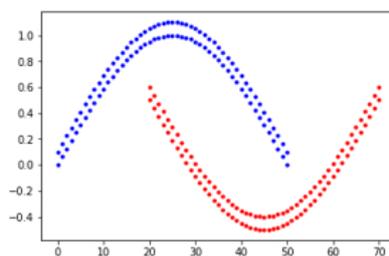


Figura 3.4: Ejemplo de agrupaciones complejas, cuya solución falla por k-medias.

agrupaciones diferentes. En el caso de inicializar erróneamente los centroides esto ha provocado que se divida en dos el grupo de iris-setosa y se agrupen juntas las variedades iris-versicolor e iris-virginica.

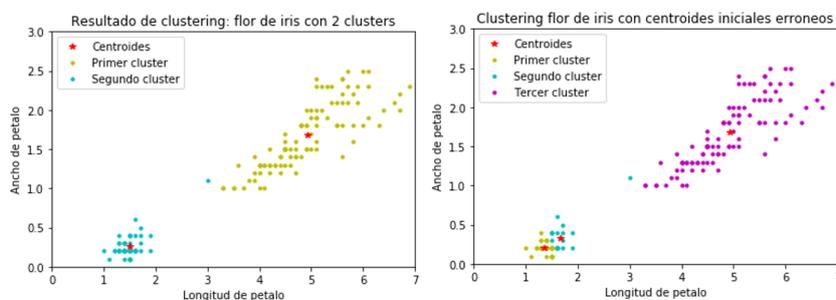


Figura 3.5: Ejemplo de clustering incorrectos con la flor de iris: agrupación con dos clústeres (izquierda) y agrupación con los clústeres iniciales erróneos (derecha)

5. **Incapacidad frente a datos ruidosos.** Al igual que la media de un conjunto de datos se ve especialmente afectada por los valores anómalos o *outliers*. El algoritmo de k-medias sufre el mismo problema, pues los *outliers* alejan al centroide del “centro real” del conjunto. Para solventar este inconveniente existe una versión denominada *k-medias recortadas* (*trimmed k-means*). Para más información consúltese [20].
6. **Alta dimensionalidad.** En un conjunto de datos con gran cantidad de atributos se puede sobrecargar el modelo. En primer lugar, puede haber atributos que no sean apropiados para la clasificación (no informativos) y en segundo lugar exigir el uso de demasiados atributos pueden hacer la clasificación demasiado exigente y dar peores resultados además de tener un coste superior.

3.4. Los diferentes enfoques en la clasificación de series temporales

En esta sección se desarrollan los principales enfoques para resolver el problema de clasificar un conjunto de series temporales. Está basado en las ideas desarrolladas en [15].

En particular, existen tres enfoques que difieren en la representación dadas a las series temporales previa a su clasificación.

- **Conversión discreta.** Para cada serie temporal se extraen un conjunto de parámetros o características discretas (en forma de vector) y se aplica el algoritmo de clasificación (k-medias tradicional) sobre los parámetros. Si se conoce el modelo de las series temporales, estas características pueden tomarse como los parámetros de dicho modelo, y en caso contrario se recurre a caracterizaciones estadísticas del conjunto de observaciones temporales que componen la serie.
- **Clasificación basada en la forma.** En este caso se trabaja con las series temporales “en bruto”, aplicando el algoritmo de k-medias bajo la distancia DTW como se presenta en las secciones anteriores.
Una variante de este enfoque consiste en aplica algún filtro o transformación a las series temporales antes de realizar la clasificación. Las transformaciones más usuales son la eliminación de los efectos estacionales (generalmente sin eliminar las variaciones cíclicas cuando existan) o la segmentación según el modelo.
- **Enfoque híbrido.** Combina los dos enfoques anteriores, realizando en primer lugar una clasificación atendiendo a características de las series temporales y en segundo lugar aplica un algoritmo de clasificación sobre series temporales. Sea \mathcal{D} el conjunto de las series temporales dato a clasificar, se distinguen dos fases:
 - 1) Se realiza una descomposición discreta de cada serie temporal, y se extraen un conjunto de parámetros a clasificar. Se realiza la clasificación de dichos parámetros y con ella se divide el conjunto \mathcal{D} en diferentes subconjuntos disjuntos, que denotamos por D_1, \dots, D_n .
 - 2) Para cada uno de los subconjuntos D_i se aplica un algoritmo de clasificación sobre series temporales basada en la forma. Con ello se tiene una segunda clasificación de las series temporales.

La ventaja principal del enfoque híbrido es que reduce el conjunto total de datos sobre el que se aplica el algoritmo, aunque este se aplique un mayor número de veces. Dado que el algoritmo de k-medias bajo la distancia DTW es un problema *np-hard* y tan solo el cálculo de la distancia DTW sobre un conjunto de varias series temporales a los centroides tiene complejidad cuadrática, el reducir el número de series dato supone una mejora notable en el consumo de recursos.

Capítulo 4

Caso de estudio: Sistema de detección de anomalías de la máquina fresadora CNC

En este capítulo se presenta una aplicación práctica de las técnicas y conceptos expuestos en los *Capítulos 2 y 3* aplicado sobre un entorno industrial real. En particular, se va a analizar la información recolectada de la máquina fresadora CNC, que se describió en la *Sección 1.1*. Los detalles de la implementación del modelo y del sistema completo de detección de anomalías en *Python*, fundamentados en el presente capítulo, se pueden consultar en [21].

En primer lugar, se elige la forma de representar la información y se realiza un análisis de los datos extraídos. Como resultado, se diseña la arquitectura del modelo de clasificación, teniendo en cuenta las peculiaridades de la información y de las técnicas empleadas. Posteriormente, se ajustan los parámetros del modelo y se analizan los resultados obtenidos.

4.1. Análisis de la información

En esta sección se desarrollan los detalles sobre la representación de la información y se realiza el primer análisis, gracias al cual se podrá definir una arquitectura para el modelo. El estudio se va a realizar sobre la información recolectada entre el 1 mayo de 2019 y el 30 de mayo de 2020 ¹.

Los sensores no intrusivos realizan mediciones cada segundo. Además, cada medición es etiquetada con la fecha exacta de realización, el tipo de operación que se realizaba, el material sobre el que se trabajaba y la herramienta utilizada. En particular, se va a trabajar con la información recogida por el sensor de fase de corriente 1 (denominada *corriente de fase* en esta memoria).

¹Debido a la situación producida por el Covid-19, la máquina fresadora ha sido destinada a otro tipo de producciones, y a partir de marzo del 2020 solo se han registrado la información relativa a los arranques.

Así, se va a representar cada operación realizada mediante una serie temporal según las mediciones del sensor de fase de corriente 1 y se van a diferenciar por el tipo de operación realizada y por el material utilizado. La *Figura 4.1* muestra un ejemplo de representación de una realización de la operación de contorneado sobre acero.

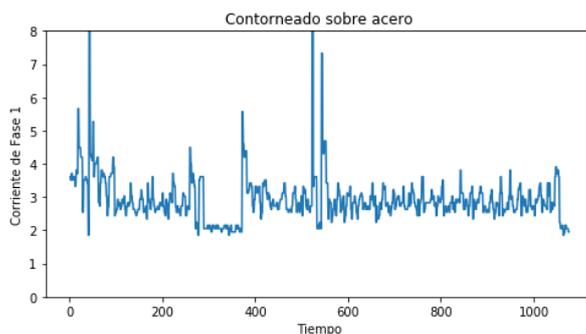


Figura 4.1: Representación de una operación de contorneado sobre el acero

Para diferenciar cada tipo de operación con cada tipo de material, se define un identificador único denominado *opermat*. Utilizando los identificadores numéricos de las *Tablas 1.1* y *1.2* para el tipo de operación y material, este identificador se define como diez veces el número asociado al tipo de operación más el número asociado al material. Por ejemplo, la operación anterior, de contorneado (operación 3) sobre acero (material 3), se le asocia el valor *opermat* de 33. Este identificador es el que se utiliza en las siguientes figuras.

El primer aspecto relevante es el número de ocurrencias de cada tipo de operación con cada tipo de material en el periodo de tiempo estudiado, ilustrado en el siguiente diagrama de barras (*Figura 4.2*).

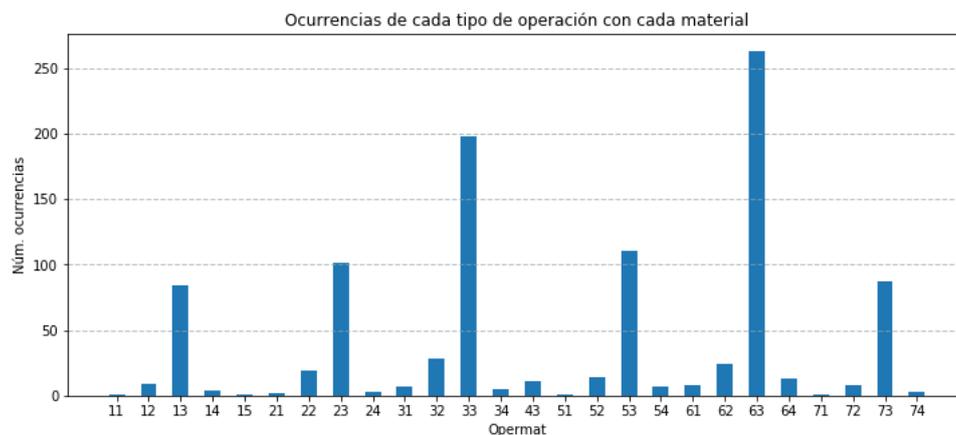


Figura 4.2: Ocurrencias de cada tipo de operación con cada material

Por tanto, el material más utilizado es el acero (código 3), que destaca notablemente para cada tipo de operación. Así, el estudio se centrará en cada tipo de operación realizada sobre acero. De este conjunto de operaciones, se va a analizar su duración con objeto de acotar el conjunto de operaciones válidas para la realización del modelo. En el *Apéndice C* se recoge la representación de ciertas actividades de los tipos de interés elegidas específicamente para ilustrar su comportamiento y su variedad. La siguiente figura (4.3) recoge las duraciones agrupadas.

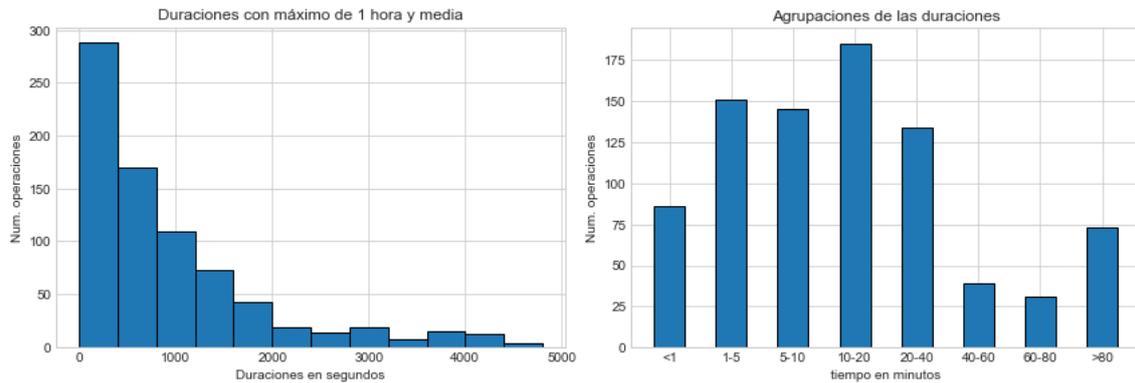


Figura 4.3: Histograma de las duraciones de las operaciones sobre el acero con duración inferior a una hora y 30 minutos (izquierda) y diagrama de barras con las duraciones agrupadas (derecha)

Las operaciones de poca duración (inferiores a cinco minutos) son asociadas al “repasso” o complemento de una pieza², y por lo tanto van a ser omitidas respecto al conjunto de construcción del modelo. Por otro lado, las operaciones de mucha duración suelen ser debidas a piezas peculiares, acciones de revisado sobre la máquina fresadora y a errores humanos en el etiquetado², y, en consecuencia, toda duración de más de una hora puede ser omitida con respecto al conjunto de construcción del modelo.

Imponiendo estas restricciones, el número de operaciones válidas para el modelo se recogen en la *Figura 4.4*. En esta figura, se puede observar que las restricciones sobre la duración pueden ser demasiado exigentes para las operaciones identificadas por los códigos 13 y 73 de *opermat*. Sin embargo, para no perder generalidad ni entrar al detalle en cada operación se mantienen dichas restricciones.

El segundo tipo de actividad que se realiza sobre la máquina y se desea estudiar son las acciones de arranque. El arranque comienza cuando el consumo de la máquina deja de tomar valores nulos y finaliza cuando el operario retira la seta de seguridad y arranca el PLC³ de la máquina.

²Información verificada por el operario de la máquina

³acrónimo del inglés Programmable Logic Controller

De forma análoga al caso de las operaciones (véase *Figura 4.5*), los arranques con una duración de menos de tres minutos suelen estar asociados a actividades de mantenimiento u otro tipo de necesidades puntuales, y por tanto se pueden omitir del modelo.

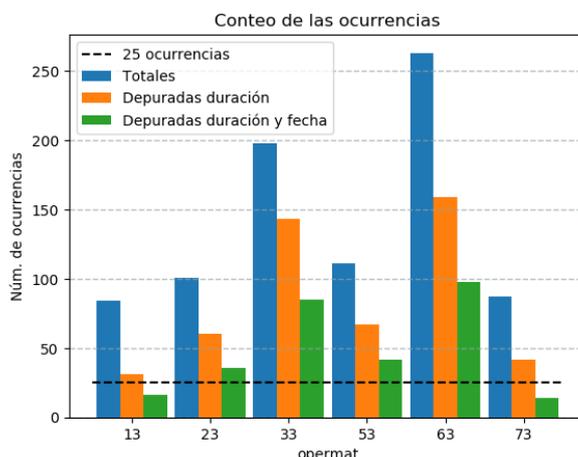


Figura 4.4: Diagrama de barras del número de operaciones resultantes: en azul sin imponer ningún tipo de restricción, en naranja tras exigir que la duración sea mayor de 5 minutos y menor de una hora y en verde tras exigir también que sean posteriores al 1 de o de 2019

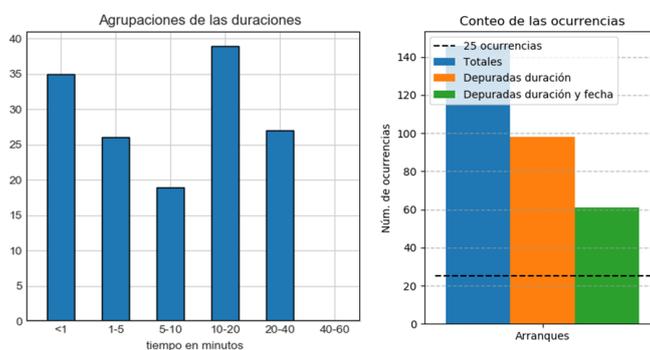


Figura 4.5: Agrupaciones de los arranques según su duración (izquierda) y diagrama de barras del número de operaciones resultantes (derecha): en azul sin imponer ningún tipo de restricción, en naranja tras exigir que la duración sea mayor de 3 minutos y menor de una hora y en verde tras exigir también que sean posteriores al 1 de septiembre de 2019

Además, como se puede observar en la *Figura 4.1*, la corriente de fase tiene un importante comportamiento estacionario que puede alterar los resultados y aumentar el coste computacional. Bajo las hipótesis del modelo aditivo, se descompone la serie y se elimina la estacionalidad aplicando el algoritmo de medias móviles en su versión centrada y con un tamaño de ventana pequeño (fijado experimentalmente en 10). En lo que resta, se supone que este tratamiento ha sido aplicado sobre las series de estudio.

4.2. Arquitectura del Modelo

En esta sección se describen los detalles sobre la elección y construcción de la técnica de clasificación.

Como resultado del estudio inicial (*Sección 4.1*) se tiene que las operaciones realizadas, siendo del mismo tipo y sobre el mismo material, pueden ser muy heterogéneas en cuanto a la duración se refiere. Esto se debe a que dependen de ciertas características (grosor, superficie, forma, etc) de la pieza del material con el que se trabaja (por ejemplo, las piezas de acero presentan diferentes grosores y dimensiones).

El uso de la distancia DTW reduce el efecto que tienen las características del material sobre la duración de la operación en la comparación de varias series. Sin embargo, presenta dos problemas asociados: el coste computacional crece como polinómicamente con orden 5 con respecto al número de observaciones y la distancia DTW puede producir deformaciones temporales excesivas (la duración excesiva también es considerada una anomalía).

En consecuencia, se opta por un modelo de clasificación híbrido (véase *Sección 3.4*). Este modelo realiza una doble clasificación:

1. En primer lugar, clasifica el conjunto total de las operaciones, representada cada una por la correspondiente serie temporal, \mathcal{D} atendiendo exclusivamente a su duración mediante el algoritmo de k-medias tradicional. Así, \mathcal{D} se descompone en varios subconjuntos disjuntos $\mathcal{D} = \mathcal{D}_1 \sqcup \dots \sqcup \mathcal{D}_p$.
2. Para cada uno de los subconjuntos $\mathcal{D}_1, \dots, \mathcal{D}_p$ se aplica el algoritmo de k-medias bajo la distancia DTW, clasificando así las series temporales de cada subconjunto, y descomponiendo $\mathcal{D}_i = K_{i,1} \sqcup \dots \sqcup K_{i,s}$. A los subconjuntos $K_{i,j}$ les denominaremos clústeres finales.

En consecuencia, se tiene que \mathcal{D} se divide como $\mathcal{D} = \bigsqcup_{i=1}^p \bigsqcup_{j=1}^s K_{i,j}$. La *Figura 4.6* resume de manera gráfica el procedimiento anterior.

Por tanto, el modelo está caracterizado por dos parámetros, el número de clústeres en los que se divide atendiendo a la duración, denotado por p , y el número de clústeres en los que se divide analizando las series temporales bajo la DTW, denotado por s . Estos parámetros son determinados experimentalmente en la siguiente sección (4.3).

Al igual que para la construcción del modelo, el proceso de evaluación de una serie temporal de muestra consta de dos fases: en primer lugar, se le asigna el clúster por su duración y en segundo lugar se le asigna un clúster por la propia serie (DTW). En ambos casos, el proceso de asignación se basa en determinar el centroide de distancia mínima a la serie muestra.

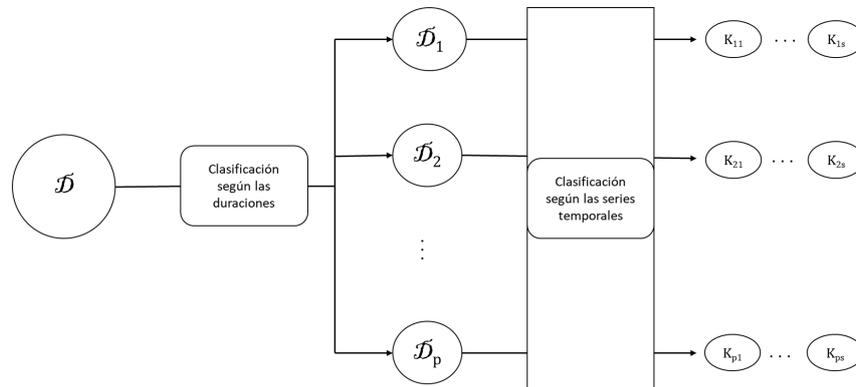


Figura 4.6: Esquema del modelo de clasificación híbrido planteado

Finalmente, una operación será considerada usual cuando el cociente del número de elementos del clúster final entre el número total de observaciones con las que se construyó el modelo sea superior a un cierto valor de tolerancia α , y será considerada anómala en caso contrario. Por ejemplo, si el modelo se construye con 90 observaciones y la muestra ha sido asignada a un clúster con 3 elementos con $\alpha = 0,1$ entonces será considerada anómala. En consecuencia, aparece un tercer parámetro a determinar que es el valor de tolerancia, es decir, suponiendo que todas las operaciones asignadas a un mismo clúster siguen un mismo modelo, α es el número de ocurrencias mínimo (en tanto por uno respecto del total) que tiene que repetirse un modelo de operación para ser considerado usual.

Como cada tipo de operación sobre cada tipo de material es esencialmente diferente al resto, entonces un modelo de clasificación híbrido como el descrito anteriormente se debe construir para cada par de tipos de operación y material. Considerando el número de ocurrencias registradas en el periodo de tiempo estudiado, estos modelos se van a construir para las operaciones de Planteado, Contorneado, Fresado y Taladrado sobre el acero (*opermat* de 23, 33, 53, 63 respectivamente) y para los arranques de la máquina (identificados por *opermat* de -10).

4.3. Estudio experimental de los parámetros

En esta sección se van a determinar los parámetros del modelo de clasificación descrito. Estos parámetros son el número de clústeres en los que se divide según la duración, que se denota por p y el número de clústeres en los que se divide según las propias series temporales bajo la distancia DTW, que se denota por s . Esto es lo que se denomina la *configuración para la clasificación* y se representa por (p, s) . Por otro lado, se debe determinar el porcentaje mínimo de ocurrencias asignadas a un clúster final, respecto del total de ocurrencias del mismo tipo, para que este sea considerado usual (denotado por α). Los tres parámetros forman la *configuración del modelo*.

4.3.1. La configuración para la clasificación

En esta sección se busca determinar la mejor configuración para los parámetros p y s . Existen diferentes técnicas, y en este caso se va a aplicar el *método del codo* sobre la distancia intra-clúster.

Sobre el algoritmo de k-medias tradicional, el método del codo supone que existe una función objetivo g real que traduce algunas variables o características (preferiblemente dependientes del número de clústeres considerados) de la clasificación en un valor real. Entonces, se representa la función objetivo de forma continua y se toma como mejor configuración el valor de número de clústeres en el que la tendencia de g cambia de un decrecimiento brusco a un decrecimiento moderado. La *Figura 4.7* ilustra un ejemplo de aplicación del método del codo sobre una función objetivo.

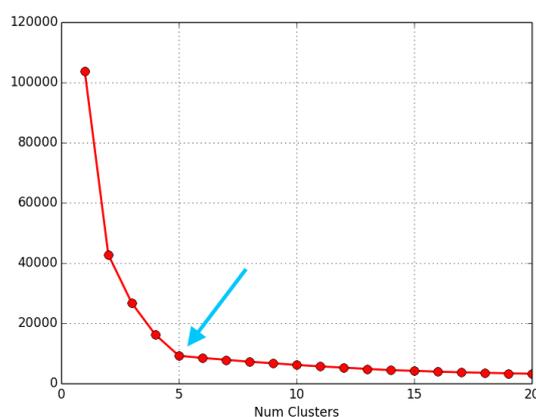


Figura 4.7: Ejemplo de aplicación del método del codo. La solución es de 5 clústeres.

En el caso de estudio, se va a considerar como función objetivo la distancia intra-clúster bajo la DTW. En esta sección se desarrolla el estudio detallado para la operación de contorneado sobre el acero (*opermat 33*), y en el *Apéndice D* se recogen los resultados para el resto de las operaciones analizadas.

En primer lugar, considerando la distancia intra-clúster como función objetivo, analizamos una aproximación del número de clústeres para las duraciones. El resultado se puede observar en la *Figura 4.8*.

Si solo se deseara clasificar las duraciones, el método del codo marca como mejor opción un número de 3 ó 4 clústeres. Sin embargo, los dos parámetros, p y s , que componen el modelo de clasificación están relacionadas, y el análisis anterior simplemente permite acotar las posibles configuraciones. Por tanto, de aplicar el método del codo en la *Figura 4.8* se deduce que para p basta considerar los valores de 2, 3 y 4.

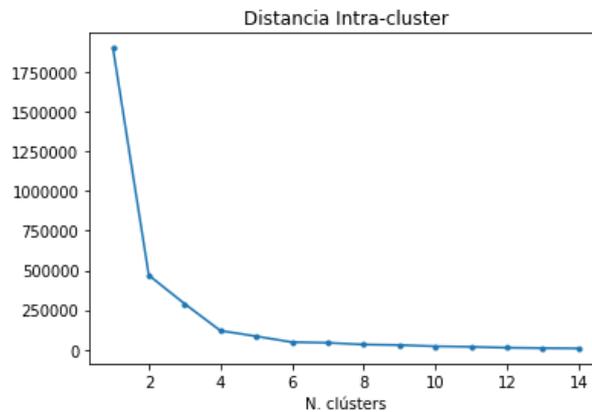


Figura 4.8: Representación de la distancia intra-clúster para diferentes configuraciones de número de clústeres para las duraciones, para el tipo de operación de contorneado sobre acero.

Para estudiar la configuración para la clasificación, se considera la distancia intra-clúster bajo la DTW para $(K_{i,j})_{1 \leq i \leq p, 1 \leq j \leq s}$ como función objetivo, y se representa la dicha función para cada configuración, fijado el número de clústeres de la duración p para los posibles números de clústeres para k-medias bajo la DTW s .

Si (p, s) representa la configuración para la clasificación, entonces aplicando el método del codo sobre la representación de la función objetivo dadas en la *Figura 4.9*, se tienen que las mejores configuraciones son $(2, 5)$, $(3, 4)$ y $(4, 4)$.

Repetiendo este análisis para cada tipo de operación considerado, y buscando elegir una configuración para la clasificación que sea válida para cada modelo independiente de la operación (salvo el arranque), se tiene que los candidatos finales son $(2, 5)$ y $(3, 4)$. Finalmente, se construyen ambos modelos y se elige como configuración para la clasificación el par $(2, 5)$ por ser el que menos divisiones realiza, y teniendo en cuenta que para ciertas operaciones (fresado sobre acero) no se utiliza el máximo de clústeres disponible.

De forma análoga se realiza el estudio para el arranque tomando como configuración para la clasificación el par $(2, 3)$.

4.4. Resultados de la clasificación

Una vez construido el modelo de clasificación y entrenado con un conjunto de datos se pueden extraer conclusiones de dicho modelo. Se supone que todas las series temporales clasificadas en el mismo clúster comparten un modelo común. En consecuencia, una realización de una operación es usual cuando la proporción de elementos presentes en el clúster al que ha sido asociado respecto del total es superior a un cierto α .

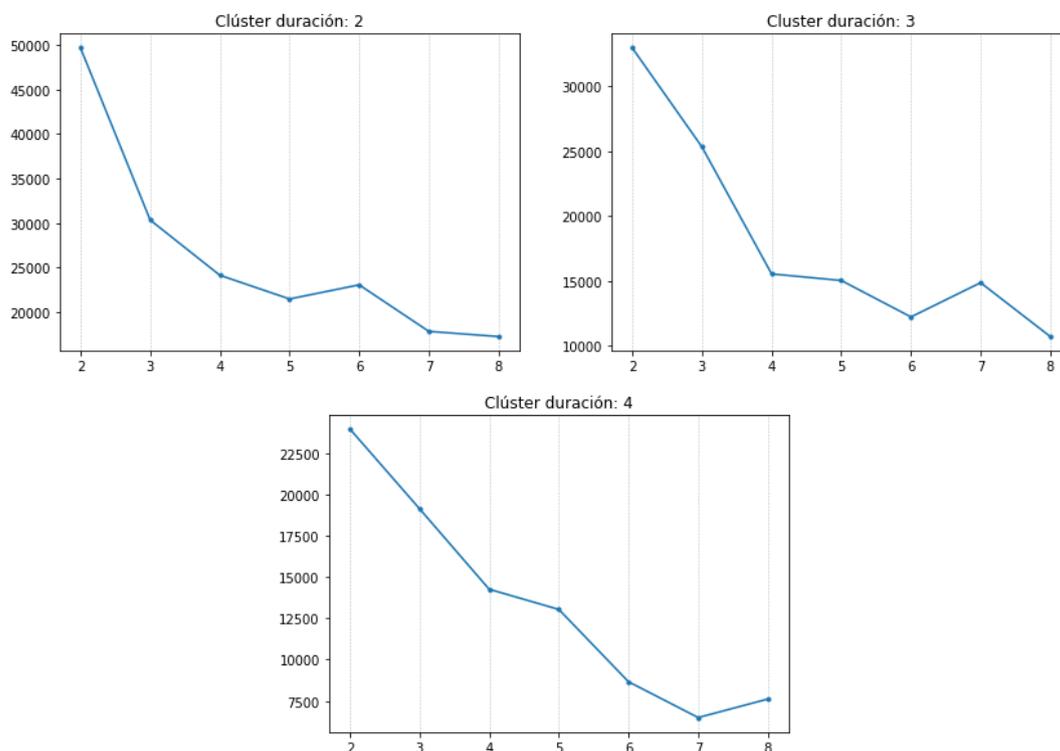


Figura 4.9: Representación de la distancia intra-clúster según el modelo descrito, para la operación de contorneado sobre acero.

Fuente: [jarroba](#)

Se consideran las operaciones de Planteado, Contorneado, Fresado y Taladrado sobre el acero (*opermat* de 23, 33, 53, 63 respectivamente) y los arranques de la máquina (identificados por *opermat* de -10). Para cada uno de estos tipos de actividad que se realiza en la máquina fresadora se va a construir un modelo de clasificación como en la *Sección 4.2*.

En primer lugar, se debe determinar el valor α que discrimina si una operación realizada es usual o anómala. Construidos los modelos, se procede de forma experimental representando el porcentaje de elementos asignados a cada clúster (*Figura 4.10*).

De dicha representación (*Figura 4.10*) se deduce que el valor de α debe estar entre 0,05 y 0,15 para las operaciones y entre 0,10 y 0,20 para los arranques. Para determinarlo con exactitud se analizan diferentes valores de α para cada uno de los casos (*Figura 4.11*), bajo la hipótesis de que el número de operaciones o arranques considerados anómalos es inferior al número número de operaciones o arranques considerados usuales.

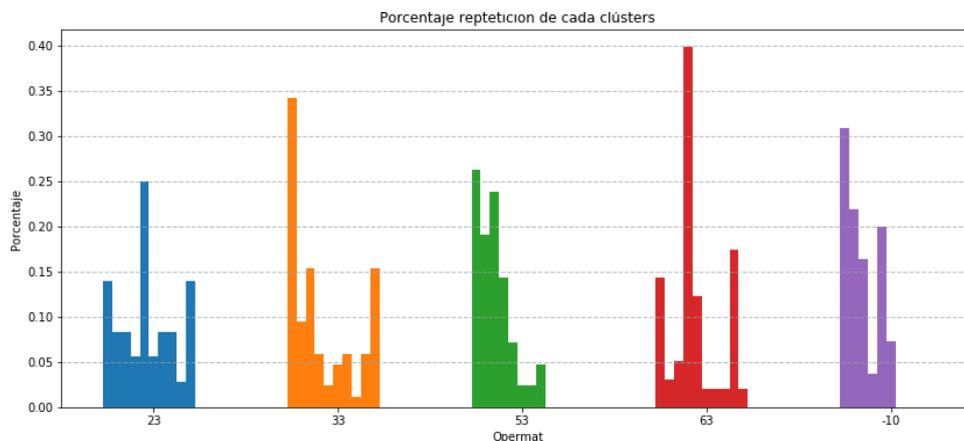


Figura 4.10: Representación del porcentaje de elementos asociados a cada clúster respecto del total clasificados para cada *opermat*.

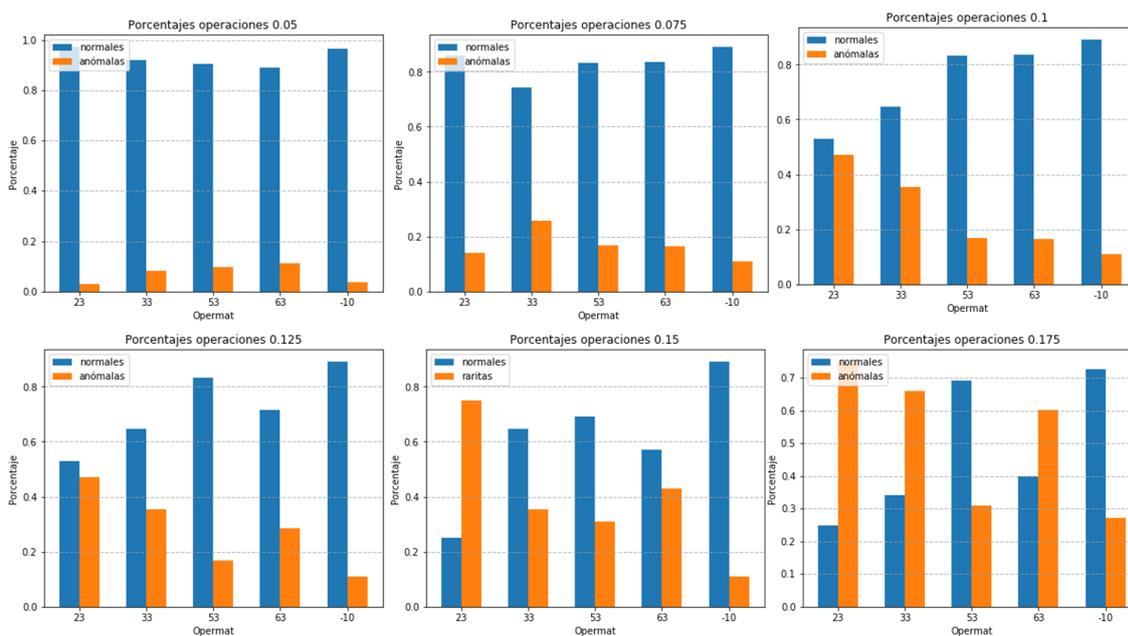


Figura 4.11: Representación del porcentaje de operaciones consideradas normales y anómalas para diferentes valores de tolerancia discriminando por el *opermat*

Así, el valor de tolerancia α se fija en 0,10 (0,125 también sería un valor adecuado), pues por definición del problema se supone que las operaciones anómalas son inferiores a una cuarta parte del total y dicha proporción se rompe cuando $\alpha \geq 0,15$. Análogamente, para el arranque se toma un valor de tolerancia de 0,15, pues el porcentaje de arranques anómalos aumenta considerablemente para valores mayores de α . Estos valores de tolerancia considerados discriminan como actividades anómalas entorno al 20 – 25 % de las operaciones y 10 – 15 % de los arranques.

4.4.1. Evaluación del modelo

El propio modelo indica qué actividades (operaciones o arranques) del conjunto de entrenamiento son poco frecuentes. Además, dada una nueva actividad, esta se puede evaluar mediante el modelo correspondiente. Para ello, se asigna primero por la duración y, seguidamente, se busca el centroide (media de Fréchet) que haga mínimo la distancia DTW de la serie temporal asociada a la nueva acción con cada centroide.

Siguiendo esta técnica, se evalúan los datos desde mayo-2019 al 1 de septiembre-2019⁴ con el respectivo modelo construido, valorando a su vez la consistencia del modelo. En la *Figura 4.12* se ilustran los resultados, y como consecuencia, se tiene que los modelos son consistentes en el tiempo (salvo quizás para la operación de planeado sobre acero *opermat 23*), suponiendo que la frecuencia en la que se producen las anomalías se mantiene en el tiempo.

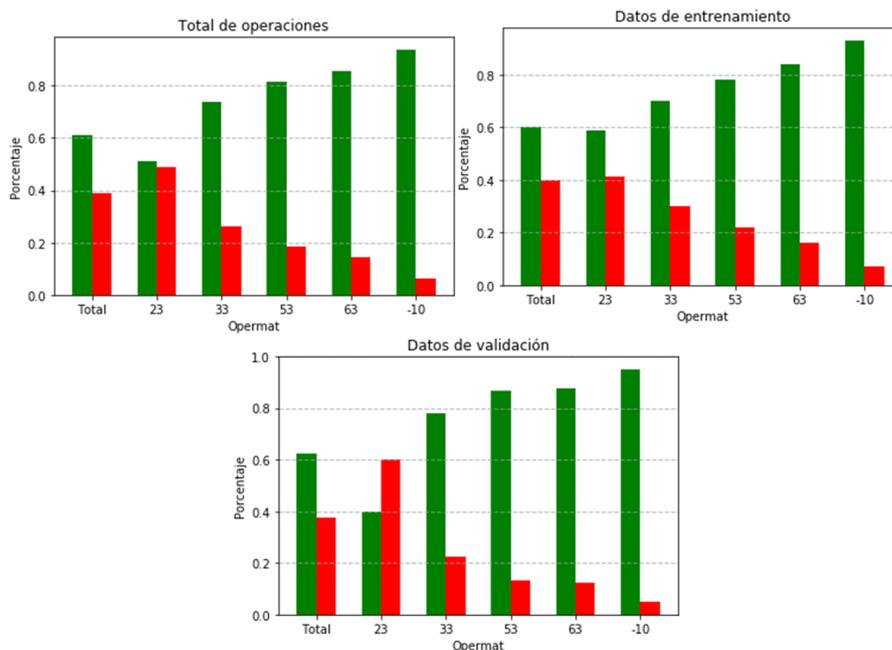


Figura 4.12: Representación de los resultados de evaluar las actividades realizadas con sus correspondientes modelos generados, para la totalidad de los datos, los datos de entrenamiento (septiembre-19 en adelante) y los datos de prueba (de mayo-19 a septiembre-19)

⁴Se disponen de datos desde mayo-2019 hasta mayo-2020 para los arranques o marzo-2020 para las operaciones. Los respectivos modelos se entrenan con los datos disponibles a partir de septiembre-2019 (inclusive) y se valida sobre los datos recogidos entre mayo-2019 hasta el día 1 septiembre-2019. Al primer subconjunto se le denomina datos de entrenamiento y al segundo conjunto se le denomina datos test. La proporción entre ambos conjuntos es respectivamente de, aproximadamente, 2/3 y 1/3, respecto del total.

4.4.2. Ventajas del modelo híbrido

El algoritmo de k-medias bajo la distancia DTW presenta el inconveniente de tener complejidad polinómica de orden 5, $\mathcal{O}(n^5)$, dado que el cálculo de la distancia DTW entre dos series tiene complejidad cuadrática. Por contra, el problema de clasificación por k-medias tradicional para conjuntos de datos unidimensionales es un problema de complejidad a lo sumo cuadrática.

Por tanto, el hecho de dividir el conjunto total de las series temporales en subconjuntos más pequeños sobre los que aplicar el algoritmo de k-medias bajo la DTW con un algoritmo de menor complejidad, supone una notable mejora en la eficiencia del modelo. En la *Figura 4.13* se muestra el tiempo requerido para diferentes configuraciones, de la operación de contorneado sobre acero, siendo notable la diferencia de no dividir por la duración a realizar dos divisiones.

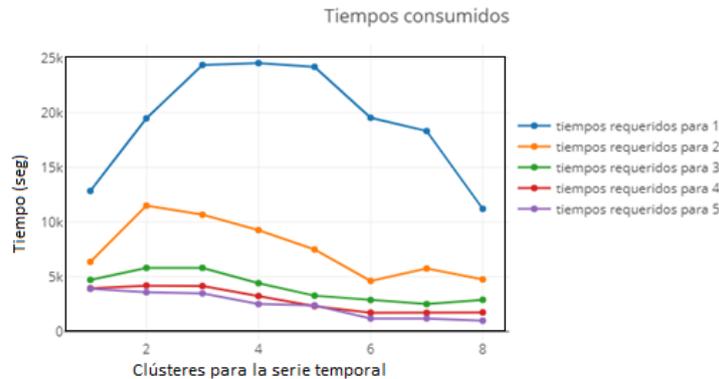


Figura 4.13: Representación de los tiempos requeridos en la construcción de modelos con diferentes configuraciones de la operación de contorneado sobre el acero. Cada representación se corresponde con un valor fijo de clústeres para la duración para cada número de clústeres para la clasificación bajo DTW.

Otra ventaja de esta técnica de división, según la duración, es que disminuye el riesgo de deformaciones temporales demasiado extensas de la distancia DTW. Pues, fijando el tamaño de los centroides como el mínimo de la longitud de las series a clasificar, y dada la alta variabilidad de los tamaños de series considerados (de 600 a 3600), es común la situación en que la longitud de la serie a clasificar quintuple la longitud del centroide. Agrupando por la duración se consigue reducir la máxima diferencia entre la longitud de las series a clasificar.

Para concluir este capítulo, cabe mencionar la importancia de los parámetros del modelo. Cuando se produce una anomalía, lo que se está generando es una desviación en un fragmento de la serie temporal respecto al supuesto modelo usual. Por tanto, dos anomalías diferentes tienden a estar a una distancia mayor entre sí que respecto a una acción usual (en general, los clústeres anómalos tienen menos de 5% de los elementos). En consecuencia, cuando el número de clústeres no es lo suficientemente grande ciertas operaciones anómalas son clasificadas como usuales.

Por otro lado, el valor de tolerancia α puede resultar engañoso y afecta enormemente a los resultados. Mientras que, un valor de α de 0,1 marca como anómalas el 23,67% de las operaciones, un valor de 0,2 marca como anómalas el 57,66% de las operaciones y un valor de 0,3 el 70,86%. Por contra, un valor de α de 0,05 marca como anómalas un escaso 9% de las operaciones.

Capítulo 5

Conclusiones

Este trabajo ha sido enfocado a la resolución del *problema de detección de anomalías sobre un conjunto de datos estructurados como series temporales*, derivado del problema de detectar las actividades anómalas realizadas en una máquina fresadora CNC sensorizada. La solución propuesta ha sido un modelo híbrido de clasificación no supervisada o *clustering*, el cuál realiza una doble clasificación atendiendo en primer lugar a parámetros propios de la serie, y seguidamente, usando la propia serie temporal.

Esta solución lleva asociado un segundo problema, la comparación de series temporales. Para ello se ha optado por utilizar la distancia *Dynamic Time Warping* o DTW, que permite “deformar” las series temporales para detectar ocurrencias con distinto momento temporal asociadas a un mismo evento (*Sección 2.2*). Para ello, se aborda el concepto de extensión de una serie temporal y las peculiaridades de la distancia DTW (*Sección 2.3*), consiguiendo que esta sea una semi-métrica.

A su vez, esta técnica de comparación permite resolver otro de los problemas asociados al *clustering* de series temporales: el cálculo de los centroides. Con ello, a partir de la DTW se pueden establecer ciertas asociaciones temporales entre los índices (instantes de medición) de las series temporales y establecer una medida promedio, denominada *Media de Fréchet*, entre un conjunto de series temporales (*Sección 3.2.1*).

En consecuencia, gracias a la DTW se puede adaptar el algoritmo de *clustering* k-medias tradicional (aplicable a conjuntos en los que todos sus miembros tienen la misma dimensión) (*Sección 3.1.1*) a su aplicación sobre un conjunto de series temporales donde cada serie tiene una longitud (dimensión) que puede ser diferente al resto *Sección 3.2*.

Finalmente, estas técnicas descritas en los *Capítulos 2 y 3* han sido aplicadas a un caso de estudio real (*Capítulo 4*) sobre una máquina fresadora CNC sensorizada, en pleno funcionamiento y sobre la que se realizan un conjunto determinado de actividades. Para ello, cada actividad realizada ha sido representada como una serie temporal según el consumo de *corriente de fase* y preprocesadas aplicando el *algoritmo de Medias Móviles* para eliminar los efectos estacionarios

de la corriente.

Para clasificar estas series se ha optado por un modelo híbrido (*Sección 3.4*), en el que se clasifica por la duración y por la propia serie temporal (*Sección 4.2*). Con dicho modelo, y bajo la hipótesis de que toda serie asignada a un mismo clúster tiene un proceso estocástico común como modelo, se determina que una operación es usual o anómala si el clúster asignado contiene un porcentaje de operaciones, respecto al total, mayor o menor que un cierto α prefijado.

Por consiguiente, el modelo tiene una serie de parámetros que deben ser determinados, para lo que se recurre a un estudio experimental (*Sección 4.3*), de gran relevancia en los resultados. Además, se cumple que, cuando el número de actividades con los que entrenar el modelo es lo suficientemente grande, el modelo es consistente en el tiempo (operaciones de contorneado y taladrado sobre el acero y arranques de la máquina).

Por contra, el problema de construcción del modelo es un problema *np-hard* con complejidad de orden 5, lo que supone que se requiere una gran capacidad de cómputo para entrenarlo. Sin embargo, el modelo híbrido reduce dicha capacidad (medido como tiempo requerido), pues divide el conjunto total bajo el algoritmo de k-medias tradicional, de menor complejidad, reduciendo el cardinal del conjunto de datos sobre el cual se aplica el algoritmo de k-medias bajo la DTW, de complejidad mayor.

Notaciones y terminología

$\mathbb{N}, \mathbb{Z}, \mathbb{R}$ Conjunto de los naturales, enteros y reales respectivamente

$A \subset B, A \subseteq B$ El conjunto A esta contenido estrictamente o esta contenido pudiendo ser igual en B

\mathbb{E}, Var, Cov Esperanza, varianza y covarianza

\cup Unión

\sqcup Unión disjunta

$N(\mu, \sigma^2)$ Normal de media μ y varianza σ^2

$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ Series temporales

$\mathbf{T}, \mathbf{S}, \mathbf{R}$ Tendencia, estacionalidad y residuo de una serie temporal

d_E Distancia euclídea

$d_{DTW}(\mathbf{X}, \mathbf{Y})$ Distancia *Dynamic Time Warping* entre las series temporales \mathbf{X} e \mathbf{Y}

A_{DTW} Matriz asociada a la DTW

d_{p-DTW} Distancia p -DTW

\mathcal{T} Conjunto de todas las series temporales de longitud finita

$\mathbf{Y} \gg \mathbf{X}$ La serie temporal \mathbf{Y} es una extensión de la serie \mathbf{X}

$\mathbf{X} \sim \mathbf{Y}$ Las series \mathbf{X} y \mathbf{Y} pertenecen a la misma clase de equivalencia de la identificación por deformación ($d_{DTW}(\mathbf{X}, \mathbf{Y}) = 0$)

\mathcal{D} Conjunto de datos

Apéndice A

Información extra sobre el caso de estudio

A.1. Relaciones de los materiales, operaciones y herramientas

En la siguiente tabla (*Tabla A.1*) se recogen las diferentes herramientas que pueden ser usadas en la máquina fresadora CNC que se estudia.

Herramienta	Código	Herramienta	Código
Centrador “bailarina”	1	∅ 32 octogonal	2
∅ 63 octogonal	3	∅ 80 octogonal	4
∅ 50 escuadrado	5	∅ 80 M4	6
∅ 65 APKT invertido	7	∅ 18	8
∅ 20 APKT largo	9	∅ 16 APKT	10
∅ 18 APKT	11	∅ 30 APKT	12
∅ 16	13	∅ 30	14
∅ 32	15	∅ 50	16
Mandrino	17	Cabeza mandrinadora	18
Portapinza ∅ 0-3	19	Portapinza ∅ 3-14	20
Cono ∅ > 14	21	Achafado ∅ 12	22
Achafado ∅ 22	23	Turbo	24
Alineado-Reloj comparador	25	Portaherramienta Ni-Fresa	26
Portapinza-Fresa	27	∅ 12	28
∅ 18 largo	29	broca de plaguitas	30
∅ 20	31		

Tabla A.1: Herramientas de la fresadora y sus códigos

Además, el uso de una herramienta depende de la operación que se realiza y del material sobre el que se trabaja. La siguiente tabla (*A.2*) recoge dichas dependencias.

Operación	Materiales	Herramientas
Toma de ceros	Indiferente	- Centrador "bailarina"
Planeado	Acero y Acero Inox.	- \varnothing 32 ortogonal - \varnothing 63 ortogonal - \varnothing 80 ortogonal
	Aluminio y Plásticos	- \varnothing 50 escuadrar - \varnothing 80 M4 - \varnothing 65 APKT inverso
Conotorneado/ Copiado	Acero y Acero Inox.	- \varnothing 12 - \varnothing 16 APKT - \varnothing 18 APKT largo - \varnothing 18 APKT - \varnothing 50 escuadrar - \varnothing 30 APKT
	Aluminio y plásticos	- \varnothing 16 - \varnothing 30 - \varnothing 18 - \varnothing 32 - \varnothing 18 largo - \varnothing 50
Mandrinado	Indiferente	- Mandrino - Cabeza mandrinadora
Taladrado	Indiferente	- Portapinzas \varnothing 0-3 - Portabroca \varnothing 3-14 - Cono \varnothing >14 - Broca Plaquilla
Especiales	Indiferente	- Achaflado \varnothing 12 - Achaflado \varnothing 22 - Turbo
Fresado	Indiferente	- Portaherramienta Ni-Fresa - Portapinza-Fresa
Alineado	Indiferente	- Reloj comparador

Tabla A.2: Relación de las operaciones, herramientas y materiales

A.1.1. La base de datos

Desde marzo del 2019, los sensores han estado realizando mediciones y almacenando información. Para contener y estructurar toda la información medida, se dispone de una base de datos relacional basada en *SQL Server* denominada CNC. En esta base de datos se disponen de varias tablas con los registros del funcionamiento y las mediciones. De esas tablas interesan especialmente dos: CNC_Funcionamiento y TiempoOperaciones.

La tabla principal, denominada CNC_Funcionamiento, almacena la información recogida cada segundo por los sensores y se compone de los campos presentes en la *Tabla A.3*.

Además de la información registrada por los sensores, el usuario operador de la fresadora introduce la información correspondiente a la operación, el material, la orden de fabricación y la herramienta que va a utilizar cuando comienza el trabajo con una pieza, siendo ella interfaz, a través de un programa de control instalado en un equipo con pantalla táctil, responsable de añadirlas en cada medición a la tabla que registra el funcionamiento.

Campo BBDD	Introduce información	Tipo de dato y rango de valores	Notas
Time_Stamp	Sistema	Cadena de caracteres con formato: año-mes-día hora:mi:-seg	Fecha en la que se escribe la medición en la base de datos
Time_Stamp_ms	Sistema	Entero positivo	Tiempo en milisegundos que transcurren en la medición
CNC_AcelerometroX	Acelerómetro-eje-x	Real	
CNC_AcelerometroY	Acelerómetro-eje-y	Real	
CNC_AcelerometroZ	Acelerómetro-eje-z	Real	
CNC_CorrienteFase1	Sensor de fase de corriente 1	Real positivo	
CNC_CorrienteFase2	Sensor de fase de corriente 2	Real positivo	
CNC_CorrienteFase3	Sensor de fase de corriente 1	Real positivo	
CNC_Temperatura	sonda de temperatura	Entero	
CNC_Sonido	Detector de sonido	Real positivo	
CNC_puerta	Sensor de la puerta	0 ó 1	Puerta abierta (0) o cerrada (1)
operacion_id	Usuario	Entero entre 0 y 7	Código de la operación realizada
material_id	Usuario	Entero entre 0 y 5	Código del material utilizado
orden-fab	Usuario	Cadena de caracteres	Identificador de la pieza
herramienta_id	Usuario	Entero entre 0 y 31	Identificador de la herramienta

Tabla A.3: Campos e información en la tabla CNC_Funcionamiento

Por último, con el objetivo de detectar errores en las mediciones, se incluyó el campo `Time_Stamp_ms` que registra el tiempo (en milisegundos) que se requiere para la realización de las mediciones desde que se produce el evento que realiza la medición. Con ello, cuando el valor presentado sea superior a 1000 se considera que se ha “perdido” una medición y puede ser un indicativo de error en los sensores.

Paralelamente, se dispone de una segunda tabla, denominada `TiempoOperaciones`, en la cual se registra, de forma independiente a la tabla principal, el comienzo y fin de cada operación, junto al identificación de la operación, de la herramienta y la orden de fabricación.



Figura A.1: Equipo con pantalla táctil con el sistema de control instalado

El objetivo de esta segunda tabla es conocer el momento de comienzo y fin de operaciones para poder procesar más fácilmente el histórico de datos almacenados. También sirve para verificar la integridad de la información almacenada. Los campos de esta tabla están recogidos en *Tabla A.4*.

Campo BBDD	Introduce información	Tipo de dato y rango de valores	Notas
Time_Stamp	Sistema	Cadena de caracteres con formato: año-mes-día hora:mi:-seg	Fecha en la que se finaliza la acción. Actúa como identificador y clave ajena respecto de la tabla principal
Time_Stamp_ms	Sistema	Entero positivo	Tiempo en milisegundos que transcurren entre mediciones y escritura
operacion_id	Usuario	Entero entre 0 y 7	Código de la operación realizada
herramienta_id	Usuario	Entero entre 1 y 31	Identificador de la herramienta
inicioOp	Sistema	Cadena de caracteres con formato: día/mes/año hora:min:seg	Fecha de inicio de la operación.
finOp	Sistema	Cadena de caracteres con formato: día/mes/año hora:min:seg	Fecha de fin de la operación

Tabla A.4: Campos e información de la tabla *tiempoOperaciones*

La relación entre las dos tablas de uso se puede enunciar como “cada fila de la tabla TiempoOperaciones se relaciona con todas las filas de la tabla CNC_Funcionamiento cuya fecha de realización (Time_Stamp) este comprendido entre los valores de inicio (inicioOp) y fin (finOp) de la fila seleccionada”.

Apéndice B

Implementación en pseudocódigo de los algoritmos

En esta sección se presentan las implementaciones en pseudocódigo correspondientes a los algoritmos relacionados con la distancia DTW (cálculo de la matriz DTW, curva de identificación y asociaciones de índices), así como de clustering, tanto sobre puntos (estándar) como su variante para ser aplicado sobre series temporales de longitud arbitraria.

B.1. Implementación del algoritmo de medias móviles

Se presenta una implementación de la versión que considera momentos pasados y futuros del algoritmo de medias móviles.

Pseudocódigo del Algoritmo de medias móviles para la versión centrada

```

1  real medias_movil(X: vector [1..n], ventana: integer ,
2  pesos: vector [1..n]) {
3  l: int
4  u: int
5  l := longitud(X)
6  u := (ventana - 1)/2
7  y : vector [1..n]
8  //Inicializado el vector de medias moviles
9  for i := 1..n do
10 y[i] = nan //nan es tipo nulo, no numérico y sin valor
11 end_for
12
13 //Cálculo de las medias moviles
14 for i := (u+1)..(l-u) do
15     suma := 0
16     for j := (i-u)..(i+u) do
17         suma := suma + x[j]*pesos[j]
18     end_for
19     y[i] = suma/ventana
20 end_for
21 return(y)
22 }
```

B.2. Implementación del algoritmo DTW

En esta sección se presentan las implementaciones de los algoritmos para el cálculo de la distancia DTW, curva de deformación y asociación de índices.

B.2.1. Implementación del algoritmo DTW versión primera

El algoritmo que se presenta calcula la matriz DTW (DTW), siendo a distancia DTW el término (n, m) -ésimo de dicha matriz. El retornar el valor de la matriz DTW en lugar del valor de la distancia DTW se debe a la importancia que dicha matriz tiene en el cálculo de la curva de deformación y de las asociaciones.

Pseudocódigo del algoritmo DTW versión primera

```

1  real DistanciaDTW (x: vector [1..n], y: vector [1..m]) {
2      DIW: = matriz [1..n, 1..m]
3
4      // Inicializacion primer termino
5      DIW[1,1] := {distancia(x[1], y[1])}
6
7      //Llenado primera fila
8      for j := 1..m do
9          DIW[1,j] = distancia(x[1], y[j]) + DIW[1,j-1]
10     end_for
11
12     //Llenado primera columna
13     for i := 1..m do
14         DIW[i,1] = distancia(x[i], y[1]) + DIW[i-1,1]
15     end_for
16
17
18     //Llenado del total de la matriz
19     for i := 2..n do
20         for j := 2..m do
21             //Calculo de la distancia entre dos observaciones
22             //de la serie
23             costo := distancia(x[i], y[j])
24             //Calculo del coeficiente de la matriz DIW
25             DIW [i, j] := costo +
26                 + minimo(DIW[i-1, j], DIW[i, j-1], DIW[i-1,j-1])
27         end_for
28     end_for
29
30     return DIW
31 }

```

B.2.2. Implementación del algoritmo de cálculo de la curva de deformación

Pseudocódigo del algoritmo de cálculo de la curva de deformación

```

1  real curvaDeformacion (A: matriz [1..n, 1..m]){
2      delta := lista_vacia
3      i := n
4      j := m
5
6      // Se supone la existencia de un método asociado a las listas
7      // añadir que recibe un elemento y lo incluye en la lista
8      delta.añadir([i,j])
9
10     while i>1 and j>1 do
11         //Si se esta en la primera fila de la matriz
12         if i = 1 do
13             j := j - 1
14
15         //Si se esta en la primera columna de la matriz
16         elif j = 1 do
17             i := i - 1
18
19         //Si no se esta ni en la primera fila ni columna
20         else:
21             coste_min:=minimo(A[i-1, j-1], A[i-1, j], A[i, j-1])
22             //Elección del par de índices según el valor mínimo
23             if A[i-1, j] = coste_min do
24                 i := i - 1
25             elif A[i, j-1] = coste_min do
26                 j := j-1
27             else:
28                 i := i - 1
29                 j := j- 1
30             end_if
31         end_if
32         delta.añadir([i, j])
33     end_while
34     delta.añadir([1,1])
35
36     return(delta.invertir())}
37     //invertir es un método asociado a la lista que invierte el
38     //orden los elementos

```

B.2.3. Implementación del algoritmo de asociaciones de índices

Pseudocódigo del algoritmo para determinar las asociaciones

```

1 vector [1..n] asociaciones(phi: lista){
2 //phi es una lista indexada en 1 con P vectores de longitud dos.
3   indices: vector [1..2]
4   phiX: int
5   phiY: int
6   gamma := vector [1..n]
7
8   for i := 1..n
9     gamma[i] = lista_vacia()
10  end_for
11
12  for i := ..P
13    indices := phi[i]
14    phiX := indices [1]
15    phiY := indices [2]
16    gamma[phiX].append(phiY)
17  end_for
18  return gamma
19  }
```

B.2.4. Implementación del algoritmo DTW en la versión segunda

De forma análoga a la implementación de la versión primera del algoritmo DTW, en esta segunda implementación se devuelve la matriz DTW en vez del valor de la distancia.

El inicializar los términos no válidos a `infinito` permite que las implementaciones dadas para hallar la curva de deformación y las asociaciones de índices sean válidas sin requerir de ninguna modificación.

Pseudocódigo del algoritmo DTW versión segunda

```

1  real DistanciaDTW (x: vector [1..n], y: vector [1..m], w: int){
2      //Comprobación del requisito sobre omega
3      w := maximo(|n-m|, omega)
4
5      //Inicialización de la matriz a infinito
6      DIW:= matriz [1..n, 1..m]
7      for i := 1 .. n do
8          for j:= 1 .. m do
9              DIW[i, j] := infinito
10         end_for
11     end_for
12
13     // Inicializacion primer termino
14     DIW[1,1] := {distancia(x[1], y[1])}
15
16     //Llenado primera fila
17     for j := 1..m do
18         DIW[1,j] = distancia(x[1], y[j]) + DIW[1,j-1]
19     end_for
20
21     //Llenado primera columna
22     for i := 1..m do
23         DIW[i,1] = distancia(x[i], y[1]) + DIW[i-1,1]
24     end_for
25
26     //Llenado del total de la matriz
27     for i := 2..n do
28         for j := maximo(2,i-w) .. minimo(1,i+w) do
29             //Calculo de la distancia entre dos observaciones
30             costo := distancia(x[i], y[j])
31             //Calculo del coeficiente de la matriz DIW
32             DIW [i, j] := costo +
33                 + minimo(DIW[i-1, j], DIW[i, j-1], DIW[i-1,j-1])
34         end_for
35     end_for
36 return DIW}

```

B.3. Implementación del algoritmo DBA

El vector pertenece de la implementación siguiente se ha incluido para su futura aplicación dentro del algoritmo de k-medias. Pues en dicho caso solo interesa calcular la media DBA de aquellas series temporales que pertenecen al clúster concreto.

También cabe destacar que la longitud de la serie muestra viene dada por parámetro por T dado que se prefija en la media anterior ,C, y para su aplicación sobre k-medias se calculara sobre el conjunto de datos en una inicialización aleatoria.

Algoritmo DBA

```

1 vector DBA(C: vector [1..T], S: lista , A: lista ,
2 pertenece: vector [1..n], T: int){
3 // C es la serie temporal de la comparación
4 // S es la lista que contiene las series temporales a promediar
5 // A es la lista de asociación. De forma que el elemento A[i][s] es
6 // una lista con el conjunto de índices de la serie i-ésima
7 // que se asocian con el indice s de la serie C.
8 // pertenece es un vector de ceros y unos que indica si el dato
9 // es tenido en cuenta para el cálculo de la media
10 C_prima: vector [1..T]
11
12 for s := 1..T do
13 suma := 0
14 for i:= 1..m do
15 for indice in A[i][s]
16 suma = suma + S[i] [indice] * pertenece[i]
17 end_for
18 end_for
19 C_prima[j] = suma/m
20 end_for
21 return C_prima}

```

B.4. Implementación del algoritmo de clustering sobre vectores

La implementación presentada se corresponde a la versión dada para según matriz de pertenencia. Se ha separado del propio algoritmo tres funcionalidades para facilitar su comprensión. Así, en la implementación del algoritmo de k-medias aparece el uso de las funciones `calculo_matriz_pertenencia`, `calculo_centroides` y `calculo_variacion` cuyas implementaciones se muestran posteriormente.

Pseudocódigo del algoritmo de k-medias

```

1 matriz[1..m, 1..k] kmeans (data: vector[1..m, 1..n], k: int ,
2     tolerancia: int) {
3 //Cada fila de la matriz data es un vector dato de dimensión 1xn
4   W: matriz[1..m, 1..k]
5   W_iteracion_ant: matriz[1..m, 1..k]
6   centroides: matriz[1..k, 1..n]
7   seguir: bool
8   variacion_cluster: int
9
10  seguir := True
11  variacion_cluster := infinito
12  //Inicialización matriz de pertenencia iteracion anterior
13  for i := 1..m
14    for j := 1..k
15      W_iteracion_ant[i,j] := 0
16    end_for
17  end_for
18  // Inicializacion centroides (existe una función para ello)
19  centroides = inicializacion_aleatoria_centroides()
20
21  //Iteraciones
22  while sigue
23    //Asignación de los clústers
24    W := calculo_matriz_pertenencia(data, centroides)
25    //Cálculo de los nuevos centroides
26    centroides = calculo_centroides(W, data, centroides)
27    //Evaluación de la condición de parada
28    variacion_cluster:= calculo_variacion(W, data, centroides)
29    if W = W_iteracion_ant
30      seguir := False
31    end_if
32    if variacion_cluster < tolerancia:
33      seguir := False
34    end_if
35    W_iteracion_ant := W
36  end_while
37  return W
38 }

```

Pseudocódigo para el cálculo de la matriz de pertenencia

```

1 matriz [1..m, 1..k] calculo_matriz_pertenencia (
2     data: matriz [1..m, 1..n],
3     centroides: matriz [1..k, 1..n]) {
4 //Inicialización matriz de pertenencia
5     for i := 1..m
6         for j := 1..k
7             W[i,j] := 0
8         end_for
9     end_for
10
11 //Cálculo de la matriz de pertenencia
12 for d := 1..m do
13     dist_min := infinito
14     cluster := 1
15     for c := 1..k do
16         for r:= 1..n do
17             centroide: vector [1..n]
18             centroide := centroides [c,r]
19             dato := vector [1..n]
20             dato := data [d,r]
21         end_for
22         // distancia es una función de comparación
23         dist := distancia(centroide , dato)
24         if dist < dist_min
25             dist_min := dist
26             cluster := c
27         end_if
28     end_for
29     W[d, cluster] := 1
30 end_for
31 return W
32 }

```

Pseudocódigo para el cálculo de los centroides

```

1 matriz [1..k, 1..n] calculo_centroides (W: matriz [1..m, 1..k],
2                                     data: vector [1..m, 1..n],
3                                     centroides: matriz [1..k, 1..n]){
4   for c := 1..k do
5     for t := 1..n do
6       suma: int
7       suma := 0
8       elementos_cluster: int
9       elementos_cluster := 0
10      for j := 1..m do
11        suma := W[j,c] * data[j,t]
12        elementos_cluster := W[j,c]
13      end_for
14      centroides [c,t] := suma/elementos_cluster
15    end_for
16  end_for
17  return centroides
18 }

```

Pseudocódigo para el cálculo de la variación total intra-clúster

```

1 real calculo_variacion (W: matriz [1..m, 1..k],
2                          data: vector [1..m, 1..n],
3                          centroides: matriz [1..k, 1..n]){
4   variacion: int
5   variacion := 0
6   for t := 1..k
7     for i := 1..m
8       for r:= 1..n do
9         centroide: vector [1..n]
10        centroide := centroides [c,r]
11        dato := vector [1..n]
12        dato := data [d,r]
13      end_for
14      variacion := variacion +
15                  distancia (centroide , dato) * W[i,t]
16    end_for
17  end_for
18  return variacion
19 }

```

B.5. Implementación del algoritmo de clustering sobre series temporales

Para la implementación sobre series temporales, se realizan las modificaciones oportunas sobre la implementación en la *Sección B.4*, siguiendo el mismo esquema y la misma implementación modular.

Se han de tener en cuenta las observaciones realizadas en la *Sección 3.2* sobre la complejidad del cálculo de la distancia DTW y la necesidad de realizar dicho cálculo el mínimo número de veces necesarias. También caben destacar el uso de las funciones `distanciaDTW`, `curvaDeformacion`, `asociaciones` implementadas en los *Apéndice B.2* y `DBA` implementada en el *Apéndice B.3*

Pseudocódigo para el cálculo de los centroides sobre series temporales

```

1 matriz [1..k, 1..n] calculo_centroides (W: matriz [1..m, 1..k] ,
2   data: lista ,
3   centroides: matriz [1..k, 1..T] ,
4   gammas: lista ) {
5   m = numero_elementos_en (data)
6   pertenece: vector [1..m]
7   for c := 1..k do
8     for i := 1..m do
9       pertenece := W[i,c]
10    end_for
11    centro := DBA(centroide[c], data, gammas, pertenece, T)
12    for j := 1..T
13      centroides[c,j] := centro[j]
14    end_for
15    return centroides
16  }
```

Pseudocódigo para el cálculo de la variación total intra-clúster

```

1 real calculo_variacion (W: matriz [1..m, 1..k] ,
2   distancias: vector [1..m]) {
3   variacion: int
4   variacion := 0
5   for t := 1..k
6     for i := 1..m
7       variacion := variacion + distancias[i] * W[i,t]
8     end_for
9   end_for
10  return variacion
11 }
```

Pseudocódigo para el cálculo de la matriz de pertenencia sobre series temporales

```

1  (matriz [1..m, 1..k], vector [1..T]) calculo_matriz_pertenencia (
2      data: lista, centroides: matriz [1..k, 1..T]) {
3      //Inicialización matriz de pertenencia
4      m := numero_elementos_en(data)
5      for i := 1..m
6          for j := 1..k
7              W[i,j] := 0
8          end_for
9      end_for
10     gammas: lista_vacia
11     distancias: vector [1..m]
12     //Cálculo de la matriz de pertenencia
13     for d := 1..m do
14         dist_min := infinito
15         cluster: int
16         dato: vector
17         dato: data[d]
18         L: int
19         L := longitud(dato)
20         for c := 1..k do
21             for r:= 1..n do
22                 centroide: vector [1..T]
23                 centroide := centroides[c,r]
24             end_for
25             ADTW := distanciaDTW(centroide, dato)
26             dist = ADTW[L,T]
27             if dist < dist_min
28                 dist_min := dist
29                 cluster := c
30                 A_DTW_cluster = ADTW
31             end_if
32         end_for
33         distancias[d] = dist_min
34         W[d,cluster] := 1
35         phi := curvaDeformacion(A_DTW_cluster)
36         gammas.añadir(identificacionIndices(phi))
37     end_for
38     return (W, gammas, distancias)
39 }

```

Pseudocódigo del algoritmo de k-medias

```

1  matriz[1..m, 1..k] kmeans (data: lista , k: int ,
2                               tolerancia: int) {
3  //Cada fila de la matriz data es un vector dato de dimensión 1xn
4  m: int
5  m := numero_elementos_en(data)
6  W: matriz[1..m, 1..k]
7  W_iteracion_ant: matriz[1..m, 1..k]
8  centroides: matriz[1..k, 1..n]
9  seguir: bool
10 variacion_cluster: int
11
12 seguir := True
13 variacion_cluster := infinito
14 //Inicialización matriz de pertenencia iteracion anterior
15 for i := 1..m
16     for j := 1..k
17         W_iteracion_ant[i,j] := 0
18     end_for
19 end_for
20 // Inicializacion centroides (existe una función para ello)
21 centroides = inicializacion_aleatoria_centroides()
22
23 //Iteraciones
24 while seguir
25     //Asignación de los clústers
26     (W,gammas, distancias) := calculo_matriz_pertenencia(
27                                     data, centroides)
28     //Cálculo de los nuevos centroides
29     centroides = calculo_centroides(W, data,
30                                     centroides , gammas)
31     //Evaluación de la condición de parada
32     variacion_cluster:= calculo_variacion(W, distancias)
33     if W == W_iteracion_ant
34         seguir := False
35     end_if
36     if variacion_cluster < tolerancia:
37         seguir := False
38     end_if
39     W_iteracion_ant := W
40 end_while
41 return W
42 }
```

Apéndice C

Ejemplo de realizaciones de ciertas actividades

En este apéndice se recogen la representación de algunas actividades (operaciones y/o arranques) realizadas (mediante la correspondiente serie temporal dada por las mediciones del sensor de corriente de fase 1), con objeto de ilustrar el caso de estudio y su enorme variabilidad.

También se incluye la evaluación de dichas actividades según su modelo correspondiente para determinar si son usuales o anómalas (recogidas en la *Tabla C.1*). Las actividades representadas han sido elegidas para ilustrar las diferentes alternativas que pueden surgir, tanto usuales como anómalas.

Id	Resultado	Id	Resultado	Id	Resultado
1	Anómalo	2	Usual	3	Usual
4	Anómalo	5	Usual	6	Usual
7	Usual	8	Usual	9	Usual
10	Anómalo	11	Usual	12	Anómalo
13	Usual	14	Anómalo	15	Usual
16	Usual	17	Usual	18	Usual
19	Usual	20	Usual	21	Anómalo
22	Usual	23	Usual	24	Anómalo
25	Usual	26	Usual	27	Usual
28	Usual	29	Usual	30	Anómalo

Tabla C.1: Resultados de evaluar las actividades seleccionadas como ejemplo por su correspondiente modelo según los identificadores dados en las *Figuras C.1, C.2, C.3, C.4 y C.5*

Operación de planeado sobre acero

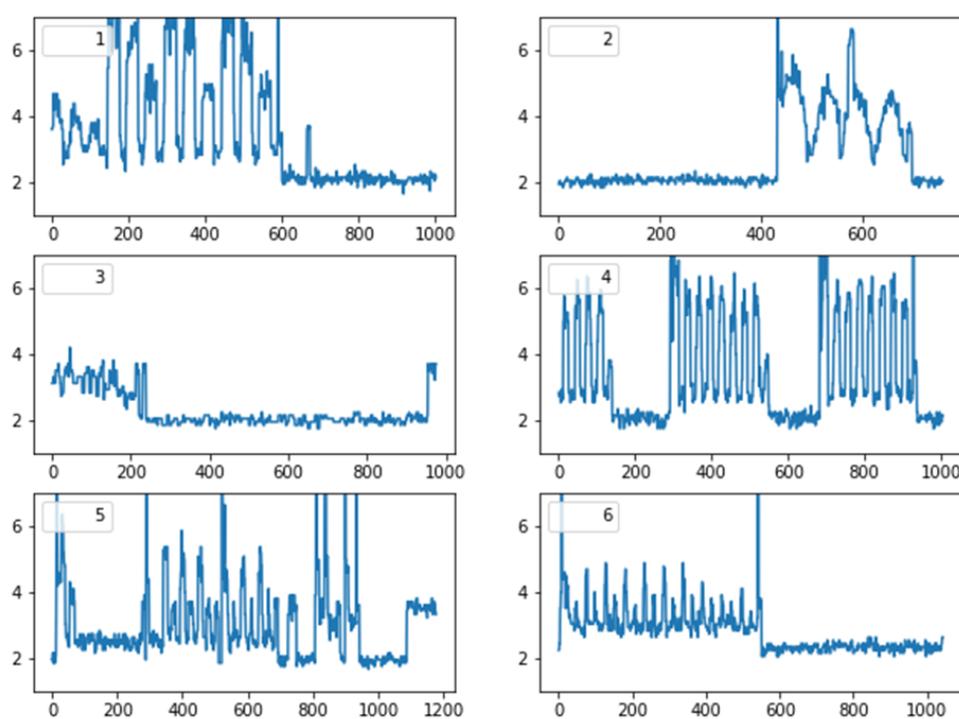


Figura C.1: Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo planeado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente

Operación de taladrado sobre acero

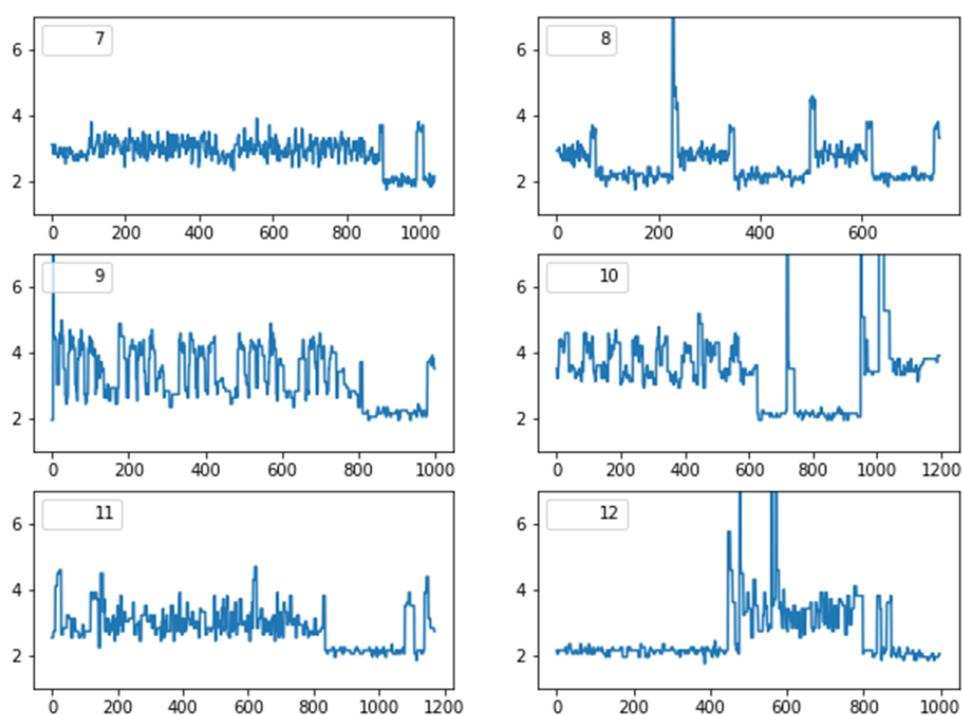


Figura C.2: Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo contorneado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente

Operación de fresado sobre acero

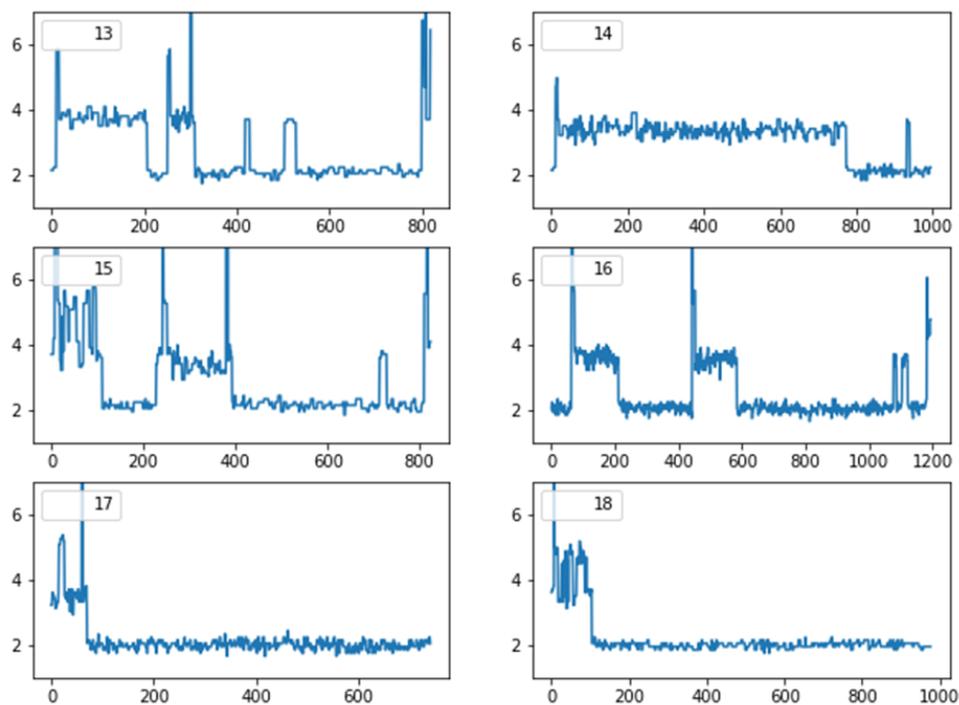


Figura C.3: Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo fresado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente

Operación de taladrado sobre acero

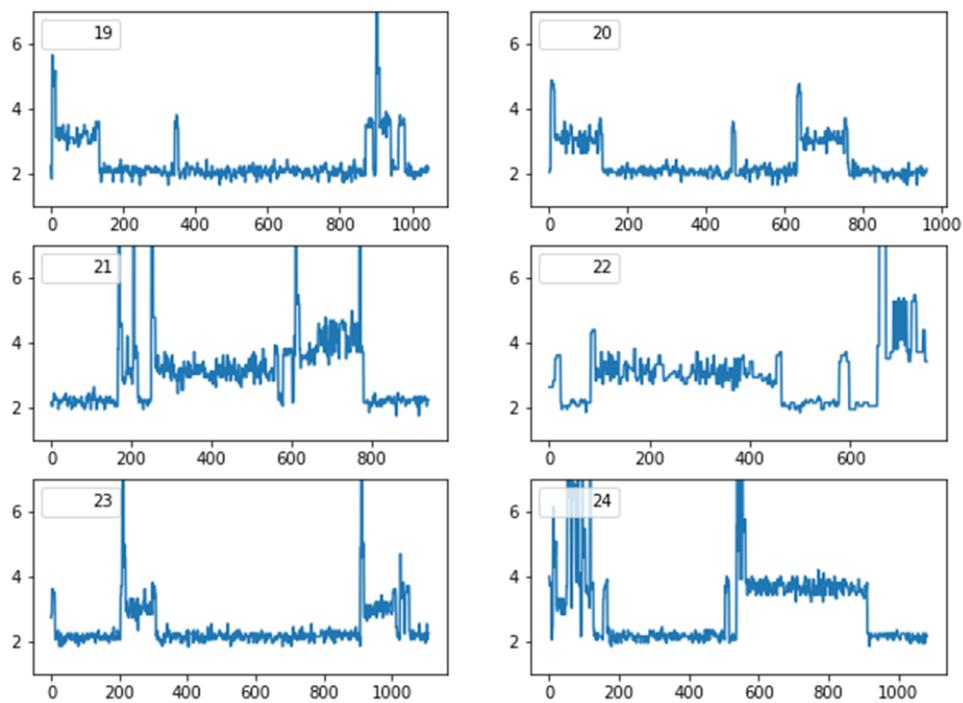


Figura C.4: Representación de las mediciones realizadas por el sensor de corriente de fase 1 para la operación tipo taladrado sobre el acero para algunas realizaciones concretas y resultado de clasificar dichas realizaciones por el modelo correspondiente

Arranques de la máquina

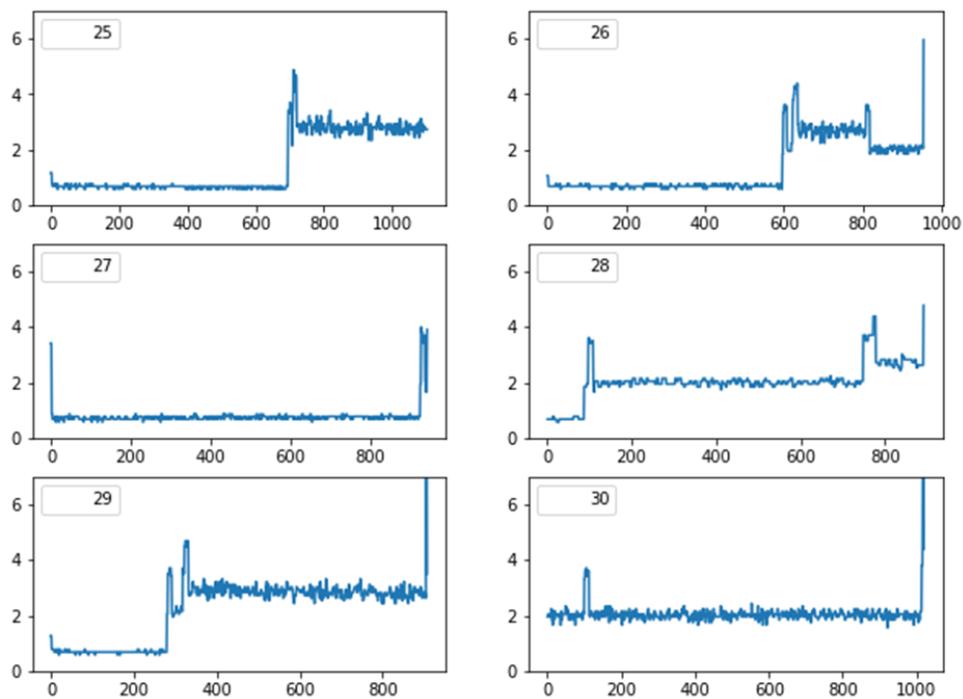


Figura C.5: Representación de las mediciones realizadas por el sensor de corriente de fase 1 para el arranque de la máquina fresadora y resultado de clasificar dichas realizaciones por el modelo correspondiente

Apéndice D

Resultados del estudio experimental

En este apéndice se recogen los resultados del estudio experimental sobre el resto de operaciones realizadas, cuyo análisis es análogo al realizado en la *Sección 4.3* para la operación de contorneado sobre acero.

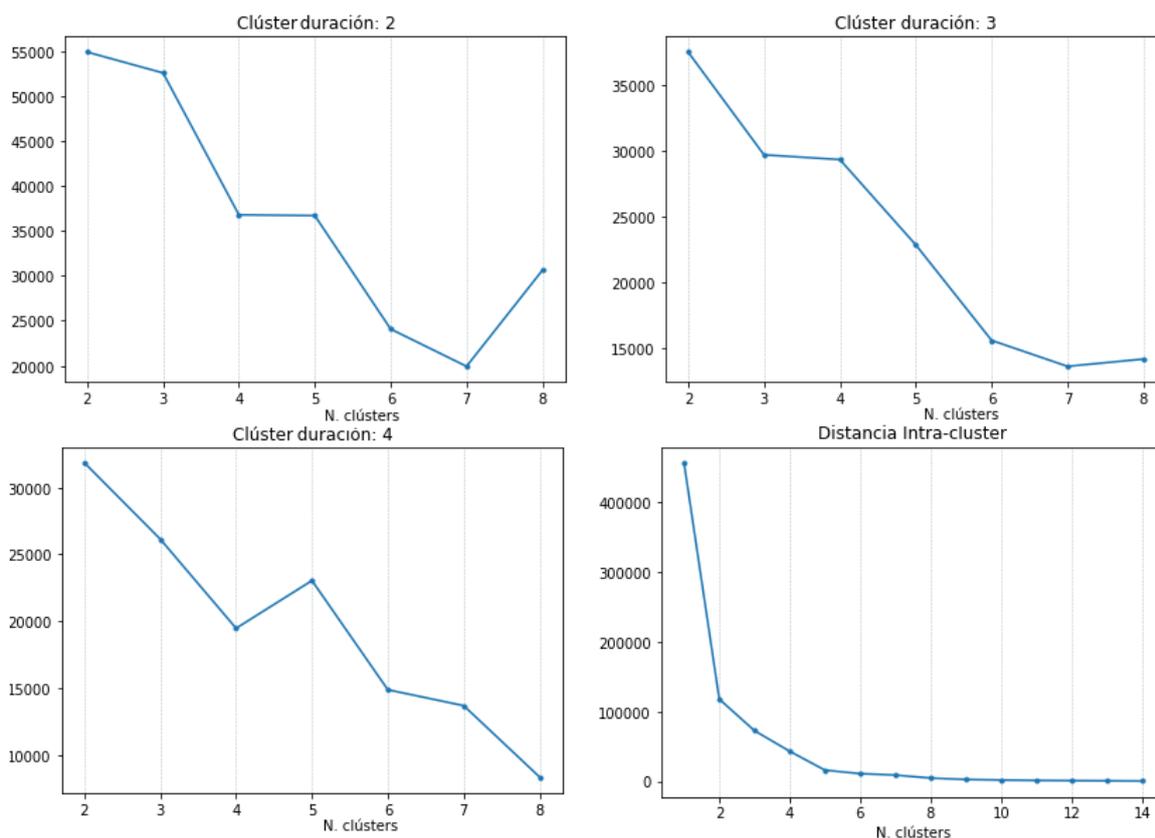


Figura D.1: Representación de la distancia intra-clúster bajo la DTW como función objetivo para la operación de taladrado sobre acero

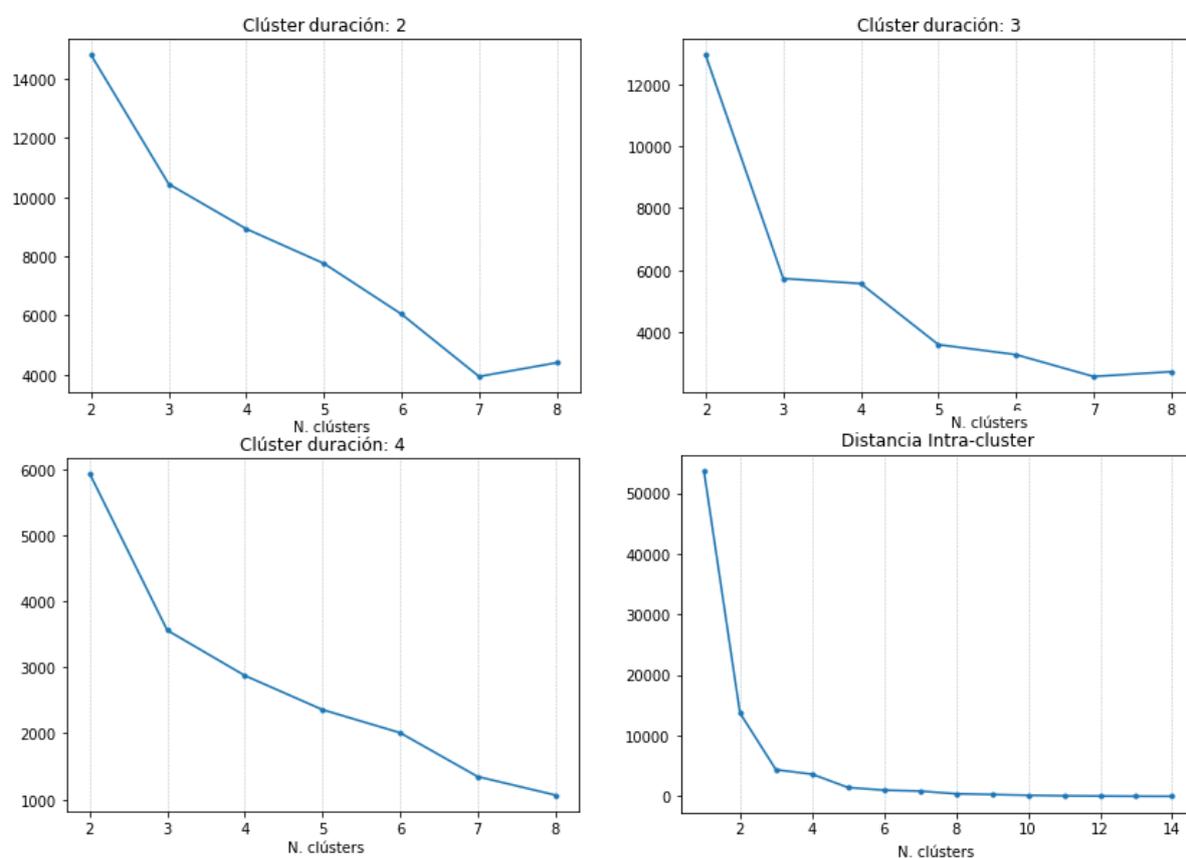


Figura D.2: Representación de la distancia intra-clúster bajo la DTW como función objetivo para la operación de fresado sobre acero.

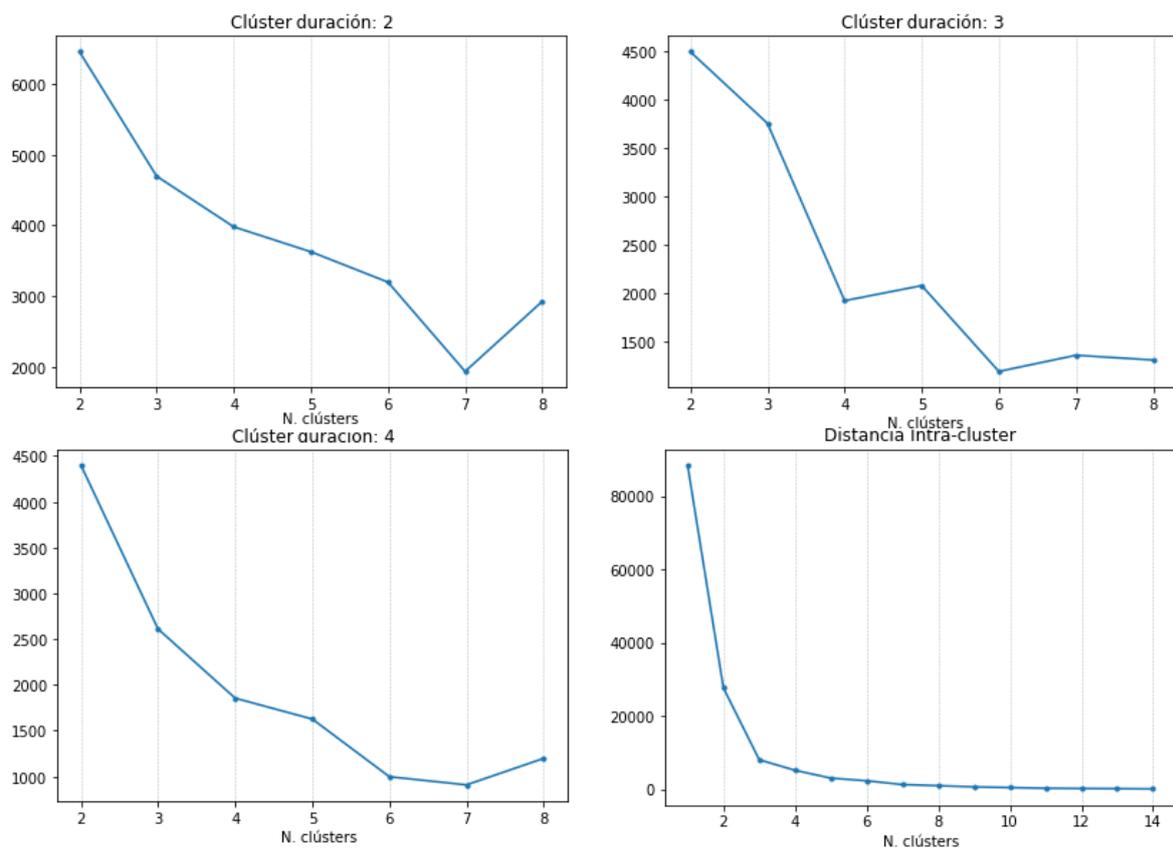


Figura D.3: Representación de la distancia intra-clúster bajo la DTW como función objetivo para la operación de planeado sobre acero

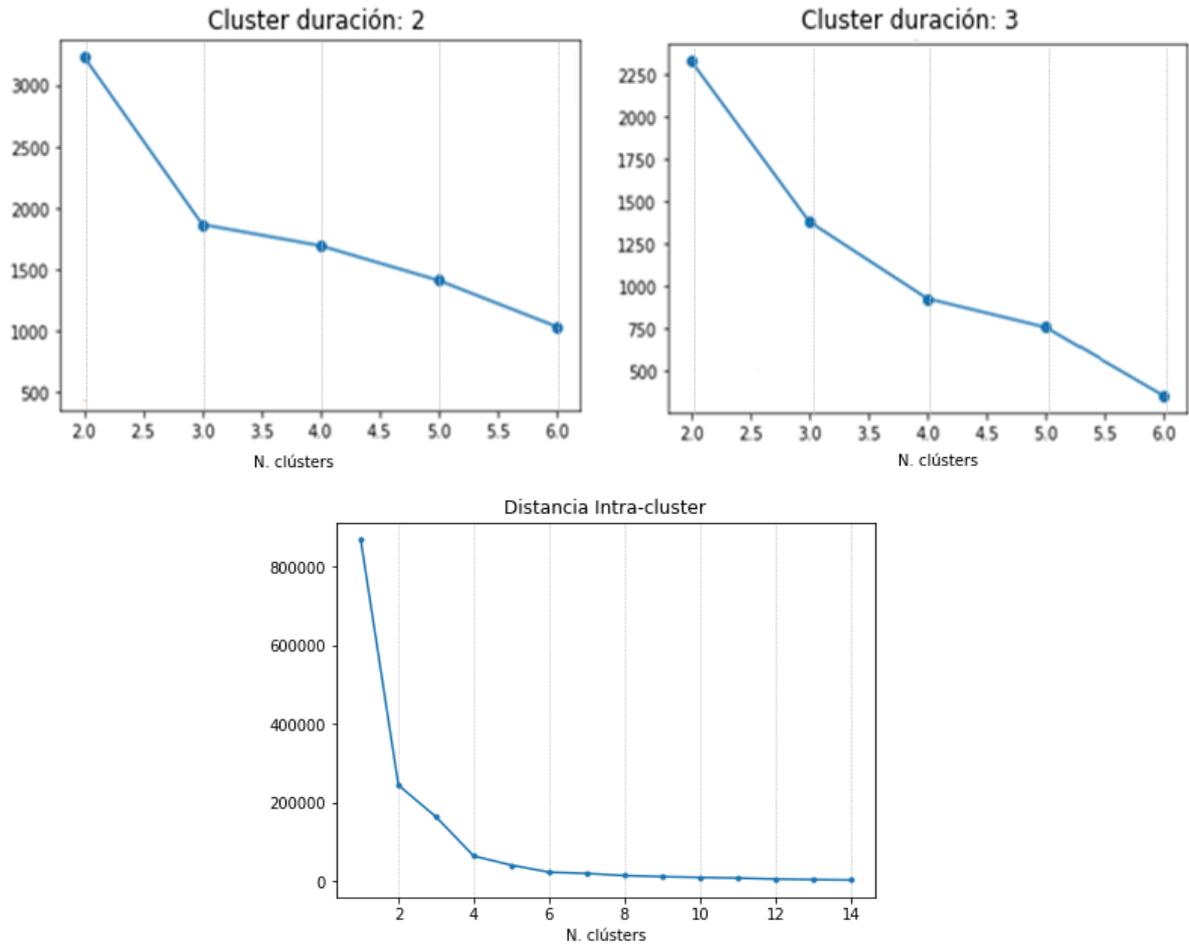


Figura D.4: Representación de la distancia intra-clúster bajo la DTW como función objetivo para los arranques de la máquina fresadora

Bibliografía

- [1] tselarn Documentation, Release 0.3.0
- [2] Peter J. Brockwell y Richard A. Davis *Introduction to Time Series and Forecasting, Third Edition*. Ed. Springer.
- [3] Geoffrey Grimmett y David Stirzaker *Probability and Random Processes. Third Edition*. Ed. Oxford.
- [4] Jacquie Nesbitt *Suavización 1: Métodos Distintos de Suavizado*. Numxl Soporte, Notas Técnicas. Disponible [aquí](#) (Mayo 2020)
- [5] D. Guijo-Rubio, A. M. Durán-Rosal, P. A. Gutiérrez, A. Troncoso and C. Hervás-Martínez. *Time-Series Clustering Based on the Characterization of Segment Typologies*. IEEE Transactions on Cybernetics, ISSN 2168-2275. Disponible [aquí](#) (Junio 2020)
- [6] Brijnesh J.Jain. *Making the dynamic time warping distance warping-invariant*. Pattern Recognition, Octubre 2019, Vol. 94, pp. 35-52. Disponible [aquí](#) (Junio 2020)
- [7] Toni Giorgino. *Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package*. Journal of Statistical Software August 2009, Volume 31, Issue 7
- [8] Diego F. Silva and Gustavo E. A. P. A. Batista. *Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation*. SIAM International Conference on Data Mining, May 2016
- [9] Z. Zhang, R. Tavenard, A. Bailly, X. Tang, P. Tang, T. Corpetti. *Dynamic time warping under limited warping path length*. Information Sciences, vol. 393, pp. 91–107, 2017
- [10] Z. Geler, V. Kurbalija, M. Ivanović, M. Radovanović and W. Dai. *Dynamic Time Warping: Itakura vs Sakoe-Chiba*. 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 2019, pp. 1-6.
- [11] F. Petitjean, A. Ketterlin y P. Gancarski. *A global averaging method for dynamic time warping, with applications to clustering*. Pattern Recognition, Elsevier, 2011, Vol. 44, Num. 3, pp. 678-693. Disponible [aquí](#) (Junio 2020)
- [12] Brijnesh Jain and David Schultz. *On the Existence of a Sample Mean in Dynamic Time Warping Spaces*. arXiv:1610.04460v3 5 Mar 2018. Disponible [aquí](#) (Junio 2020)

- [13] K. Krishna and M. Narasimha Murty. *Genetic K-Means Algorithm*. IEEE transactions on systems, man, and cybernetics-part B, Vol. 29, N. 3, pp 433-439, junio 1999. Disponible [aquí](#) (Junio 2020)
- [14] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi y Teh Ying Wah. *Time-series clustering – A decade review*. Elsevier, Vol. 53, pp16-38, octubre-noviembre 2015. Disponible [aquí](#) (Junio 2020)
- [15] Meena Mahajan, Prajakta Nimbhorkar y Kasturi Varadarajan. *The Planar k-Means Problem is NP-Hard*. Springer, WALCOM 2009: Proceedings of the 3rd International Workshop on Algorithms and Computation, February 2009 pp 274–285. Disponible [aquí](#) (Junio 2020)
- [16] Álvaro García García *Desarrollo de modelos predictivos adaptados a los nuevos ecosistemas digitales de mantenimiento industrial*. Revista Preditécnico, Núm. 24, pp. 12-15. Disponible [aquí](#) (Abril 2020)
- [17] Álvaro García García *iDigital4: Nuevos ecosistemas de fabricación basados en el concepto de gemelo digital*. Automática e Instrumentación, Febrero 2020, Num. 517, pp. 38-39. Disponible [aquí](#) (Junio 2020)
- [18] Hermann, T. Pentek, and B. Otto. *Design Principles for Industrie 4.0 Scenarios: A Literature Review Working Paper A Literature Review*, 2015 (publicado en 2016, 49th Hawaii International Conference on System Sciences (HICSS)). Disponible [aquí](#) (Junio 2020)
- [19] Klaus Schwab. 2017. *The Fourth Industrial Revolution*. Crown Publishing Group, New York, NY, USA.
- [20] Luis A. García-Escudero, Alfonso Gordaliza, Carlos Matrán y Agustín Mayo-Isacar, *A General Trimming Approach to Robust Cluster Analysis*, The Annals of Statistics 2008, Vol. 36, Núm. 3, pp 1324–1345. Disponible [aquí](#) (Junio 2020)
- [21] Adrián Manzano Santos, *Trabajo de Fin de Grado de Ingeniería Informática de Servicio y Aplicaciones: construcción de un sistema de asistencia al operador de aprendizaje semi-autónomo*. 2020