Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABAJO FIN DE GRADO
GRAO EN CIENCIA E ENXEÑARÍA DE DATOS

Cluster difuso de series de tiempo: El papel clave de la métrica seleccionada.

Estudiante: Xoel Montes Varela

Dirección: José Antonio Vilar Fernández

A Coruña, noviembre de 2024.

A mi familia y amigos, por su apoyo incondicional

Agradecimientos

Quiero agradecer a todas las personas que, de una forma u otra, han colaborado en la realización de este proyecto. A mi tutor, José, por su valiosa guía y dedicación; a los profesores Manuel Febrero Bande y Manuel Oviedo de la Fuente por permitirme utilizar los datos meteorológicos de su librería; a la profesora Rebeca Pélaez por haberme cedido los datos de demanda eléctrica; a mis amigos, tanto los de siempre como los recién llegados, por su apoyo constante y ánimo; y, de manera especial, a mi familia, por estar siempre a mi lado.

Resumen

El análisis de cluster, o clustering, de series temporales es un tema de gran interés con aplicaciones en muchas disciplinas. El carácter dinámico de las series añade complejidad al clustering, ya que no es trivial establecer un criterio de disimilitud entre series de tiempo. El presente proyecto pretende mostrar la importancia de usar una métrica adecuada para desarrollar clustering difuso, poniendo especial atención al problema de agrupar series estacionarias generadas por modelos estocásticos similares. En concreto, se analiza el comportamiento del algoritmo fuzzy C-medoids empleando un rango de métricas específicamente diseñadas para comparar realizaciones de series. El estudio proporciona una visión general del problema, destacando las dificultades inherentes al clustering de series y permitiendo concluir sobre las fortalezas y debilidades de las diferentes métricas examinadas mediante experimentos con datos simulados. El procedimiento de clustering se ejecuta también sobre conjuntos de datos reales para ilustrar su interés y aplicabilidad.

Abstract

The clustering of time series is a topic of great interest with applications in many disciplines. The dynamic nature of time series adds complexity to clustering, as establishing a dissimilarity criterion between time series is not trivial. This project aims to show the importance of using an appropriate metric to develop fuzzy clustering, with special attention to the problem of grouping stationary series generated by similar stochastic models. In particular, the behavior of the fuzzy C-medoids algorithm is analyzed using a range of metrics specifically designed to compare series realizations. The study provides an overview of the problem, highlighting the inherent difficulties in clustering series and allowing conclusions about the strengths and weaknesses of the different metrics examined through experiments with simulated data. The clustering procedure is also applied to real datasets to illustrate its interest and applicability.

Palabras clave:

- Análisis cluster
- Series temporales
- Clustering difuso
- Métricas de disimilitud
- Series Estacionarias
- Modelos Estocásticos
- Algoritmo fuzzy C-medoids

Keywords:

- Clustering
- Time series
- Fuzzy clustering
- Dissimilarity metrics
- Stationary Series
- Stochastic Models
- Fuzzy C-medoids algorithm

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Gestión del proyecto

1.1 Herramientas y tecnologías

A continuación se describen las diferentes herramientas y tecnologías más necesarias y destacables para el desarrollo del proyecto.

- **R Studio**

  Entorno de desarrollo integrado (IDE) para el lenguaje de programación R, diseñado para facilitar la programación, visualización de datos y análisis estadístico [1].

- **Visual Studio Code (VS Code)**

  Editor de código fuente, desarrollado por Microsoft, que soporta múltiples lenguajes de programación con el cual editar, depurar y compilar código y, finalmente, publicar una aplicación [2].

- **DBeaver**

  Aplicación de software cliente de SQL y una herramienta de administración que soporta una amplia variedad de sistemas de bases de datos. DBeaver facilita la conexión, consulta y administración de bases de datos, ofreciendo una interfaz visual para desarrollar y realizar análisis sobre los datos almacenados [3].

- **GeoServer**

  Plataforma de código abierto que facilita la publicación y edición de datos geoespaciales. Permite compartir y visualizar datos en formatos estándar como WMS, WFS y WCS, siendo ideal para la creación de mapas interactivos y la gestión de información geográfica [4].

- **Leaflet**

  Biblioteca de JavaScript que facilita la creación de mapas interactivos en aplicaciones web [5].

- **PostgreSQL con la extensión PostGIS**

  Sistema de gestión de bases de datos relacional que, junto a la extensión PostGIS, permite trabajar con datos geoespaciales. PostGIS [6] convierte a PostgreSQL [7] en una base de datos ideal para el almacenamiento, procesamiento y análisis de datos geográficos.

1.2 Metodología

Para llevar a cabo este proyecto, se ha optado por seguir una metodología de tipo cascada, basada en el enfoque secuencial de siete tareas.

- **Revisión**: En primer lugar, se llevó a cabo una breve revisión del estado del arte sobre las medidas de disimilitud entre series temporales, con especial énfasis en aquellas basadas en características extraídas de las realizaciones seriales sujetas a cluster.

- **Implementación en R**: Posteriormente, se procedió a la implementación de las métricas de distancia seleccionadas, así como del algoritmo fuzzy C-medoids, utilizando el lenguaje de programación R.

- **Estudio de simulación**: Completada la fase de programación, se identificaron y seleccionaron algunos escenarios de generación de datos útiles para examinar y comparar el comportamiento cluster con las distintas métricas bajo simulaciones controladas.

- **Análisis**: Los resultados de los experimentos de simulación se emplearon para evaluar la eficacia del análisis cluster con un criterio preestablecido, lo que permitió examinar la efectividad y eficiencia de cada métrica empleada en el estudio de simulación.

- **Datos reales**: Aplicación del algoritmo a conjuntos de datos reales, lo que permitió ilustrar la utilidad en la práctica de los procedimientos y alcanzar una validación empírica de los resultados obtenidos.

- **Herramienta de visualización**: Desarrollo de una herramienta para facilitar la visualización de los resultados generados por el algoritmo en uno de los escenarios con datos reales.

- **Memoria**: Documentación de todo el proceso, incluyendo información detallada sobre los objetivos planteados, los métodos aplicados, los resultados obtenidos, las conclusiones alcanzadas, y otros aspectos relevantes. Esta tarea se desarrolla desde el comienzo

hasta la finalización del proyecto, lo cual asegura una documentación completa y precisa del trabajo realizado.

Se mantuvieron reuniones semanales con el tutor académico como parte del seguimiento del proyecto. Estas sesiones fueron cruciales debido a la dependencia entre las tareas, ya que cualquier error no resuelto podría propagarse de una etapa a la siguiente, afectando el desarrollo hasta la finalización del proyecto.

Este enfoque estructurado aseguró que cada etapa del proyecto estuviera bien definida y finalizada antes de avanzar a la siguiente, garantizando la coherencia y calidad del trabajo final.

1.3 Planificación

Para gestionar de manera adecuada un proyecto de esta magnitud, con una carga considerable de trabajo, es importante realizar una planificación inicial para abordar la completitud del mismo de forma eficiente.

En este sentido, se diseñó un diagrama de Gantt, mostrado en la Figura 1.1, con el fin de ofrecer una representación visual clara de las distintas tareas involucradas a lo largo de toda la duración del proyecto.

Cada una de estas tareas cuenta con un plazo estimado de inicio y finalización, determinados en función de su complejidad y del tiempo necesario para su desarrollo. Es importante señalar que las tareas no son etapas aisladas, sino que se desarrollan de forma secuencial, lo que garantiza una evolución continua del sistema.

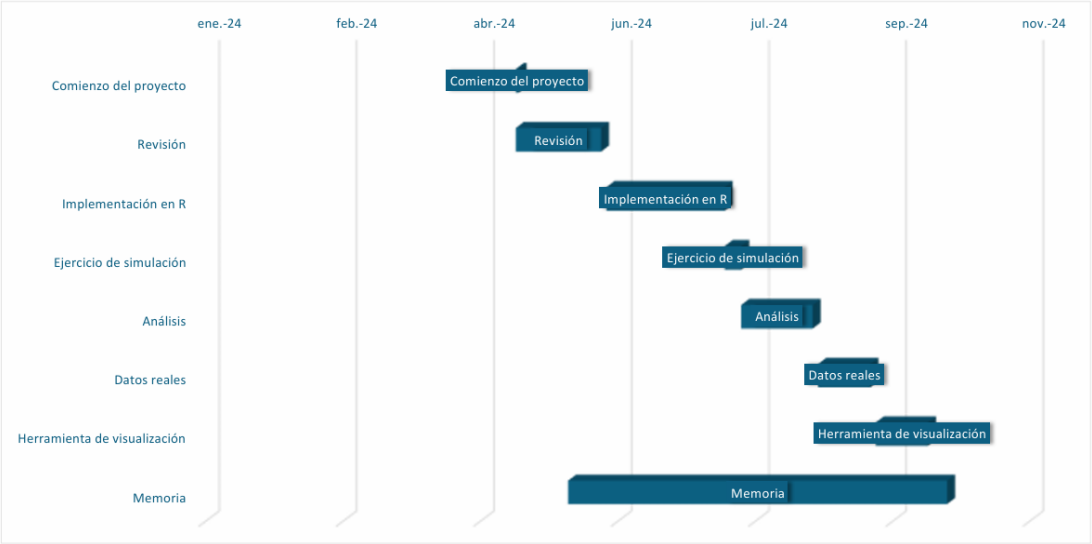


Figura 1.1: Diagrama de Gantt del proyecto.

1.4 Estimación de costes

La estimación de costes permite prever y gestionar el presupuesto necesario para completar las distintas etapas del proyecto de forma eficiente. En esta sección se realiza una estimación de los costes asociados al proyecto, considerando tanto los recursos materiales como humanos.

En cuanto a los recursos materiales, incluimos un ordenador propio, del que ya se disponía, un equipo de sobremesa, propiedad del grupo de investigación en Modelización, Optimización e Inferencia Estadística (MODES) de la UDC, y las herramientas mencionadas en la Sección 1.1, que tampoco suponen ningún tipo de coste, ya que se trata de soluciones abiertas.

En relación con los recursos humanos, la Tabla 1.1 muestra los costes estimados de acuerdo con los salarios estipulados según Glassdoor para cada puesto [8, 9].

Rol	**Salario/hora**	**Horas trabajadas**	**Coste total**
Científico de datos junior	12,4€	300	3720€
Catedrático universitario	24,8€	30	744€
Total recursos humanos			**4464€**

Tabla 1.1: Costes recursos humanos

En total, los costes asociados al proyecto se aproximan a los 4.500€.

Capítulo 2

Introducción

Los sofisticados sistemas de recogida y procesado de datos existentes en la actualidad permiten disponer a menudo de grandes bases de datos, incluyendo datos de elevada complejidad como es el caso de realizaciones de series temporales o datos funcionales. Extraer conocimiento de un conjunto grande de series de tiempo observadas es, de hecho, una tarea habitual en numerosas disciplinas, tales como la economía, la meteorología, la medicina, el aprendizaje automático y la ingeniería, entre otras muchas. Desde luego, el análisis de series temporales proporciona herramientas para modelizar las series, pero frecuentemente el objetivo no es el análisis individual de cada serie, sino disponer de procedimientos computacionalmente eficientes para identificar grupos de series similares y obtener así una clasificación de las mismas. El análisis cluster de series temporales persigue este objetivo: agrupar un conjunto de series temporales no etiquetadas en grupos homogéneos, de tal manera que observaciones de un mismo grupo compartan características de interés (patrones temporales, estructuras de dependencia, anomalías,...), facilitando así la identificación de comportamientos comunes, la segmentación de datos y la extracción de conocimiento útil.

Es muy sencillo identificar aplicaciones específicas donde el cluster de series es relevante. A título ilustrativo, algunos ejemplos serían: determinar productos con patrones de venta similares, identificar perfiles de usuarios según su comportamiento de navegación en un sitio web, clasificar empresas en base a las series de precios de sus acciones en bolsa, agrupar países con similar crecimiento poblacional o la identificación de regiones con registros de temperatura semejantes. En todas estas problemáticas, el interés en evaluar el grado de similitud entre series temporales y detectar grupos homogéneos surge de manera natural y resulta de gran valor para la toma de decisiones y el análisis estratégico.

El enorme rango de aplicaciones y el creciente interés de investigadores de diversas disciplinas justifica que el clustering de series temporales se haya convertido en un área de investigación muy activa en la actualidad. En las últimas tres décadas, ha habido una cantidad significativa de contribuciones en este campo. Una visión general exhaustiva del clustering

de series temporales ha sido proporcionada por Liao [10], y más recientemente por Fu [11], Rani-Sikka [12], y Caiado *et al.* [13], abarcando avances recientes, referencias clave y áreas de aplicación específicas.

Al tratar con series de tiempo, el carácter dinámico de los registros dificulta el proceso de clasificación, generando desafíos adicionales al clustering estándar de datos "estáticos". Por ejemplo, es bien conocido el papel relevante en análisis cluster de la métrica seleccionada para medir distancia entre objetos, toda vez que diferentes criterios de distancia pueden conducir a distintos agrupamientos. Este problema se acentúa con series de tiempo porque no es trivial definir una distancia entre objetos que evolucionan en el tiempo. En ocasiones, el interés radica en detectar diferencias entre los perfiles geométricos de las series (diferencias "en forma") y, en tal caso, métricas estándar de tipo Minkowski habitualmente utilizadas con datos estáticos son válidas. Otras veces se desea discriminar entre las estructuras de dependencia subyacentes a realizaciones seriales de procesos estacionarios (diferencias "en estructura") y en este escenario se requieren nuevas vías para evaluar discrepancia entre series. En la literatura se han propuesto diferentes criterios para medir similitud y distancia entre un par de series y uno de los principales objetivos de este trabajo es ilustrar el papel crucial que la métrica empleada tiene sobre la solución cluster. El caso más complejo de tratar con series estacionarias recibe particular atención.

Existen diversas técnicas de clustering, cada una con sus particularidades y ventajas. Dos enfoques importantes son el clustering jerárquico y el clustering particional.

El clustering jerárquico construye una estructura de grupos mediante un proceso aglomerativo o divisivo. Los métodos aglomerativos comienzan con un número de grupos igual al número de observaciones, es decir, cada observación inicia en su propio cluster. Luego, en cada etapa, los dos grupos más cercanos se fusionan, y este proceso se repite hasta que todos los datos están agrupados en un único cluster. Este enfoque permite una representación visual a través de un dendrograma, que muestra las relaciones jerárquicas entre los datos y facilita la identificación de grupos en diferentes niveles de granularidad. Por otro lado, los métodos divisivos comienzan con un único cluster que contiene todas las observaciones y, en cada etapa, dividen un cluster en dos. Este proceso continúa hasta que cada observación está en su propio cluster o se alcanza el número deseado de grupos.

En contraste, el clustering particional divide directamente los datos en un conjunto predefinido de K grupos. Un ejemplo clásico de esta técnica es el algoritmo K-Means, que agrupa las observaciones minimizando la suma de las distancias al centroide de cada clúster. Este método es eficiente y sencillo de implementar, pero requiere que el número de grupos sea especificado de antemano, lo que puede ser una limitación si no se conoce esta información.

Los algoritmos de análisis cluster pueden agruparse en dos categorías conocidas genéricamente como clustering "hard" y clustering "soft". Los algoritmos en la categoría de cluster

“hard” asignan cada objeto a un único grupo de manera exclusiva, es decir, cada observación pertenece únicamente a un grupo sin ambigüedad. Este enfoque es intuitivo y sencillo de interpretar; sin embargo, frecuentemente las bases de datos comprenden subconjuntos mal delineados que no pueden fragmentarse adecuadamente de esta manera tan nítida. En ocasiones, existe cierto grado de solapamiento entre los grupos y, en tales casos, resulta mucho más informativo el uso de algoritmos de agrupación ”soft”, diseñados para que cada objeto pertenezca a todos los grupos con diferentes grados de membresía [14]. Una técnica representativa del clustering ”soft” es el algoritmo Fuzzy K-Means, donde cada objeto tiene un grado de pertenencia a cada grupo que varía entre 0 y 1. Este enfoque es más flexible y puede capturar la ambigüedad y la incertidumbre en los datos, proporcionando una representación más rica de las relaciones entre los datos. En contrapartida, una solución ”soft” suele resultar más compleja de interpretar y requiere una gestión adecuada de los grados de pertenencia.

El presente estudio se focaliza en analizar el comportamiento con series de tiempo de un algoritmo cluster particional dentro de la categoría soft, a saber, el algoritmo C-Medoids en su versión difusa, conocido como Fuzzy C-Medoids. A diferencia del Fuzzy K-Means, que calcula los centroides como un promedio ponderado de los datos dentro de cada grupo, el Fuzzy C-Medoids utiliza observaciones reales del conjunto de datos para definir los prototipos de los grupos identificados.

Se discuten los resultados generados por el algoritmo Fuzzy C-Medoids en diversos escenarios simulados artificialmente y utilizando un abanico de métricas específicamente diseñadas para tratar con series temporales. Este enfoque, donde se conoce la estructura real de grupos subyacente (“ground truth”), permite analizar cómo diferentes métricas pueden influir en la estructura de los grupos identificados. Más aún, se diseñan escenarios simulados con diferente grado de cohesión y separación de los grupos y diferentes longitudes de las series observadas, lo que permite evaluar la estabilidad y consistencia de los procedimientos cluster.

Sobre la base de los resultados de estos experimentos, se obtienen conclusiones sobre las fortalezas y debilidades de cada métrica en el contexto del clustering de series temporales. Esto no solo proporciona una comprensión más profunda sobre cómo seleccionar la métrica adecuada para distintos tipos de datos temporales, sino que también ilustra las circunstancias en las que el algoritmo Fuzzy C-Medoids puede ofrecer ventajas significativas frente a otros métodos de clustering.

Por último, se lleva a cabo un estudio utilizando datos reales para evaluar la aplicabilidad y el desempeño del algoritmo en un contexto práctico. Este análisis empírico permitirá comparar los resultados obtenidos en escenarios simulados con aquellos derivados de datos del mundo real, proporcionando una validación adicional de las conclusiones teóricas previamente obtenidas.

Capítulo 3

Algoritmo Fuzzy C-Medoids

El algoritmo Fuzzy C-Medoids, como se ha mencionado, es una variante del Fuzzy K-Means. En lugar de calcular los centroides como un promedio ponderado de cada grupo, los prototipos de los grupos son seleccionados de entre las propias observaciones reales del conjunto de datos y se denominan medoides.

Conviene subrayar que el algoritmo K-Means no es una elección adecuada para tratar con series de tiempo porque los centroides, calculados como promedios ponderados, podrían no caracterizar adecuadamente los protopipos dinámicos de los grupos. En efecto, el promedio de los objetos de un grupo, ya sea tratando directamente con los datos en crudo o reemplazando las series observadas por vectores de características extraídas (ver Capítulo 4 para más detalles), no tiene por qué conducir a una estructura de serie temporal bien definida. Por ejemplo, si la base de datos está formada por realizaciones de modelos autorregresivos integrados de medias móviles (ARIMA), entonces no hay garantías de que los centroides así calculados representen a uno de estos modelos. De hecho, el centroide resultante podría no satisfacer las restricciones requeridas a los coeficientes que definen estos modelos. De esta manera, los centroides pueden ser series temporales "ficticias", lo que conlleva serios inconvenientes. Primero, la distancia entre las series temporales observadas y los centroides podría no estar adecuadamente definida si la distancia se ha definido asumiendo que ambos objetos son modelos ARIMA. Por otro lado, la agrupación de series temporales a menudo tiene como objetivo encontrar series temporales "representativas" para cada grupo, es decir, un conjunto de patrones que resuman las diferentes dinámicas subyacentes, y nuevamente esto no está garantizado.

Una forma natural de superar estos inconvenientes es utilizar un algoritmo basado en medoides, donde los prototipos están restringidos a ser elegidos entre los puntos de datos reales. Este enfoque convierte al algoritmo K-Medoids en una herramienta valiosa para analizar y comprender estructuras complejas de datos temporales.

Por otro lado, un enfoque de partición difusa ("fuzzy") podría proporcionar una solución

mucho más atractiva e interpretable tratando con series temporales. Como ya se ha indicado, una partición fuzzy permite el solapamiento de grupos. Más precisamente, asocia a cada objeto un vector de etiquetas con grados de membresía a cada grupo moviéndose de cero a uno. De esta manera, una serie con un patrón de comportamiento próximo a varios grupos podría ser asignada a todos ellos con diferentes niveles de confianza, pero sin necesidad de forzar su asignación a solo uno de esos grupos. Esto podría estar perfectamente justificado por cambios en la dinámica de las series o por la existencia de series temporales que exhiben patrones con características intermedias respecto a varios grupos [15, 16]. Si una métrica adecuada es capaz de capturar las diferencias entre los patrones dinámicos subyacentes, entonces un algoritmo fuzzy basado en dicha distancia debería ganar versatilidad para construir los prototipos y, por lo tanto, obtener una mejor caracterización del patrón temporal de las series (véase la discusión en [15]).

En resumen, el algoritmo Fuzzy C-Medoids se presenta como una opción muy adecuada en nuestro marco de trabajo debido al uso de medoides en lugar de centroides y a su capacidad para manejar la pertenencia solapada de las series a múltiples grupos. A continuación se formaliza el algoritmo en términos del problema de optimización que lo define.

3.1 Problema de optimización

Sea $X = \{x_1, \ldots, x_n\}$ un conjunto de n vectores de longitud T de números reales. A lo largo de esta memoria, los elementos de X representarán realizaciones de longitud T de las series sujetas a cluster o, alternativamente, vectores de igual longitud de características extraídas de las mismas.

El algoritmo Fuzzy C-Medoids surge como solución del problema de optimización que sigue:

Determinar un subconjunto de C elementos en X (medoides), $\mathcal{C} = \{\hat{x}_1, \ldots, \hat{x}_C\} \subset X$, y una matriz $\mathcal{U} = (u_{ij})$ de dimensión $n \times C$ (matriz de grados de membresía), tales que $\mathcal{C}$ y $\mathcal{U}$ minimicen la función objetivo χ_d dada por:

$$\chi_d(\mathcal{C}, \mathcal{U}) = \sum_{i=1}^{n} \sum_{c=1}^{C} u_{ic}^m \, d(x_i, \hat{x}_c) \tag{3.1}$$

sujeto a las restricciones:

$$\sum_{c=1}^{C} u_{ic} = 1 \quad \text{and} \quad u_{ic} \geq 0, \text{ para todo } i, c,$$

donde $d(x_i, \hat{x}_c)$ denota la distancia entre el punto de datos x_i y el medoide $\hat{x}_c$ asociado al

cluster c, para una medida de distancia $d(\cdot, \cdot)$ prefijada de antemano; y $m > 1$ es un parámetro que controla cómo de difuso puede ser el particionado.

Mientras que con datos estáticos, la métrica seleccionada $d(a, b)$ suele ser la distancia Euclídea (o la de Mahalanobis) entre a y b, con series temporales existen muchas otras opciones como se discutirá en el Capítulo 4.

En definitiva, dado que $\mathcal{C}$ representa el conjunto de medoides y $\mathcal{U}$ los grados de membresía o de pertenencia de los objetos a los grupos, el objetivo del algoritmo Fuzzy C-Medoids es determinar una partición difusa en C grupos de manera que se minimice la suma final de distancias entre puntos y medoides ponderadas por una potencia prefijada m de los correspondientes grados de membresía.

3.2 Coeficiente de nivel de solapamiento m

El parámetro $m > 1$ determina el grado de solapamiento (coeficiente de borrosidad -"fuzziness"-) que se introduce en el procedimiento de particionado. Valores de m muy cercanos a 1 tienden a una versión "hard" del procedimiento ya que cada objeto presentará un alto grado de membresía para un solo grupo y valores próximos a cero para el resto. Se tendrá así una solución cluster muy limpia, sin solapamientos. A medida que el valor de m aumenta, los límites entre los grupos se vuelven más difusos, tendiendo a que las series presenten niveles de membresía muy similares para todos los grupos y generando una partición muy poco rígida. Es preciso por tanto seleccionar un valor apropiado para el valor del coeficiente de solapamiento m, el cual típicamente oscila entre 1.5 y 2.5.

3.3 Cómputo iterativo de medoides y grados de membresía

La resolución del problema de optimización planteado en (3.1) y que define el algoritmo Fuzzy C-Medoids se realiza en base a un procedimiento iterativo que proporciona secuencialmente los grados de pertenencia y los medoides. Específicamente, fijado un conjunto de medoides, los grados de membresía óptimos se pueden aproximar de acuerdo a [17]:

$$u_{ic} = \left(\sum_{c'=1}^{C} \left(\frac{d(x_i, \hat{x}_c)}{d(x_i, \hat{x}_{c'})} \right)^{\frac{1}{m-1}} \right)^{-1} \quad \text{para } i = 1, \ldots, n \text{ y } c = 1, \ldots, C. \tag{3.2}$$

Los valores de $\mathcal{U}$ obtenidos de acuerdo a (3.2) se emplean entonces para actualizar los medoides minimizando la función objetivo χ_d dada en (3.1) y este proceso en dos pasos se ejecuta iterarivamente hasta que los medoides no cambian o se alcanza un número máximo de iteraciones prefijado de antemano.

Una buena práctica es seleccionar los medoides iniciales utilizando métodos robustos como el algoritmo PAM (Partitioning Around Medoids) [18]. Este enfoque de clustering hard proporciona una buena aproximación inicial de los medoides. La selección cuidadosa de los medoides iniciales facilita una convergencia más rápida y eficiente del algoritmo, contribuyendo a obtener agrupamientos más estables y representativos de los datos reales. Alternativamente, también se podría optar por seleccionar aleatoriamente los medoides iniciales.

3.4 Pseudocódigo

El algoritmo Fuzzy C-Medoids, en adelante FCMdC, se implementa de acuerdo con la descripción proporcionada en el Algoritmo 1.

Algorithm 1: Algoritmo Fuzzy C-Medoids (FCMdC)

Input: Fix C, m, max.iter
Initialize: iter $\leftarrow 0$
Fijar los medoides iniciales $\hat{\mathcal{C}} = \{\hat{x}_1, \ldots, \hat{x}_C\}$; // aleatorio o PAM
repeat
  Establecer $\hat{\mathcal{C}}_{\text{OLD}} = \hat{\mathcal{C}}$; // guardar los medoides actuales
  Calcular u_{ic}, $i = 1, \ldots, n$, $c = 1, \ldots, C$; // (3.2)
  for $c \in \{1, \ldots, C\}$ **do**
    determinar el índice $j_c \in \{1, \ldots, n\}$ satisfaciendo:

$$j_c = \arg\min \sum_{i=1}^{n} \sum_{c=1}^{C} u_{ic}^m \, d(x_i, \hat{x}_c)$$

    return $\hat{x}_c = x_{j_c}$, for $c = 1, \ldots, C$; // Actualizar medoides
  iter $\leftarrow$ iter $+ 1$
  $\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_C\}$
until $\hat{\mathcal{C}}_{OLD} = \hat{\mathcal{C}}$ ***or*** *iter* $=$ *max.iter*;

3.5 Ejemplo ilustrativo con datos estáticos

Para observar y entender el comportamiento del algoritmo FCMdC, se aplica sobre la bien conocida base de datos *Iris*. Este conjunto de datos contiene medidas en centímetros de las variables longitud y ancho del sépalo y longitud y ancho del pétalo para 50 flores de cada una de 3 especies de iris, a saber *Iris setosa*, *Iris versicolor* e *Iris virginica* [19, 20]. Este conjunto de datos ha sido ampliamente utilizado en la literatura para examinar técnicas de clasificación y análisis de datos debido a su estructura sencilla y fácilmente interpretable [21].

La ejecución del algoritmo se lleva a cabo con la función `FKM.med`, accesible a través de la librería **fclust**, desarrollada por Giordani *et al.* [22, 23].

Específicamente, se realiza un pequeño ejercicio de clustering basado en un subconjunto de *Iris* conformado por los registros de las variables "Sepal.Length" y "Sepal.Width" para las especies *Iris setosa* e *Iris versicolor*. El objetivo es comprobar si estas dos variables están lo suficientemente correlacionadas con la especie como para poder discriminar entre ambas especies basándonos en sus valores.

Dado que `FKM.med` es un algoritmo particional de clasificación no supervisada, se requiere especificar el número de grupos (argumento k), que en este ejercicio ilustrativo es dos. A modo ilustrativo, la asignación cluster y los grados de membresía para nueve de las flores producidos por el algoritmo con $k = 2$ y para dos valores diferentes del parámetro m controlando el nivel de solapamiento se muestran en la Tabla 3.1.

(a) $m = 1.5$				(b) $m = 2$			
id	Clus 1	Clus 2	Especie	id	Clus 1	Clus 2	Especie
7	0.9948	0.0052	setosa	7	0.9286	0.0714	setosa
8	1	0	setosa	8	0.9921	0.0079	setosa
9	0.9463	0.0537	setosa	9	0.8312	0.1688	setosa
10	0.9936	0.0064	setosa	10	0.9615	0.0385	setosa
50	0.9999	1e-04	setosa	50	1	0	setosa
78	0.0262	0.9738	versicolor	78	0.1437	0.8563	versicolor
79	0	1	versicolor	79	0	1	versicolor
80	0.0247	0.9753	versicolor	80	0.1552	0.8448	versicolor
85	0.5721	0.4279	versicolor	85	0.5968	0.4032	versicolor

Tabla 3.1: Asignación cluster y grados de membresía obtenidos para 9 flores específicas del subconjunto considerado de *Iris* tras ejecutar FCMdC.

Los resultados para $m = 1.5$ en la Tabla 3.1(a) muestran niveles de membresía muy elevados para un cluster específico y próximos a cero para el otro para todas las observaciones con excepción de la flor con id$= 85$. Así pues, cabe concluir que, excepto esta última, todas las otras flores de esta muestra han sido ubicadas en un único cluster sin ambigüedad. Más aún, a la vista de la etiqueta "Especie", el Cluster 1 agrupa las flores *setosa* y el Cluster 2 las *versicolor*. La flor con id$= 85$ presenta niveles de pertenencia similares, 0.5721 al Cluster 1 y 0.4279 al Cluster 2, indicando un elevado nivel de incertidumbre pues ninguna de las dos especies sería totalmente descartable para esta flor. La nube de puntos con los registros de las dos variables consideradas, "Sepal.Length" y "Sepal.Width", en la Figura 3.1 corrobora esta

cicunstancia ya que esta observación se encuentra en una posición intermedia entre los dos grupos, justificando así los grados de membresía obtenidos. Nótese que las flores con id 8 y 79 presentan grados de membresía exactamente iguales a 1 para uno de los clusters, concluyendo así que son los medoides de sus respectivos grupos.

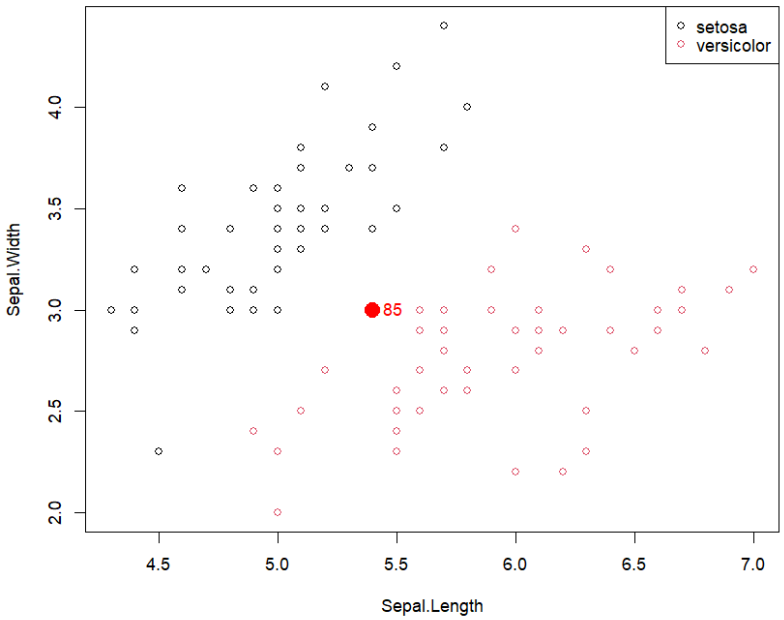


Figura 3.1: Gráfico de dispersión de las variables "Sepal.Length" y "Sepal.Width" para las flores de las especies *setosa* y *versicolor* en el conjunto de datos *Iris*.

Las mismas conclusiones se derivan de los resultados para $m = 2$ en la Tabla 3.1(b), aunque aquí los niveles de membresía no están tan cerca de 0 y 1, reflejando un leve incremento de la incertidumbre en la asignación de las flores a una única especie. Nótese que distintos valores del parámetro m pueden conducir a diferentes medoides.

Capítulo 4

Midiendo disimilitud entre series de tiempo

Por construcción, un procedimiento de análisis cluster distribuye las observaciones en grupos caracterizados por contener datos "semejantes", pero "diferentes" a los de los otros grupos. Por consiguiente, establecer adecuadamente el concepto de similitud (disimilitud) entre observaciones es crucial para alcanzar una solución interpretable y acorde con el propósito del agrupamiento.

En el clustering convencional con datos "estáticos", la selección de la métrica para evaluar disimilitud viene fundamentalmente condicionada por el tipo de datos (escala, categóricos, mixtos,...). Con datos numéricos es habitual considerar una distancia de tipo Minkowski (basada en la norma L_q), frecuentemente la distancia Euclídea ($q = 2$) o la de Manhattan ($q = 1$).

Cuando se trabaja con realizaciones de series de tiempo numéricas, cada objeto es una secuencia de registros evaluados en el tiempo que puede incluir el efecto de componentes varias como tendencia, ciclos, componente estacional y el impacto del ruido aleatorio. Se trata en consecuencia de datos complejos de naturaleza dinámica y medir el grado de similitud o de discrepancia entre ellos no es en absoluto una cuestión trivial.

En este capítulo se establece una primera clasificación de medidas de distancia entre series que atiende a si el cluster persigue diferenciar trayectorias o modelos generadores y se introduce un abanico (no exhaustivo) de métricas propuesto en la literatura [24, 25, 26], enfatizando algunas de sus propiedades más relevantes y motivando el interés en las mismas.

4.1 Forma vs estructura

Una primera cuestión a dilucidar al abordar clustering de series es decidir si el interés radica en discriminar las series en base a los perfiles geométricos de sus trayectorias (diferencias en *forma*) o de acuerdo con los modelos de dependencia que las han generado (diferencias

en *estructura*). La elección de uno u otro enfoque puede influir significativamente en los resultados y, obviamente, la interpretación de los grupos resultantes será diferente. El enfoque adecuado dependerá básicamente de la naturaleza de los datos y del objetivo específico del análisis [27, 28].

La disimilitud basada en forma tiene particular interés con series dominadas por sus patrones estacionales y de tendencia, y también con series cortas para las que no es factible inferir propiedades de sus estructuras de dependencia con la precisión deseable. La disimilitud basada en forma se caracteriza por realizar comparaciones locales, evaluando las diferencias punto por punto a lo largo del rango temporal. Un ejemplo clásico de una medida de disimilitud basada en forma es la distancia Euclídea.

En contraste, la disimilitud basada en estructura se focaliza en comparar los modelos de dependencia que subyacen a las realizaciones. Este enfoque es adecuado cuando se desean comparar conductas globales y a más largo plazo y, en particular, con series de tiempo estacionarias (series con media y varianza constantes en el tiempo y covarianza entre registros dependiente sólo del tiempo transcurrido entre ellos). Estas métricas restan importancia a discrepancias puntuales y se centran en identificar diferencias entre los patrones de dependencia que gobiernan el comportamiento global de las series.

Para ilustrar las diferencias entre ambos criterios de distancia, se han simulado 9 series a partir de tres patrones generadores distintos, digamos G1, G2 y G3. Las realizaciones simuladas, tres de cada patrón, se muestran en la Figura 4.1, empleando líneas de distinto tipo y color para cada grupo.

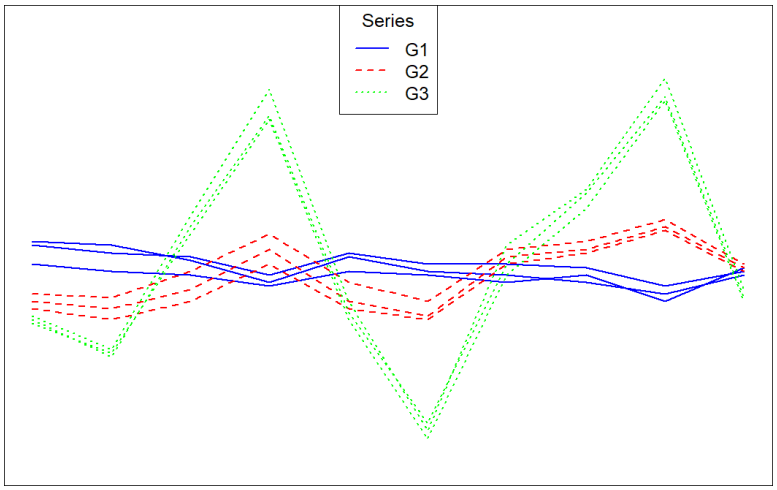


Figura 4.1: Ejemplo simulado para ilustrar diferencias de disimilitud en forma y en estructura.

Atendiendo a un criterio de disimilitud basado en forma, los grupos G1 y G2 son los más similares. En efecto, sus trayectorias están más cercanas ya que se enmarcan en una estrecha franja que no incluye a la mayoría de registros de las series de G3. Ahora bien, si se compa-

ran las autocorrelaciones parciales de retardo tres de las series, es decir, las conductas de las observaciones en un instante t a la luz de sus valores previos en $t-1$, $t-2$ y $t-3$, excluyendo la influencia de los valores intermedios en cada retardo, entonces G1 y G3 son los grupos más similares. Este segundo enfoque compara estadísticos extraídos de las series que miden el grado de dependencial lineal subyacente y, por tanto, el principio es evaluar distancia en estructura.

Este análisis, basado en una simple inspección visual de los perfiles de las series, queda refrendado con las soluciones cluster mostradas en la Figura 4.2. Estos agrupamientos resultan al realizar cluster hard con el algoritmo C-Medoids, fijando $C = 2$ y usando la distancia Euclídea entre: (i) los registros en crudo (Figura 4.2(a)), y (ii) las estimaciones de los coeficientes de autocorrelaciones parciales de retardo tres (Figura 4.2(b)).

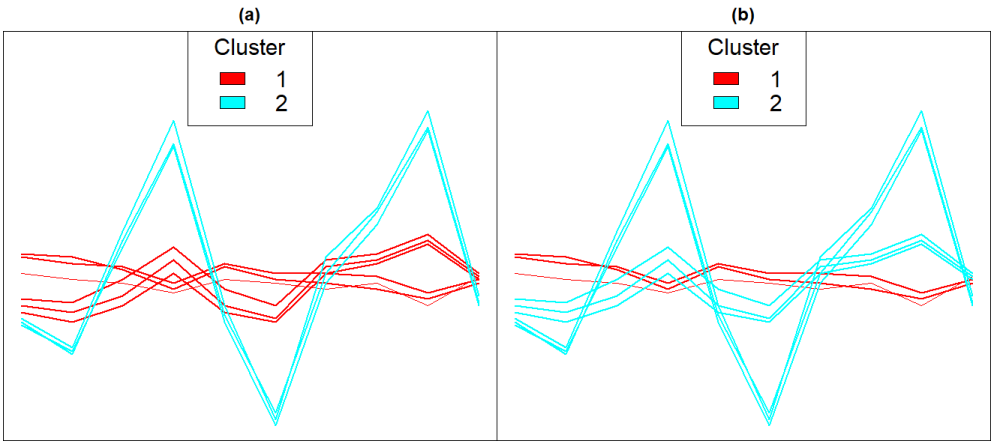


Figura 4.2: Solución cluster con el conjunto de 9 series del ejemplo simulado basada en el algoritmo C-Medoids, con $C = 2$, usando la distancia Euclídea entre observaciones (a) y entre autocorrelaciones parciales estimadas (b).

4.2 Medidas de distancia

Siguiendo la monografía de Caiado *et al.* [13] (ver también D'Urso *et al.* [29]), desde un punto de vista metodológico, los métodos cluster de series temporales pueden clasificarse en las siguientes categorías: cluster basado en observaciones en crudo, cluster basado en características y cluster basado en modelos. Cada uno de estos enfoques supone evaluar disimilitud entres series de forma distinta. En esta sección se introducen y describen brevemente algunos criterios relevantes propuestos en la literatura dentro de cada una de estas categorías. El objetivo no es proporcionar una revisión exhaustiva de métricas, sino un subconjunto significativo de ellas, profundizando en sus principales propiedades y limitaciones cuando sea el caso.

En lo que sigue, a menos que se especifique lo contrario, $X_T = (X_1, \ldots, X_T)$ denota una realización específica de longitud T de un proceso $X = \{X_t\}_{t\in\mathbb{Z}}$ tomando valores en $\mathbb{R}$.

Ejemplo 4.2.1 *Se generan realizaciones de longitud $T = 55$, $X_T = (X_1, \ldots, X_T)$, $Y_T = (Y_1, \ldots, Y_T)$ y $W_T = (W_1, \ldots, W_T)$, de tres procesos diferentes $X = \{X_t\}_{t\in\mathbb{Z}}$, $Y = \{Y_t\}_{t\in\mathbb{Z}}$ y $W = \{W_t\}_{t\in\mathbb{Z}}$, respectivamente. Los gráficos de línea de estas realizaciones se muestran en la Figura 4.3. En las secciones que siguen se evaluará la distancia entre cada par de estas realizaciones de acuerdo a diferentes métricas.*

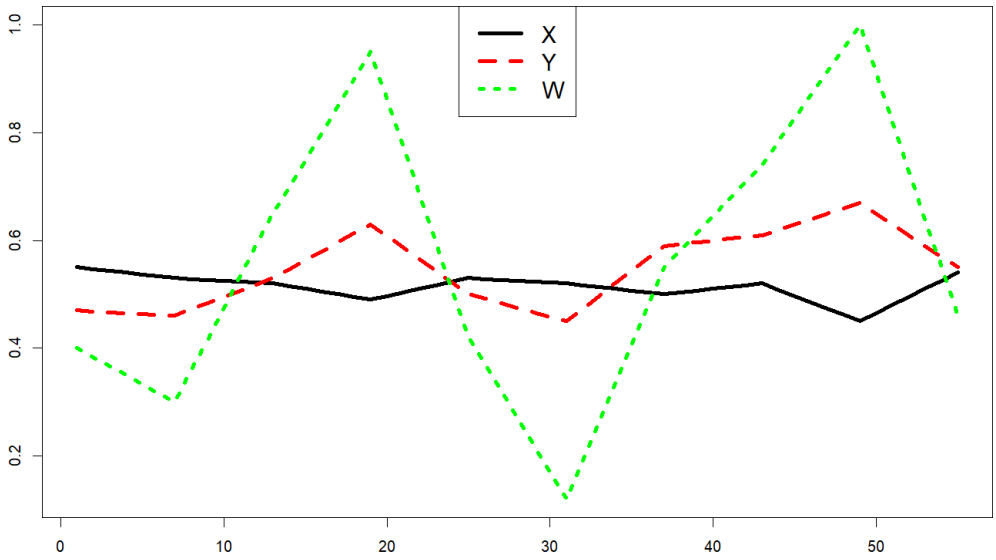


Figura 4.3: Realizaciones de longitud $T = 55$ de tres series temporales.

4.2.1 Cluster basado en observaciones en crudo

Si el objetivo del análisis cluster es agrupar series bajo el principio de similitud en forma, una distancia convencional evaluando discrepancia entre las observaciones en crudo de las realizaciones de las series podría deparar resultados satisfactorios. A continuación, se presentan algunas de estas métricas.

- **Distancia de Minkowski**

La distancia de Minkowski de orden q, con q un entero positivo, también llamada distancia en norma L_q, se define por:

$$d_{L_q}(X_T, Y_T) = \left(\sum_{t=1}^{T} |X_t - Y_t|^q \right)^{1/q}. \tag{4.1}$$

Generaliza las bien conocidas distancias Euclídea ($q = 2$) y de Manhattan ($q = 1$). La métrica d_{L_q} tiene buenas propiedades analíticas y es ampliamente utilizada en cluster de datos estáticos. Sin embargo, por construcción, no considera la dependencia temporal entre observaciones, lo cual puede limitar su efectividad en escenarios donde la estructura secuencial de los datos es importante. También presenta una alta sensibilidad a transformaciones de la señal, como desplazamientos o cambios en la escala temporal (es decir, estiramiento o contracción del eje del tiempo [30]). Además, requiere igual longitud de las realizaciones comparadas, lo cual no siempre ocurrirá ya que frecuentemente el conjunto de series sujeto a cluster incluirá realizaciones de distinta longitud[1].

En esta memoria se pone el foco en el caso particular de la distancia Euclídea, $q = 2$ en (4.1), que se denotará específicamente por d_{EUCL} y que toma la forma:

$$d_{\text{EUCL}}(X_T, Y_T) = d_{L_2}(X_T, Y_T) = \sqrt{\sum_{t=1}^{T}(X_t - Y_t)^2}. \tag{4.2}$$

Los valores de esta métrica sobre cada par de realizaciones del Ejemplo 4.2.1 se dan en la Tabla 4.1, concluyendo que, si se emplea la distancia Euclídea, X_T e Y_T son más similares entre sí que con W_T.

$d_{\text{EUCL}}(X_T, Y_T)$	$d_{\text{EUCL}}(X_T, W_T)$	$d_{\text{EUCL}}(Y_T, W_T)$
0.6753908	1.8947537	1.3025147

Tabla 4.1: Distancias Euclídeas entre las realizaciones de la Figura 4.3.

- **Dynamic Time Warping (DTW)**

La distancia de alineamiento temporal dinámico (DTW) fue estudiada en profundidad por Sanko-Kruskal [31] y propuesta para encontrar patrones semejantes en series temporales no sincronizadas en el tiempo por Berndt-Clifford [32]. DTW tiene como objetivo "deformar" las series (estirando o comprimiendo el tiempo) de tal forma que se minimice alguna medida convencional de distancia entre las series deformadas (usualmente la distancia Euclídea). Si, por ejemplo, existen subsecuencias iguales en las dos series comparadas que aparecen desfasadas en el tiempo, entonces una "deformación" apropiada podría sincronizar estas subsecuencias. Básicamente, DTW busca ese alinamiento óptimo sobre el tiempo (sujeto a ciertas restricciones) y computa entonces una distancia convencional entre las series alineadas.

[1] Nótese que en esta memoria se asume que todas las series sujetas a cluster tienen siempre la misma longitud T sólo por simplificar la notación. Por este motivo es relevante enfatizar aquellas métricas que pueden evaluarse sobre realizaciones de diferente longitud, al tratarse de una buena propiedad.

Más formalmente, dadas dos realizaciones X_T e Y_T, denótese por M al conjunto de todas las posibles secuencias de m pares que preservan el orden de las observaciones en la forma

$$r = \{(X_{a_1}, Y_{b_1}), (X_{a_2}, Y_{b_2}), \ldots, (X_{a_m}, Y_{b_m})\},$$

con a_i y b_i tales que $1 \leq a_i, b_i \leq T$ y satisfaciendo que $a_1 = b_1 = 1$, $a_m = b_m = T$, y $a_{i+1} = a_i$ o $a_{i+1} = a_i + 1$ y $b_{i+1} = b_i$ o $b_{i+1} = b_i + 1$, para $i = 1, 2, \ldots, m-1$. Entonces, DTW toma la siguiente forma:

$$d_{\text{DTW}}(X_T, Y_T) = \min_{r \in M} \left(\sum_{i=1}^{m} (X_{a_i} - Y_{b_i})^2 \right)^{1/2}. \tag{4.3}$$

Mientras la distancia Euclídea compara los valores de las series de manera contemporánea, uno a uno y en la misma posición temporal, DTW permite alineaciones flexibles, comparando múltiples valores de una serie con un solo valor de la otra sobre la alineación temporal óptima, aquella que minimiza la distancia global. Esta estrategia es especialmente útil cuando las series presentan patrones similares pero desfasados temporalmente.

La Figura 4.4 ilustra las diferencias en los alineamientos empleados por DTW y la distancia Euclídea. Con la última, se comparan sólo valores en los instantes de tiempo contemporáneos, y así, por ejemplo, los emparejamientos en t_5 y en t_6 suponen mayor disimilitud que las comparaciones de la serie roja en t_5 y la azul en t_4, y de la roja en t_6 y la azul en t_5, ambas incluidas en el alineamiento temporal mucho más flexible de la distancia DTW.

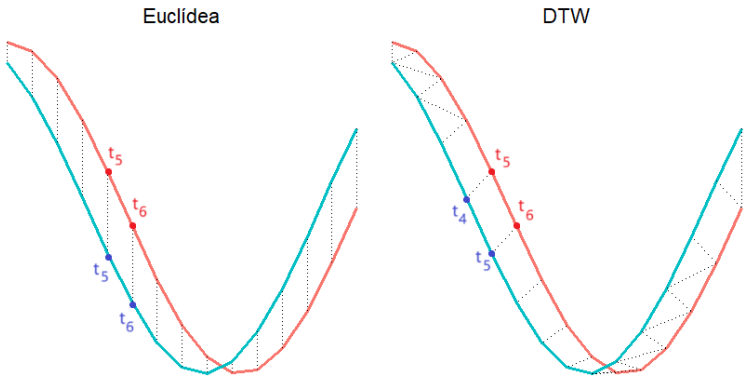


Figura 4.4: Alineamientos con la distancia Euclídea y con DTW.

A diferencia de la distancia Euclídea, DTW puede calcularse con series de distinta longitud, tal y como se muestra a modo de ejemplo en la Figura 4.5.

Existen limitaciones para DTW. Una debilidad importante de DTW es su alta complejidad computacional y, al igual que la distancia Euclídea, no tiene en cuenta la dependencia temporal entre observaciones.

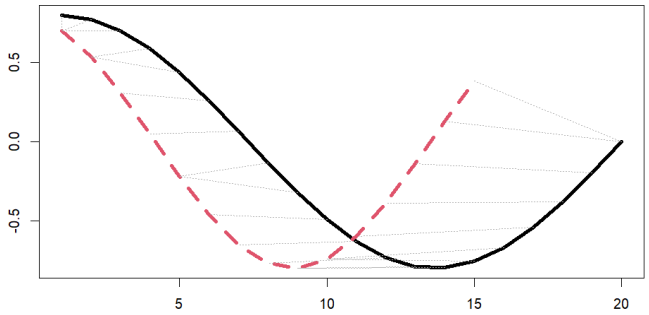


Figura 4.5: DTW sobre series de diferente longitud.

Las distancias DTW entre las realizaciones del Ejemplo 4.2.1 (ver Tabla 4.2) no modifican la relación de disimilitud de la distancia Euclídea.

$d_{\text{DTW}}(X_T, Y_T)$	$d_{\text{DTW}}(X_T, W_T)$	$d_{\text{DTW}}(Y_T, W_T)$
3.143	10.923	7.500

Tabla 4.2: Distancias DTW entre las realizaciones de la Figura 4.3.

En resumen, tanto la distancia Euclídea como Dynamic Time Warping (DTW) son métricas útiles para evaluar disimilitud entre series temporales cuando se desea discriminar entre formas ya que comparan valores en crudo de las series. DTW otorga además robustez a desfases temporales y permite realizaciones de distinta longitud . Sin embargo, ambas son inapropiadas si el objetivo es diferenciar sus estructuras de dependencia o sus conductas a largo plazo.

4.2.2 Cluster basado en características

Los métodos basados en características emplean las observaciones para generar valores que representan propiedades de las series y entonces miden disimilitud entre estos conjuntos de valores. En otros términos, una vez que el usuario establece aquellas propiedades de las series que deben regir el agrupamiento, los datos en crudo de cada serie temporal se reemplazan por un vector de características que reflejan adecuadamente las propiedades deseadas y se procede con el cluster. Por ejemplo, para detectar diferencias estructurales de manera global, Wang *et al.* [33] proponen sustituir las series observadas por vectores de un número elevado de estadísticos que miden: tendencia, estacionalidad, correlación serial, asimetría, no linealidad. . .

Los métodos basados en características presentan ventajas evidentes, como reducir la dimensión del problema, poder aplicarse a series con distinta longitud, y ser menos sensibles al

ruido aleatorio y a la ausencia de registros (el caso de observaciones faltantes es relativamente frecuente con series temporales).

En esta sección se introducen tres métricas basadas en características que requieren que los procesos generadores sean estrictamente estacionarios. La estacionariedad garantiza que las propiedades estadísticas de los procesos permanezcan invariables en el tiempo. En otras palabras, la distribución conjunta de cualquier conjunto de observaciones no cambia si se desplaza en el tiempo. Es decir, para cualquier conjunto de tiempos $t_1, t_2, \ldots, t_n$ y cualquier desplazamiento h, la distribución conjunta de $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ debe ser la misma que la de $(X_{t_1+h}, X_{t_2+h}, \ldots, X_{t_n+h})$. Si esta propiedad no se cumple, las características que se emplean no están bien definidas y por lo tanto los resultados obtenidos no son fiables.

- **Distancia basada en autocorrelaciones**

Varios autores han considerado análisis cluster basado en funciones de autocorrelaciones simples estimadas (ver p.e. Bohte *et al.* [34], Galeano-Peña [35], D'Urso-Maharaj [15]).

Sean $\hat{\rho}_{X_T} = (\hat{\rho}_{1,X_T}, \ldots, \hat{\rho}_{L,X_T})^\intercal$ y $\hat{\rho}_{Y_T} = (\hat{\rho}_{1,Y_T}, \ldots, \hat{\rho}_{L,Y_T})^\intercal$ los vectores de autocorrelación de hasta retardo L estimadas desde las realizaciones X_T e Y_T, respectivamente. El valor de L se elige de tal forma que la autocorrelación correspondiente se pueda estimar con un número razonable de observaciones y tal que $\hat{\rho}_{i,X_T} \approx 0$ y $\hat{\rho}_{i,Y_T} \approx 0$ para $i > L$. Galeano y Peña [35] definen la distancia entre X_T y Y_T en base a estos vectores de autocorrelación como sigue:

$$d_{\text{ACF}}(X_T, Y_T) = \sqrt{(\hat{\rho}_{X_T} - \hat{\rho}_{Y_T})^\intercal \, \Omega \, (\hat{\rho}_{X_T} - \hat{\rho}_{Y_T})}, \tag{4.4}$$

donde Ω es una matriz de pesos.

Con pesos uniformes, o sea, cuando $\Omega = I$, con I la matriz identidad, d_{ACF} en (4.4) evalúa la distancia Euclídea entre los vectores de autocorrelaciones simples estimadas:

$$d_{\text{ACFU}}(X_T, Y_T) = \sqrt{\sum_{i=1}^{L} (\hat{\rho}_{i,X_T} - \hat{\rho}_{i,Y_T})^2}. \tag{4.5}$$

Alternativamente se pueden considerar pesos geométricos que decaigan con el retraso de la autocorrelación y entonces d_{ACF} toma la forma:

$$d_{\text{ACFG}}(X_T, Y_T) = \sqrt{\sum_{i=1}^{L} p(1-p)^i (\hat{p}_{i,X_T} - \hat{p}_{i,Y_T})^2}, \text{ con } 0 < p < 1. \tag{4.6}$$

A modo de ejemplo, las autocorrelaciones estimadas hasta un retardo máximo $L = 10$ para las series del Ejemplo 4.2.1 se representan en la Figura 4.6. Las distancias Euclídeas entre cada par de estos vectores proporcionan las distancias d_{ACFU} entre los pares de realizaciones

originales (ver Tabla 4.3).

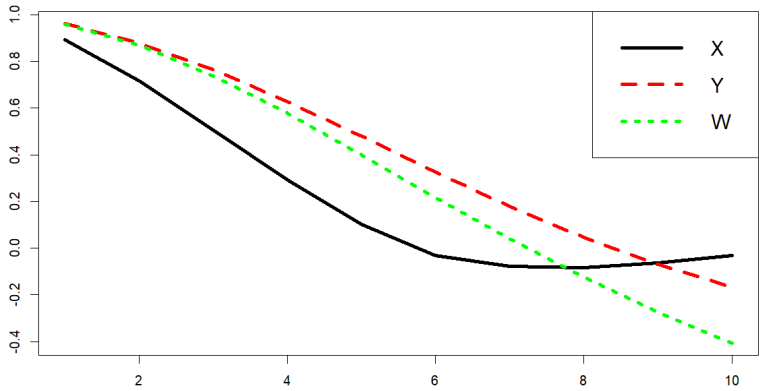


Figura 4.6: Primeras $L = 10$ autocorrelaciones estimadas para las series de la Figura 4.3.

$d_{\text{ACFU}}(X_T, Y_T)$	$d_{\text{ACFU}}(X_T, W_T)$	$d_{\text{ACFU}}(Y_T, W_T)$
0.763	0.717	0.410

Tabla 4.3: Distancias d_{ACFU} para las series de la Figura 4.3.

Las distancias d_{ACFU} en la Tabla 4.3 conducen a unas relaciones de proximidad diferentes a las indicadas con las distancias Euclídea y DTW en las Tablas 4.1 y 4.2, respectivamente. Con d_{ACFU}, Y_T y W_T son las series más cercanas, mientras que con los criterios basados en los datos en crudo las más próximas son X_T e Y_T. En otros términos, aunque el perfil de la realización Y_T está más próximo al de X_T (ver Figura 4.3), su estructura de autocorrelación lineal es más similar a la de W_T (ver Figura 4.6). Naturalmente, el usuario debe establecer el criterio apropiado atendiendo al concepto de similitud que desee evaluar.

- **Distancia basada en autocorrelaciones parciales**

En lugar de utilizar funciones estimadas de autocorrelación simple (ACF), también se pueden comparar las funciones estimadas de autocorrelación parcial (PACF), que miden correlación entre observaciones con retardo tras eliminar de ambas el efecto lineal de observaciones intermedias. La PACF es una herramienta útil y complementaria a la ACF para caracterizar modelos de series estacionarias, de modo que una métrica basada en PACF puede arrojar luz sobre diferencias entre este tipo de procesos.

Denotando por $\hat{\phi}_{X_T} = (\hat{\phi}_{1,X_T}, \ldots, \hat{\phi}_{L,X_T})^\intercal$ y $\hat{\phi}_{Y_T} = (\hat{\phi}_{1,Y_T}, \ldots, \hat{\phi}_{L,Y_T})^\intercal$ a los vectores de autocorrelaciones parciales de hasta retardo L estimadas a partir de las realizaciones X_T e Y_T, la nueva distancia $d_{\text{PACF}}(X_T, Y_T)$ se define de forma análoga a $d_{\text{ACF}}(X_T, Y_T)$ en (4.4) sin más que reemplazar $\hat{\rho}_{X_T}$ y $\hat{\rho}_{Y_T}$ por $\hat{\phi}_{X_T}$ y $\hat{\phi}_{Y_T}$, respectivamente.

Retomando el Ejemplo 4.2.1, los vectores de autocorrelaciones parciales estimadas hasta un retardo máximo $L = 10$ se representan en la Figura 4.7. Las distancias d_{PACFU} entre cada par de series considerando estos vectores estimados y pesos uniformes se proporcionan en la Tabla 4.4 y básicamente reportan valores semejantes a las d_{ACFU}.

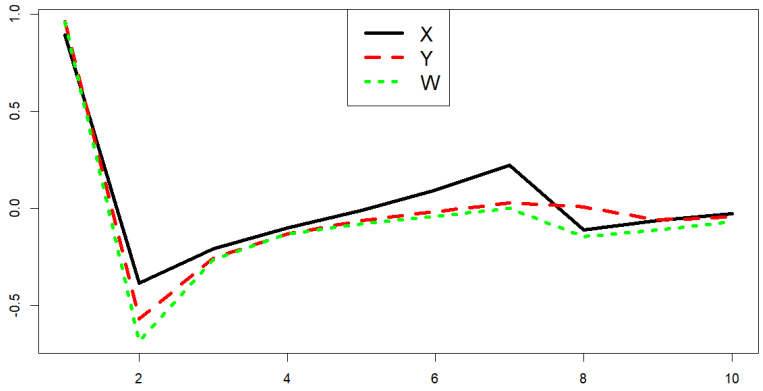


Figura 4.7: Primeras $L = 10$ autocorrelaciones parciales estimadas para las series de la Figura 4.3.

$d_{\text{PACFU}}(X_T, Y_T)$	$d_{\text{PACFU}}(X_T, W_T)$	$d_{\text{PACFU}}(Y_T, W_T)$
0.331	0.417	0.203

Tabla 4.4: Distancias d_{PACFU} para las series de la Figura 4.3.

- **Distancia basada en la función de autocovarianzas cuantil**

En Vilar *et al.* [36] se propone una distancia basada en comparar estimaciones de autocovarianzas cuantil y se ilustra su potencial en análisis cluster de series estacionarias.

Sea F_X la distribución marginal de un proceso estacionario X_t y $q_{\tau,X} = F_X^{-1}(\tau)$, $\tau \in [0, 1]$, la función cuantil correspondiente. La función de autocovarianzas cuantil (QAF) para un retardo l se define como:

$$\gamma_l(\tau, \tau') = \text{Cov}\left(I(X_t \leq q_{\tau,X}), I(X_{t+l} \leq q_{\tau',X})\right), \tag{4.7}$$

con $\tau, \tau' \in [0, 1]$ e $I(\cdot)$ denotando la función indicadora que toma el valor 1 si su argumento es cierto y 0 en otro caso.

Nótese que $\gamma_l(\tau, \tau') = \mathbb{P}(X_t \leq q_{\tau,X}, X_{t+l} \leq q_{\tau',X}) - \tau\tau'$, donde el primer término es la probabilidad conjunta de que X_t esté por debajo del τ-cuantil y X_{t+l} por debajo del τ'-cuantil. Por tanto, dada una realización X_T del proceso X y fijados dos niveles de probabilidad τ y

τ', se obtiene una estimación $\hat{\gamma}_l(\tau, \tau')$ calculando los respectivos cuantiles muestrales $\hat{q}_{\tau,X}$ y $\hat{q}_{\tau',X}$ y después la frecuencia observada relativa del suceso $(X_t \leq \hat{q}_{\tau,X}, X_{t+l} \leq \hat{q}_{\tau',X})$. En otros términos:

$$\hat{\gamma}_l(\tau, \tau') = \frac{1}{T-l} \sum_{t=1}^{T-l} I\left((X_t \leq \hat{q}_{\tau,X}) \cap (X_{t+l} \leq \hat{q}_{\tau',X})\right) - \tau\tau' \tag{4.8}$$

La covarianza cuantil informa en qué medida la probabilidad acumulada hasta un punto en X_t ayuda a predecir la probabilidad acumulada en ese punto l instantes más tarde. Es por tanto mucho más informativa que la autocovarianza convencional que se centra exclusivamente en las medias en ambos instantes de tiempo. Además, está siempre bien definida porque trabaja con funciones indicadoras y tiene una alta capacidad para capturar diversos tipos de dependencia serial, incluyendo casos donde la autocorrelación es cero o existe dependencia en las colas de la distribución. Por construcción, QAF es robusta a valores atípicos y distribuciones con colas pesadas. Estas propiedades, junto con su bajo costo computacional, convierten a una métrica basada en comparar QAF en una potente herramienta para realizar clustering de series temporales.

Para construir una distancia entre dos realizaciones de series X_T e Y_T considerando los valores estimados de sus QAF, se procede fijando un retardo máximo L y una secuencia de niveles de probabilidad $\tau_1, \ldots, \tau_r \in [0, 1]$ y usando (4.8) para calcular las matrices $\hat{\Gamma}_X = (\hat{\gamma}_l(\tau_{i,X}, \tau_{j,X}))_{i,j=1}^r$ y $\hat{\Gamma}_Y = (\hat{\gamma}_l(\tau_{i,Y}, \tau_{j,Y}))_{i,j=1}^r$. Entonces estas matrices se reordenan en forma de vectores $vec(\hat{\Gamma}_X)$ y $vec(\hat{\Gamma}_Y)$ y se computa una distancia convencional entre ambos. En esta memoria se usará siempre la distancia Euclídea, resultando:

$$d_{\text{QAF}}(X_T, Y_T) = \sqrt{\sum_{l=1}^{L} \sum_{i=1}^{r} \sum_{j=1}^{r} \left(\hat{\gamma}_l(\tau_{i,X}, \tau_{j,X}) - \hat{\gamma}_l(\tau_{i,Y}, \tau_{j,Y})\right)^2} \tag{4.9}$$

En la práctica es suficiente considerar unos pocos niveles cuantil para obtener una buena precisión de la métrica d_{QAF}. Por ejemplo, se han calculado las distancias basadas en QAF para las realizaciones de la Figura 4.3 usando $L = 1$ y sólo tres niveles cuantil, $\tau \in \{0.1, 0.5, 0.9\}$. Los vectores de autocovarianza cuantil resultantes, $vec(\hat{\Gamma}_X)$, $vec(\hat{\Gamma}_Y)$ y $vec(\hat{\Gamma}_W)$, se representan en la Figura 4.8, mientras que los valores de distancia a que dan lugar se reproducen en la Tabla 4.5.

Los resultados mantienen la misma ordenación de distancias que con ACF y PACF, de modo que los tres criterios separan de igual forma las estructuras de dependencia subyacentes. Sin embargo, con QAF esta separación es menos nítida.

Si se aumenta el número de niveles cuantil, p.e. $\tau \in \{0.1, 0.25, 0.4, 0.5, 0.6, 0.75, 0.9\}$, se obtienen las distancias de la Tabla 4.6, observándose cambios en magnitud pero con dife-

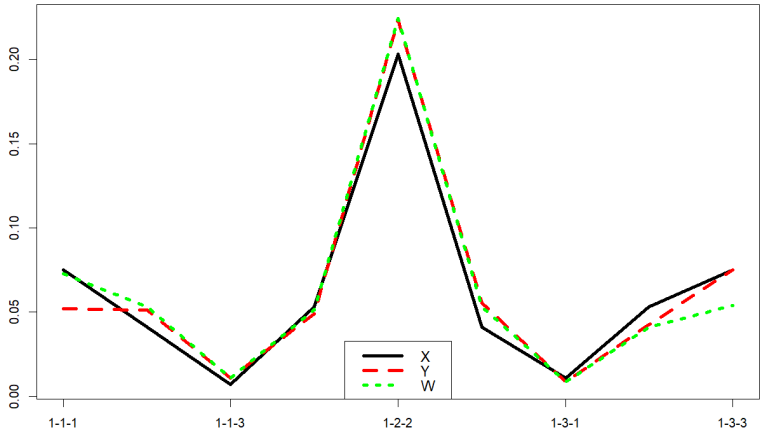


Figura 4.8: Vectores de autocovarianzas cuantil $vec(\hat{\Gamma}_X)$, $vec(\hat{\Gamma}_Y)$ y $vec(\hat{\Gamma}_W)$ usando $\tau \in \{0.1, 0.5, 0.9\}$ y $L = 1$ para las series de la Figura 4.3.

$d_{\text{QAF}}(X_T, Y_T)$	$d_{\text{QAF}}(X_T, W_T)$	$d_{\text{QAF}}(Y_T, W_T)$
0.0375	0.0367	0.0297

Tabla 4.5: Distancias d_{QAF} basadas en niveles cuantil $\tau \in \{0.1, 0.5, 0.9\}$ y $L = 1$ para las series de la Figura 4.3.

rencias semejantes, corroborando así que, en general, usar un conjunto pequeño de niveles cuantil es suficiente.

$d_{\text{QAF}}(X_T, Y_T)$	$d_{\text{QAF}}(X_T, W_T)$	$d_{\text{QAF}}(Y_T, W_T)$
0.107	0.095	0.041

Tabla 4.6: Distancias d_{QAF} basadas en niveles cuantil $\tau \in \{0.1, 0.25, 0.4, 0.5, 0.6, 0.75, 0.9\}$ y $L = 1$ para las series de la Figura 4.3.

En resumen, se han descrito algunas métricas basadas en características, concretamente en autocorrelaciones simples, autocorrelaciones parciales y autocovarianzas cuantil. Todas ellas capturan propiedades estructurales de las series y asumen estacionariedad de los procesos generadores. Son por tanto apropiadas para realizar análisis cluster en este contexto y pueden detectar patrones que no resultan evidentes al visualizar los perfiles gráficos de las series. Una de sus ventajas es que se pueden calcular con series de diferente longitud, aunque una evaluación precisa de similitud con estas métricas requiere disponer de realizaciones no demasiado cortas para capturar adecuadamente las dinámicas subyacentes.

4.2.3 Cluster basado en modelos

Otra vía para definir distancias entre realizaciones de series de tiempo es asumir que sus procesos generadores proceden de familias de modelos paramétricos específicos. En tal caso se pueden estimar los parámetros y medir la discrepancia que existe entre estas estimaciones. Un buen número de trabajos se han centrado en el caso de series generadas por modelos ARIMA invertibles. Uno de las primeros criterios de distancia siguiendo esta idea fue la métrica propuesta por Piccolo en 1990 [37] que se presenta a continuación.

- **Distancia de Piccolo**

La expansión autorregresiva AR(∞) de un proceso ARIMA invertible contiene toda la información relevante sobre la estructura estocástica de estos procesos (excluyendo los valores iniciales). En base en este resultado, Piccolo [37] propone evaluar disimilitud entre dos modelos ARIMA invertibles a través de la distancia Euclídea entre los operadores AR(∞) que aproximan las estructuras ARIMA correspondientes. En el caso de series no estacionarias, se diferencian para lograr la estacionariedad y, en presencia de estacionalidad, ésta debe cancelarse antes de proceder con el análisis.

Con series estacionarias y libres de componente estacional, se emplea un criterio definido de selección del modelo, como el Criterio de Información de Akaike (AIC) o el Criterio de Información Bayesiano (BIC), para ajustar modelos AR(∞) truncados que aproximen los procesos generadores de las dos series comparadas. Si los órdenes de truncamiento apropiados, digamos k_1 y k_2, son distintos, por ejemplo $k_1 < k_2$, entonces los k_1 coeficientes autorregresivos de la estructura AR(k_1) se completarían con $k_2 - k_1$ coeficientes nulos. De esta forma se tendría dos vectores de coeficientes estimados de igual longitud (k_2), facilitando así el uso de una distancia convencional entre los mismos, tal y como se formaliza a continuación..

Si $\psi_X = (\psi_{1,X_T}, \ldots, \psi_{k_1,X_T})^\intercal$ y $\psi_Y = (\psi_{1,Y_T}, \ldots, \psi_{k_2,Y_T})^\intercal$ denotan los vectores de estimaciones de parámetros AR(k_1) y AR(k_2) para realizaciones X_T e Y_T, respectivamente, entonces la distancia de Piccolo se define como:

$$d_{\text{PIC}}(X_T, Y_T) = \sqrt{\sum_{j=1}^{k} (\psi'_{j,X_T} - \psi'_{j,Y_T})^2}, \tag{4.10}$$

donde $k = \max(k_1, k_2)$, $\psi'_{j,X_T} = \psi_{j,X_T}$, si $j \leq k_1$, y $\psi'_{j,X_T} = 0$, en otro caso; y análogamente, $\psi'_{j,Y_T} = \psi_{j,Y_T}$, si $j \leq k_2$, y $\psi'_{j,Y_T} = 0$, en otro caso.

El procedimiento descrito para definir d_{PIC} en (4.10) evita la ardua labor de estimar modelos ARMA para cada serie, seguramente de diferentes órdenes. Por otro lado, d_{PIC} es una métrica basada en modelos que satisface las propiedades de una distancia, a saber no negatividad, simetría y desigualdad triangular, garantizando su validez para realizar análisis cluster

de series de tiempo.

En orden a evaluar la distancia de Piccolo sobre los pares de realizaciones del Ejemplo 4.2.1, se estiman primero los parámetros AR para cada proceso, resultando $\psi_X = (1.155 - 0.127)$, $\psi_Y = (1.362 - 0.181 - 0.258)$ y $\psi_W = (1.436 - 0.258 - 0.262)$. Es decir, los órdenes de truncamiento han sido 2, 3 y 3 para X_T, Y_T y W_T, respectivamente. Las distancias de Piccolo entre las series coinciden con las distancias Euclídeas entre estos vectores de parámetros estimados, completado con un cero adicional el de X_T. Los resultados se muestran en la Tabla 4.7.

$d_{\text{PIC}}(X_T, Y_T)$	$d_{\text{PIC}}(X_T, W_T)$	$d_{\text{PIC}}(Y_T, W_T)$
0.335	0.406	0.107

Tabla 4.7: Distancias d_{PIC} para las series de la Figura 4.3.

En resumen, la distancia de Piccolo es una métrica específicamente diseñada para medir disimilitud entre series temporales generadas desde procesos ARIMA. Por tanto es un criterio muy apropiado para discriminar entre modelos generadores siempre y cuando los modelos estén bien especificados.

Entre sus puntos débiles, su precisión depende de la calidad de las estimaciones de los parámetros AR(∞), que puede verse afectada con series cortas o ruidosas y, por supuesto, como cualquier criterio basado en modelos, es absolutamente dependiente de la correcta especificación de los modelos.

Recapitulando, en este capítulo se ha presentado un abanico de criterios de distancia entre series de tiempo. Se han incluido métricas que evalúan disimilitud en forma basadas en las observaciones en crudo de las series temporales , tales como la distancia Euclídea y Dynamic Time Warping (DTW). Para medir discrepancias estructurales se han presentado distancias basadas en características (libres de modelo) y distancias basadas en modelos. Entre las primeras se encuentran las distancias basadas en funciones de autocorrelaciones simples (ACF), funciones de autocorrelaciones parciales (PACF) y vectores de autocovarianzas cuantil (QAF), mientras que se ha elegido la distancia de Piccolo, que compara modelos ARIMA, como representante de distancias basadas en modelos. En todos los casos se han enfatizado pros y contras de las métricas presentadas.

Capítulo 5

Estudio de simulación

En este capítulo se presentan los resultados de un estudio de simulación diseñado para evaluar el desempeño de las distancias introducidas en la Sección 4.2 en análisis cluster de series. Se pone el foco en series estacionarias, de modo que el cluster está orientado a discriminar entre estructuras de dependencia. En concreto, se realiza clustering difuso particional mediante el algoritmo Fuzzy C-Medoids con cada una de las métricas y se analiza la calidad de las diferentes soluciones cluster sobre una variedad de escenarios. El objetivo es doble. Por un lado, ilustrar la importancia de elegir una distancia adecuada según el tipo de series sujetas a cluster. Por otro, enfatizar las ventajas del camino fuzzy. Con estas metas en mente, se diseñan escenarios conformados por grupos de series temporales con distintos procesos generadores, incluyendo modelos lineales y no lineales. El capítulo se estructura como sigue. Tras detallar el procedimiento experimental, se describen y justifican los escenarios y se analizan la eficiencia cluster y el coste computacional con las diferentes métricas.

5.1 Procedimiento experimental

5.1.1 Diseño de las simulaciones

Cada escenario considerado consta de dos o de tres grupos de realizaciones de series temporales estacionarias, con 10 series de igual longitud T en cada grupo. La homogeneidad del grupo radica en que todas las series que lo forman se han generado del mismo proceso estocástico o con pequeñas diferencias en los parámetros del modelo que lo definen. En cambio, series de diferentes grupos provienen de procesos estocásticos distintos o de un mismo proceso, pero definido por parámetros bien diferentes. En todos los casos, las innovaciones (ε_t) siguen una distribución normal estándar.

Se controla el efecto de algunos parámetros del procedimiento. En particular, se examina la influencia de la longitud de las series repitiendo los experimentos para distintos valores de T, incluyendo en esta memoria los resultados con $T = 50$ y $T = 200$. La única razón para

considerar igual longitud de las series es poder ejecutar todas las métricas, pero conviene recordar que algunas trabajan también con series de diferente longitud (ver Sección 4.2).

El coeficiente de nivel de solapamiento m también tiene un papel crucial y en la práctica debe determinarse de antemano. Hasta donde sabemos, no existen argumentos teóricos que respalden una elección óptima de m (ver [38]). En base a otros estudios experimentales [39, 40] y a argumentos esgrimidos en [38, 41], se optó por emplear tres valores: $m \in \{1.2, 1.5, 1.8\}$.

Con respecto a las métricas, la métrica basada en autocovarianzas cuantil (QAF) se ejecuta considerando los niveles cuantil $\{0.1, 0.5, 0.9\}$, en tanto que se emplean las diez primeras autocorrelaciones para calcular las distancias basadas en la ACF y la PACF.

Fijado un escenario, se simula el conjunto de series temporales sujetas a cluster y se ejecuta el algoritmo Fuzzy C-Medoids (FCMdC) con cada una de las distancias evaluadas. Para ello se proporcionan también como argumentos de entrada: el número correcto de grupos (C), el valor de m, la matriz simétrica con las distancias entre cada par de series, el número máximo de iteracciones (20) y los medoides iniciales para arrancar el algoritmo, que se seleccionan utilizando el método PAM [18]. La salida del algoritmo incluye la matriz de grados de pertenencia de las series a los clusters y los medoides resultantes. Para cada escenario este procedimiento se replica $N = 100$ veces.

5.1.2 Criterios de evaluación

Como en escenarios simulados se conoce la agrupación real, una vía de medir la calidad del procedimiento cluster es calcular el promedio de los porcentajes de series correctamente clasificadas en cada una de las 100 réplicas del experimento. Ahora bien, para ello es necesario establecer sin ambigüedad cuándo una serie está "correctamente clasificada" porque en cluster difuso una serie puede asignarse simultáneamente a varios clusters. Por tal motivo, se adopta el criterio de considerar que una serie está bien clasificada si y solo si su grado de pertenencia al grupo verdadero es superior a un umbral prefijado, por ejemplo, 0.7 si la "ground truth" está formada por dos clusters, y algo menor si la conforman tres clusters.

A modo de ejemplo, supóngase un escenario con dos grupos de tres series ($\{S_1, S_2, S_3\}$ y $\{S_4, S_5, S_6\}$) y que el análisis cluster conduce a los grados de membresía de la Tabla 5.1, donde $u_{S_j,i}$ denota el grado de membresía de la serie S_j, $j = 1, \ldots, 6$, para el i-ésimo cluster, $i = 1, 2$. Un total de 4 series, S_1, S_3, S_4 y S_6, reciben un grado de pertenencia a su grupo de origen superior a 0.7, de modo que, con este umbral, el porcentaje de éxito es 66.67%.

Como métricas adicionales de evaluación se calculan también sendas extensiones de los índices de Rand ajustado (ARI) y de Jaccard, convenientemente adaptadas para el particionamiento difuso y referidas en esta memoria como ARI.F y Jaccard.F, respectivamente. Los índices ARI y Jaccard evalúan el grado de concordancia entre dos particiones, en tanto que sus versiones difusas tienen en cuenta también los grados de membresía asociados a los ob-

	Series grupo 1			Series grupo 2		
	S_1	S_2	S_3	S_4	S_5	S_6
$u_{S_j,1}$	**0.9**	0.6	**0.75**	0.2	0.4	0.15
$u_{S_j,2}$	0.1	0.4	0.25	**0.8**	0.6	**0.85**

Tabla 5.1: Evaluación cluster basada en fijar en 0.7 el umbral para el grado de membresía.

jetos de cada grupo. Para más información sobre estas métricas se puede consultar [42]. Para su correcta interpretación se tendrá en cuenta lo que sigue. El ARI.F se mueve entre -1 y 1, de modo que valores cercanos a 1 indican una alta concordancia entre las particiones, valores entorno a 0 sugieren una concordancia similar a la esperada por azar, y valores cerca de -1 indican una coincidencia menor que la esperada por azar. Por otro lado, el índice de Jaccard.F devuelve un valor entre 0 y 1, aumentando (disminuyendo) el nivel de concordancia entre particiones cuanto más cerca a 1 (0) sea su valor. Ambas métricas están implementadas en la librería **fclust**. En el ejemplo de la Tabla 5.1, ARI.F=0.335 y Jaccard.F=0.446, indicando ambos un razonable nivel de concordancia entre la agrupación real y la partición experimental.

5.2 Experimentos con procesos lineales

5.2.1 Escenarios

Los primeros experimentos engloban escenarios con realizaciones de procesos estacionarios lineales. En concreto, todas las series se generan de procesos autorregresivos de orden uno, AR(1). Como las innovaciones son ruido blanco en todos los casos, cada proceso queda caracterizado por el coeficiente autorregresivo (ϕ). Es decir, dos series son tanto más distintas cuanto más disten los coeficientes autorregresivos de sus procesos generadores. En este marco se consideran los escenarios cluster 1.A, 1.B y 1.C detallados en la Tabla 5.2

Las series de cada cluster proceden de modelos AR(1) con coeficientes ϕ extraídos al azar de un rango delimitado y diferente para clusters distintos. Al no fijar el valor de ϕ para cada cluster se pretende añadir más incertidumbre en la configuración de los clusters y dificultar la tarea de clasificación. Con respecto al Escenario 1.A, el 1.B incrementa el número de clusters. El Escenario 1.C replica el 1.A, pero añadiendo una serie equidistante de ambos clusters con la idea de comprobar si el algoritmo fuzzy es capaz de asignar esta serie a los dos clusters simultáneamente. Las realizaciones de una réplica elegida al azar del Escenario 1.A se muestran en la Figura 5.1.

Los perfiles de las series en la Figura 5.1 permiten intuir que métricas basadas en forma como la distancia Euclídea o DTW podrían no dar lugar a resultados óptimos. Por el contra-

Proceso Generador	Escenario	Estructura
$X_t = \phi X_{t-1} + \varepsilon_t$	1.A	Cluster C1: 10 series con $\phi \sim U(0.1, 0.4)$
		Cluster C2: 10 series con $\phi \sim U(0.6, 0.9)$
	1.B	Cluster C1: 10 series con $\phi \sim U(-0.8, -0.4)$
		Cluster C2: 10 series con $\phi \sim U(-0.2, 0.2)$
		Cluster C3: 10 series con $\phi \sim U(0.4, 0.8)$
	1.C	Cluster C1: 10 series con $\phi \sim U(0.1, 0.4)$
		Cluster C2: 10 series con $\phi \sim U(0.6, 0.9)$
		Una serie equidistante con $\phi = 0.5$

Tabla 5.2: Escenarios de simulación clustering de modelos lineales.

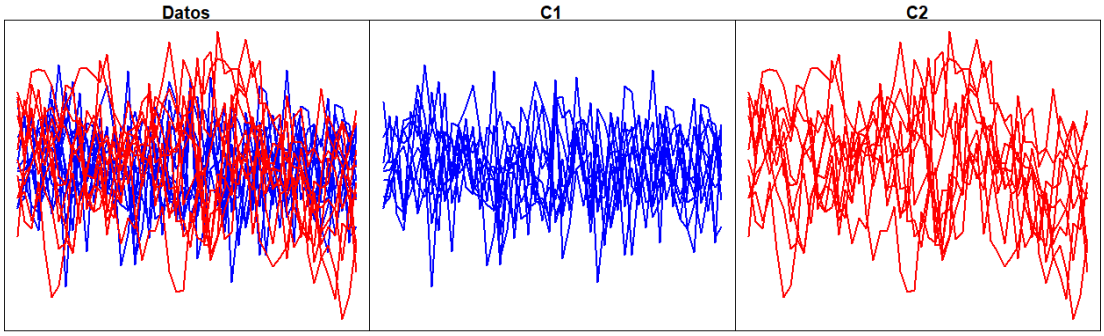


Figura 5.1: Realizaciones de una réplica del escenario de simulación 1.A.

rio, la distancia de Piccolo se fundamenta en discriminar entre estructuras ARIMA, de modo que es razonable esperar que proporcione buenos resultados en estos escenarios envolviendo modelos AR(1).

5.2.2 Resultados

Escenario 1.A.

Con un umbral de valor 0.7, los porcentajes promedio de series bien clasificadas en el Escenario 1.A se presentan en la Tabla 5.3, ordenados de mayor a menor y para longitudes $T = 50$ (a) y $T = 200$ (b). Las desviaciones estándar de estos promedios sobre las 100 réplicas se indican en paréntesis.

Los resultados son acordes a lo esperado. El clustering basado en métricas fundamentadas en discriminar entre estructuras (Piccolo, QAF, ACF y PACF) conduce a tasas de éxito muy superiores a aquellas obtenidas empleando métricas basadas en discriminar en forma (Euclídea

Métrica	m	% Éxito (sd)
PICCOLO	1.2	0.850 (0.147)
QAF	1.2	0.795 (0.192)
PICCOLO	1.5	0.779 (0.144)
ACF	1.2	0.718 (0.186)
QAF	1.5	0.697 (0.208)
PACF	1.2	0.680 (0.194)
PICCOLO	1.8	0.672 (0.145)
ACF	1.5	0.579 (0.201)
QAF	1.8	0.516 (0.224)
ACF	1.8	0.349 (0.216)
PACF	1.5	0.310 (0.198)
DTW	1.2	0.235 (0.145)
EUCL	1.2	0.182 (0.160)
PACF	1.8	0.133 (0.076)
DTW	1.5	0.073 (0.028)
DTW	1.8	0.069 (0.024)
EUCL	1.8	0.059 (0.019)
EUCL	1.5	0.058 (0.020)

(a) T = 50

Métrica	m	% Éxito (sd)
PICCOLO	1.2	0.980 (0.036)
QAF	1.2	0.969 (0.041)
PACF	1.2	0.963 (0.044)
PICCOLO	1.5	0.947 (0.049)
QAF	1.5	0.940 (0.059)
QAF	1.8	0.891 (0.075)
PICCOLO	1.8	0.870 (0.077)
ACF	1.2	0.864 (0.158)
PACF	1.5	0.848 (0.114)
ACF	1.5	0.839 (0.124)
ACF	1.8	0.772 (0.131)
PACF	1.8	0.544 (0.208)
DTW	1.2	0.148 (0.137)
DTW	1.5	0.070 (0.025)
DTW	1.8	0.070 (0.025)
EUCL	1.2	0.056 (0.051)
EUCL	1.5	0.050 (0.005)
EUCL	1.8	0.050 (0.005)

(b) T = 200

Tabla 5.3: Porcentajes promedio de éxito del algoritmo FCMdC en el Escenario 1.A con diferentes métricas y niveles de solapamiento (m) para un umbral 0.7.

y DTW). Como el Escenario 1.A engloba series autorregresivas, la métrica de Piccolo genera los mejores resultados, con una media de fallos de sólo un 2% con series de longitud 200. Sin embargo, es muy relevante remarcar que la métrica basada en autocovarianzas cuantil (QAF), libre de modelo por construcción, es claramente competitiva con Piccolo. En una posición intermedia figuran ACF y PACF, que son penalizadas por el ruido introducido al usar coeficientes de autocorrelación no significativos, lo que también explica que, en general, presenten las desviaciones estándar más elevadas. Si se omite el caso $m = 1.8$, las tasas de éxito con todas las métricas basadas en discriminar entre estructuras no bajan de 0.839 cuando $T = 200$, lo que pone de manifiesto que requieren de realizaciones no demasiado largas para capturar adecuadamente las dinámicas subyacentes. De hecho, incluso con series cortas ($T = 50$) estas métricas, particularmente Piccolo y QAF, también alcanzan puntuaciones altas.

Al no existir solapamiento pronunciado en la definición de los clusters, valores pequeños de m deberían mejorar los resultados ya que producen clusters más separados, es decir, niveles de membresía menos equirepartidos entre los dos clusters, favoreciendo así superar el umbral 0.7. Esta característica del escenario simulado justifica que la eficiencia cluster disminuya al crecer m, independientemente de la métrica considerada.

La distancia Euclídea y DTW conducen a una conducta cluster muy pobre, con tasas de éxito entre el 5 y el 15% cuando $T = 200$. Básicamente, no son métricas apropiadas para el objetivo cluster que se persigue en este escenario. Tal es así que ni siquiera mejoran sus resultados al aumentar T, al contrario que el resto de métricas que estiman con mayor precisión

las características que las definen con series más largas y así mejoran su conducta cluster.

La totalidad de valoraciones previas se sostienen considerando como criterios de calidad cluster los índices ARI.F y Jaccard.F. La Figura A.1 en el Apéndice A de esta memoria muestra los diagramas de caja para estos índices basados en las 100 réplicas de la simulación y para todas las combinaciones de métricas y valores de m. La ubicación y ancho de las cajas caracteriza la distribución de los índices y corrobora en todos sus puntos el análisis de los resultados según el criterio de fijar un umbral 0.7 para clasificar correctamente una serie.

Escenario 1.B.

La peculiaridad del Escenario 1.B radica en introducir un tercer cluster equidistante de los otros dos y analizar de qué forma afecta a la efectividad cluster. Los perfiles de las realizaciones de una réplica arbitraria en este escenario se muestran en la Figura 5.2.

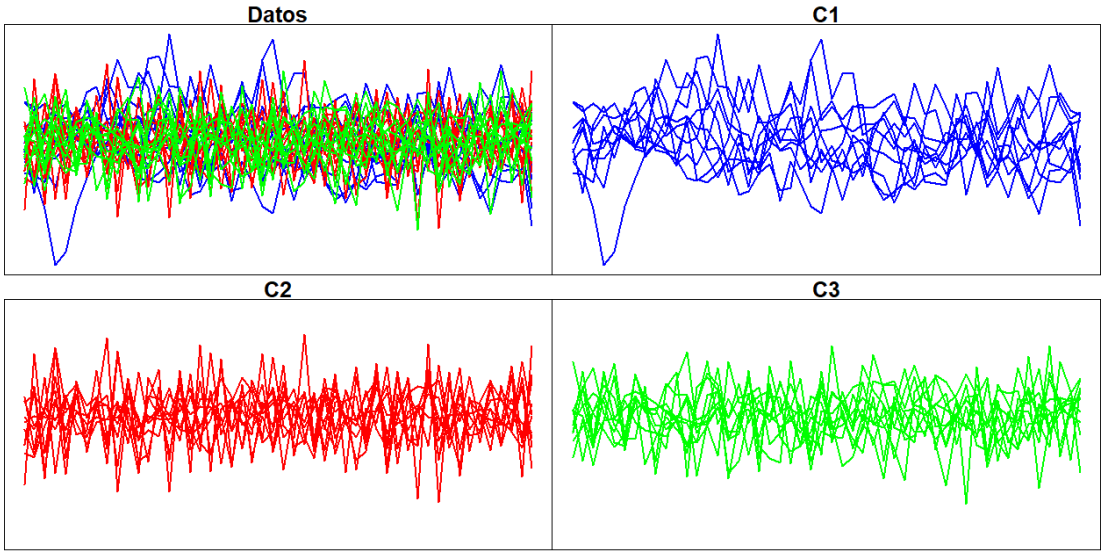


Figura 5.2: Realizaciones de una réplica del escenario de simulación 1.B.

Con tres clusters, se opta por fijar un umbral 0.6. Superar este umbral para un cluster supone una confianza razonable en la pertenencia al mismo ya que los niveles de membresía para los otros dos clusters suman menos de 0.4. Nótese que, en un contexto difuso, una serie podría estar a caballo de dos o incluso de tres clusters, con niveles de membresía bastante repartidos y, por esta razón, no sería recomendable fijar un valor excesivamente alto para este umbral. Los resultados del análisis cluster en el Escenario 1.B se muestran en la Tabla 5.4.

No se observan cambios importantes en las tasas de éxito con respecto al Escenario 1.A. La métrica de Piccolo toma de nuevo ligera ventaja respecto a las demás métricas estructurales, con promedios más altos y mínimas desviaciones estándar. Con longitud 50 se obtienen ya estimaciones precisas de los coeficientes autorregresivos, lo que redunda en su buen comportamiento aún incrementando la complejidad del escenario. En general, las mismas conclusio-

Métrica	m	% Éxito (sd)
PICCOLO	1.2	0.862 (0.142)
PICCOLO	1.5	0.806 (0.143)
QAF	1.2	0.768 (0.218)
PACF	1.2	0.750 (0.193)
QAF	1.5	0.711 (0.204)
PICCOLO	1.8	0.705 (0.126)
ACF	1.2	0.696 (0.207)
QAF	1.8	0.554 (0.173)
ACF	1.5	0.491 (0.173)
PACF	1.5	0.444 (0.172)
ACF	1.8	0.233 (0.132)
PACF	1.8	0.175 (0.093)
DTW	1.2	0.093 (0.073)
EUCL	1.2	0.058 (0.036)
DTW	1.5	0.053 (0.020)
DTW	1.8	0.051 (0.021)
EUCL	1.5	0.044 (0.019)
EUCL	1.8	0.044 (0.019)

(a) T = 50

Métrica	m	% Éxito (sd)
PICCOLO	1.2	0.981 (0.025)
PACF	1.2	0.967 (0.036)
QAF	1.2	0.961 (0.068)
PICCOLO	1.5	0.952 (0.040)
QAF	1.5	0.945 (0.042)
PACF	1.5	0.884 (0.072)
ACF	1.2	0.876 (0.132)
QAF	1.8	0.875 (0.056)
PICCOLO	1.8	0.870 (0.060)
ACF	1.5	0.814 (0.131)
ACF	1.8	0.639 (0.114)
PACF	1.8	0.639 (0.098)
DTW	1.2	0.057 (0.028)
DTW	1.5	0.051 (0.017)
DTW	1.8	0.048 (0.017)
EUCL	1.5	0.037 (0.010)
EUCL	1.8	0.037 (0.010)
EUCL	1.2	0.036 (0.009)

(b) T = 200

Tabla 5.4: Porcentajes promedio de éxito del algoritmo FCMdC en el Escenario 1.B con diferentes métricas y niveles de solapamiento (m) para un umbral 0.6.

nes extraídas de los resultados del Escenario 1.A son válidas en este nuevo escenario con un cluster adicional. También el análisis basado en los índices ARI.F y Jaccard.F (ver Figura A.2 en el Apéndice A) reporta las mismas conclusiones.

Escenario 1.C.

Como se ha indicado, la motivación del Escenario 1.C es evaluar si el algoritmo Fuzzy C-Medoids (FCMdC) con la métrica correspondiente es capaz de detectar la equidistancia de una serie a los dos clusters subyacentes, mostrando así flexibilidad para tratar con escenarios difusos donde algunas series podrían asignarse a más de un cluster. Las realizaciones de una réplica arbitraria en el Escenario 1.C se muestran en la Figura 5.3.

La serie equidistante impide calcular de forma rigurosa los índices ARI.F y Jaccard.F. Por ello, para evaluar el desempeño clustering se pone el foco en los grados de membresía resultantes como sigue. Una serie de un cluster concreto se asigna correctamente a ese cluster si presenta un grado de membresía al mismo superior a 0.7. Por otro lado, la serie equidistante se considera bien clasificada si presenta valores de membresía razonablemente equirepartidos, ambos moviéndose entre 0.3 y 0.7, indicando así su pertenencia difusa a ambos grupos.

A la luz del mal comportamiento de las métricas basadas en forma con series siguiendo un modelo AR(1), se decide no incluirlas en el estudio del Escenario 1.C.

La Figura 5.4 recoge una serie de gráficos de caja basados en los grados de membresía resultantes sobre las 100 réplicas del Escenario 1.C, con $T = 200$, que permiten interpretar la calidad del clustering con cada métrica. Específicamente, las cajas de la fila superior concier-

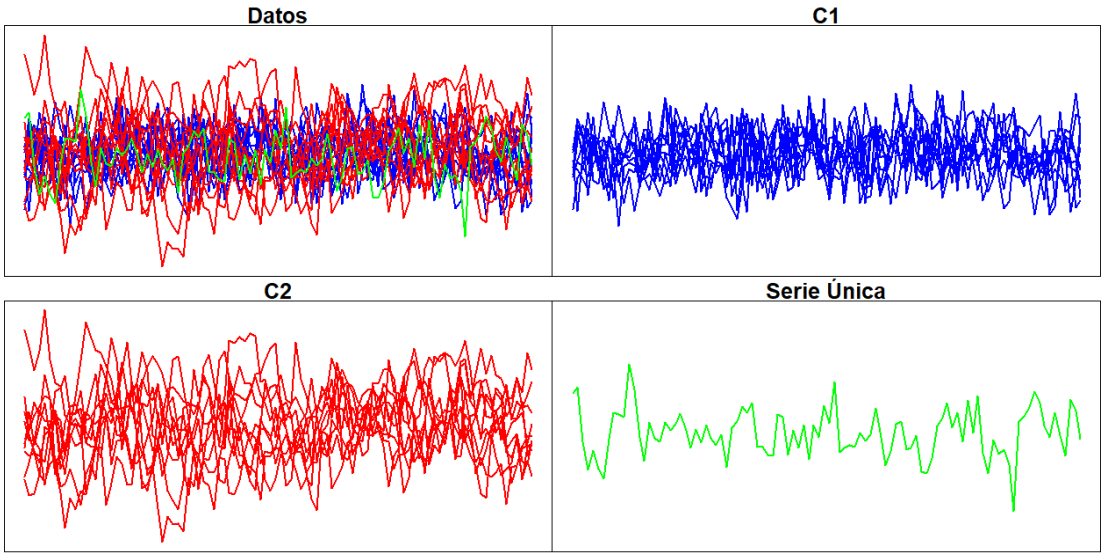


Figura 5.3: Realizaciones de una réplica del escenario de simulación 1.C.

nen a las series de ambos clusters y se forman con los niveles de membresía retornados para el cluster real de pertenencia de cada serie. O sea, si una serie es original del Cluster C1, se considera sólo su valor de membresía a C1, y análogamente si pertenece al Cluster C2. En la fila inferior, las cajas se forman con los niveles de membresía alcanzados para el Cluster C1 de las series equidistantes.

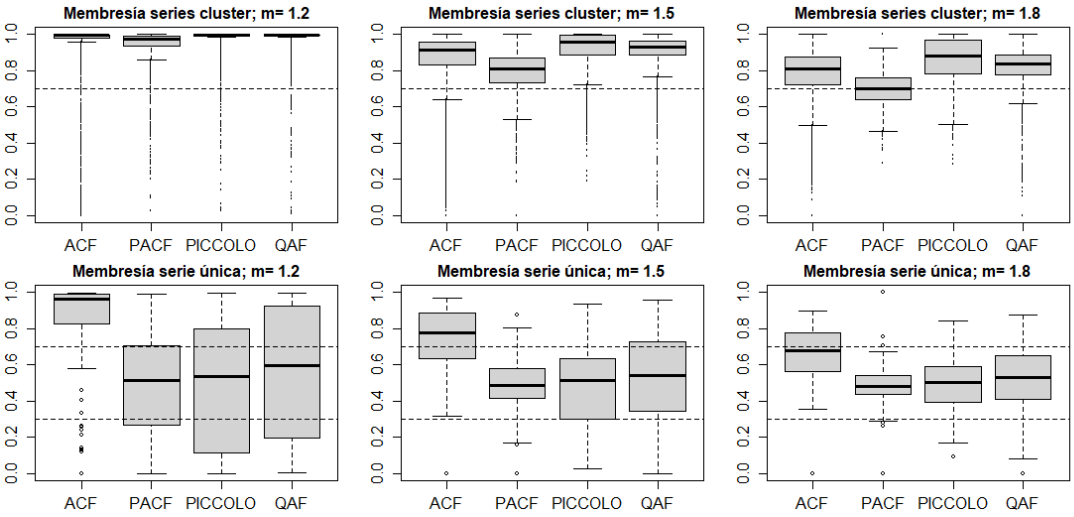


Figura 5.4: Diagramas de caja con grados de membresía retornados en el Escenario 1.C para $T = 200$. Para las series pertenecientes a los grupos (fila superior) se usan los grados de membresía asignados al cluster real de pertenencia, mientras para la serie equidistante (fila inferior) se usan los grados de membresía al Cluster C1.

Las cuatro métricas clasifican bien a las series de los clusters, con niveles de membresía al cluster verdadero muy elevados (raras veces por debajo de 0.7) y empeorando su conducta levemente cuando crece m. De nuevo Piccolo se revela como la mejor, con QAF y ACF generando resultados muy parecidos (la última con mayor desviación estándar). Por consiguiente, la serie equidistante no genera el ruido suficiente para que el algoritmo deje de identificar apropiadamente los clusters. Detectar la serie equidistante como tal es más complejo. Naturalmente, su identificación se propicia incrementando el valor de m, ya que esto conduce de por sí a niveles de membresía más equirepartidos. En general, se observa que ACF no trabaja bien pues tiende a ubicar a la serie equidistante en el Cluster C1 con cualquier valor de m. Por el contrario, PACF identifica muy bien la pertenencia de esta serie a los dos clusters (niveles entre 0.3 y 0.7), pero sin embargo es, de las cuatro, la que peor tasa de clasificación correcta presenta para las series de los clusters. Cabe así concluir que, en presencia de series "difusas", las métricas de Piccolo y QAF son preferidas igualmente, aunque en este caso es preferible emplear valores de m no demasiado pequeños.

Una herramienta útil para visualizar el nivel de separabilidad de los grupos y la capacidad de las métricas para identificarlos es el escalamiento multidimensional métrico (MDS). Dados n objetos y una matriz de distancias entre ellos, un escalado 2-dimensional (2DS) devuelve n puntos en $\mathbb{R}^2$ tales que las distancias Euclídeas entre ellos aproximan aquellas entre los objetos originales.

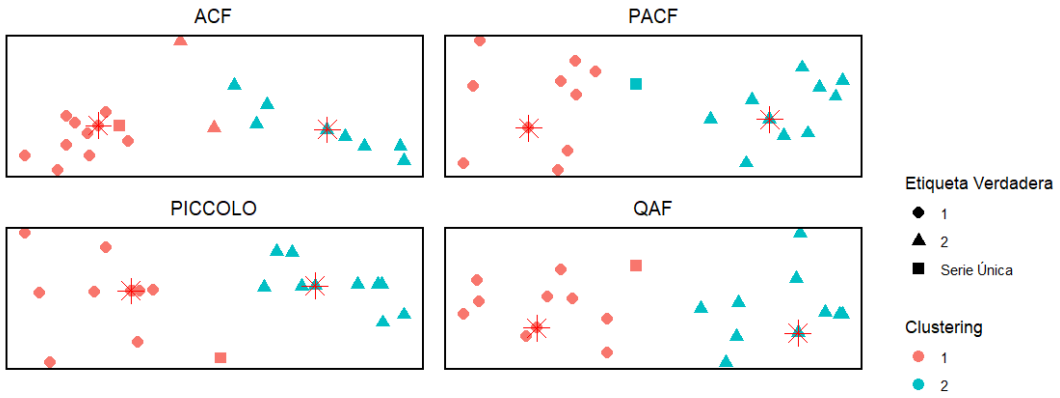


Figura 5.5: Realizaciones de una réplica arbitraria del Escenario 1.C proyectadas en un plano bidimensional utilizando escalamiento 2-dimensional.

Se realiza un 2DS sobre las realizaciones de una réplica arbitraria del Escenario 1.C empleando la función `cmdscale`, disponible en R. Los resultados para ACF, PACF, Piccolo y QAF se muestran en la Figura 5.5. La forma de los puntos indica la etiqueta verdadera, mientras que el color representa la clasificación resultante del clustering. Salvo ACF, las métricas forman grupos compactos y bien separados, mientras que la serie equidistante se ubica entre

los clusters, constatando el análisis de los diagramas de caja de la Figura 5.4.

5.3 Experimentos con procesos no lineales

5.3.1 Escenarios

En una segunda fase de los experimentos se realiza análisis cluster de series estacionarias generadas por modelos no lineales. La Tabla 5.5 proporciona la estructura de los escenarios considerados en este nuevo contexto.

Procesos Generadores	Escenario	Estructura
$X_t = \phi X_{t-1} - \phi\varepsilon_{t-1}X_{t-1} + \varepsilon_t$	2.A	Cluster C1: 10 series con $\phi \sim U(0.1, 0.2)$
		Cluster C2: 10 series con $\phi \sim U(0.4, 0.5)$
		Cluster C3: 10 series con $\phi \sim U(0.7, 0.8)$
	2.B	Cluster C1: 10 series con $\phi \sim U(0.4, 0.5)$
		Cluster C2: 10 series con $\phi \sim U(0.7, 0.8)$
		Una serie equidistante con $\phi = 0.6$
$X_t = (\alpha - 10\exp(-X_{t-1}^2))X_{t-1} + \varepsilon_t$	3.A	Cluster C1 : 10 series X_t con $\alpha \sim U(0.1, 0.4)$
$Y_t = \phi Y_{t-1} - \phi\varepsilon_{t-1}Y_{t-1} + \varepsilon_t$		Cluster C2: 10 series Y_t con $\phi \sim U(0.4, 0.6)$
$W_t = \rho\|W_{t-1}\|(3 + \|W_{t-1}\|)^{-1} + \varepsilon_t$		Cluster C3: 10 series W_t con $\rho \sim U(0.1, 0.4)$

Tabla 5.5: Escenarios de simulación clustering de modelos no lineales.

En todos los casos cada cluster está formado por 10 realizaciones. Los escenarios 2.A y 2.B contienen series con el mismo proceso generador, un modelo bilineal (BL), que depende del valor de un parámetro ϕ, el cual se mueve en distintos rangos para definir los clusters. El 2.A presenta 3 clusters y el 2.B sólo 2 clusters más una serie equidistante (a semejanza de los escenarios lineales 1.B y 1.C, respectivamente). El tercer y último escenario (3.A) consiste en 3 clusters definidos por diferentes procesos no lineales. Los modelos seleccionados han sido previamente considerados en la literatura en el marco de pruebas experimentales de cluster hard de series de tiempo [43]. Como en el caso lineal, se trata de escenarios propicios para usar distancias basadas en estructura y no en forma, previéndose en consecuencia un mal comportamiento cluster cuando se empleen DTW y la distancia Euclídea.

5.3.2 Resultados

Escenario 2.A.

Con carácter ilustrativo se representan en la Figura 5.6 las realizaciones de una de las réplicas simuladas del Escenario 2.A.

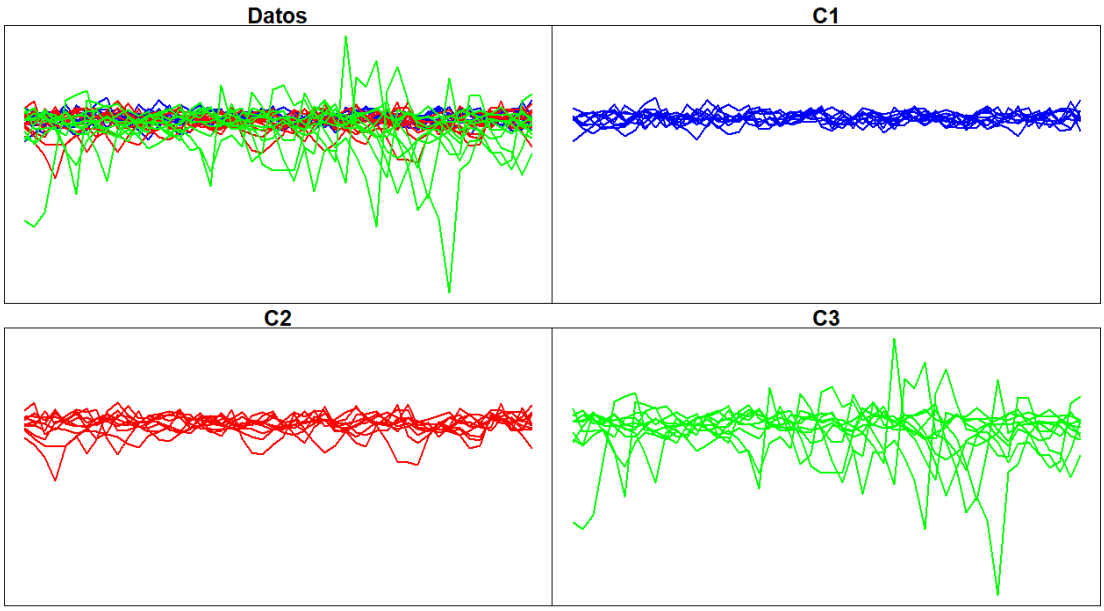


Figura 5.6: Realizaciones de una réplica del escenario de simulación 2.A.

Las tasas de series bien clasificadas para un umbral 0.6 se proporcionan en la Tabla 5.6.

Métrica	m	% Éxito (sd)
QAF	1.2	0.561 (0.137)
PICCOLO	1.2	0.530 (0.133)
PICCOLO	1.5	0.438 (0.121)
ACF	1.2	0.398 (0.147)
DTW	1.2	0.393 (0.069)
QAF	1.5	0.378 (0.150)
PICCOLO	1.8	0.378 (0.113)
EUCL	1.2	0.337 (0.089)
PACF	1.2	0.333 (0.133)
ACF	1.5	0.180 (0.106)
QAF	1.8	0.156 (0.090)
DTW	1.5	0.116 (0.108)
PACF	1.5	0.089 (0.059)
ACF	1.8	0.075 (0.042)
PACF	1.8	0.059 (0.030)
DTW	1.8	0.059 (0.024)
EUCL	1.5	0.056 (0.057)
EUCL	1.8	0.039 (0.014)

(a) T = 50

Métrica	m	% Éxito (sd)
QAF	1.2	0.892 (0.094)
QAF	1.5	0.823 (0.104)
PICCOLO	1.2	0.639 (0.103)
ACF	1.2	0.600 (0.131)
QAF	1.8	0.595 (0.133)
PACF	1.2	0.566 (0.122)
PICCOLO	1.5	0.529 (0.115)
ACF	1.5	0.450 (0.148)
PICCOLO	1.8	0.443 (0.094)
DTW	1.2	0.386 (0.042)
EUCL	1.2	0.290 (0.132)
PACF	1.5	0.286 (0.106)
ACF	1.8	0.230 (0.111)
PACF	1.8	0.111 (0.052)
DTW	1.5	0.088 (0.064)
DTW	1.8	0.058 (0.018)
EUCL	1.5	0.036 (0.011)
EUCL	1.8	0.034 (0.005)

(b) T = 200

Tabla 5.6: Porcentajes promedio de éxito del algoritmo FCMdC en el Escenario 2.A con diferentes métricas y niveles de solapamiento (m) para un umbral 0.6.

Los resultados con series cortas dejan entrever la mayor dificultad de este escenario comparado con los escenarios lineales de la Sección 5.2.1. En efecto, con $T = 50$, la métrica con mejor conducta, QAF, alcanza una muy pobre tasa de éxito de 0.56. Para obtener tasas de efec-

tividad cluster interesantes es necesario trabajar con series más largas. Con $T = 200$, QAF ya ofrece muy buen rendimiento, con tasas de éxito cercanas a las obtenidas en los escenarios lineales, dejando así patente su robustez al modelo generador una vez que dispone de series suficientemente largas. El resto de métricas lo hacen mal, incluso con series largas. La pobre conducta de Picccolo se explica por una mala especificación del modelo, ya que esta métrica se fundamenta en modelos subyacentes ARIMA. Análogamente, ACF y PACF persiguen estructuras de autocorrelación lineal que no están presentes en este escenario. Naturalmente, este análisis queda refrendado con los índices ARI.F y Jaccard.F (Figura A.3 en el Apéndice A).

Las nubes de puntos en la Figura 5.7 resultan de escalados bidimensionales basados en las matrices de distancia entre realizaciones de una réplica de este escenario. Es muy relevante observar que: (i) QAF es la única métrica capaz de formar grupos compactos y bien diferenciados, y (ii) las demás métricas identifican el cluster C1, pero no tienen capacidad de separar los grupos C2 y C3, generando clusters muy desbalanceados. Por ejemplo, con la métrica de Piccolo se forma un cluster C2 con 18 realizaciones y un cluster C3 con sólo una realización.

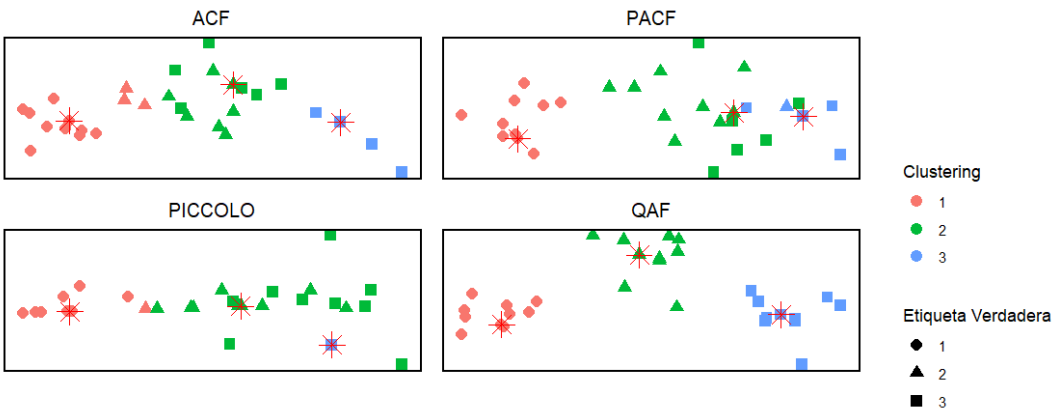


Figura 5.7: Series de una réplica arbitraria del Escenario 2.A proyectadas en un plano bidimensional utilizando 2DS.

Escenario 2.B.

Como en los experimentos con procesos lineales, el Escenario 2.B se configura con dos de los clusters del Escenario 2.A y una serie adicional equidistante de ambos. Como entonces, el objetivo es examinar la capacidad del procedimiento para detectar que una serie está a caballo de dos clusters. Del análisis del Escenario 2.A se concluyó que los clusters C2 y C3 son los más difícilmente separables y es por ello que son elegidos para conformar el Escenario 2.B, ahora con denominaciones respectivas C1 y C2 (ver Tabla 5.5). Realizaciones de una réplica arbitraria del Escenario 2.B se muestran en la Figura 5.8.

Para evaluar los resultados se procede de igual forma que en el análisis del Escenario 1.C.

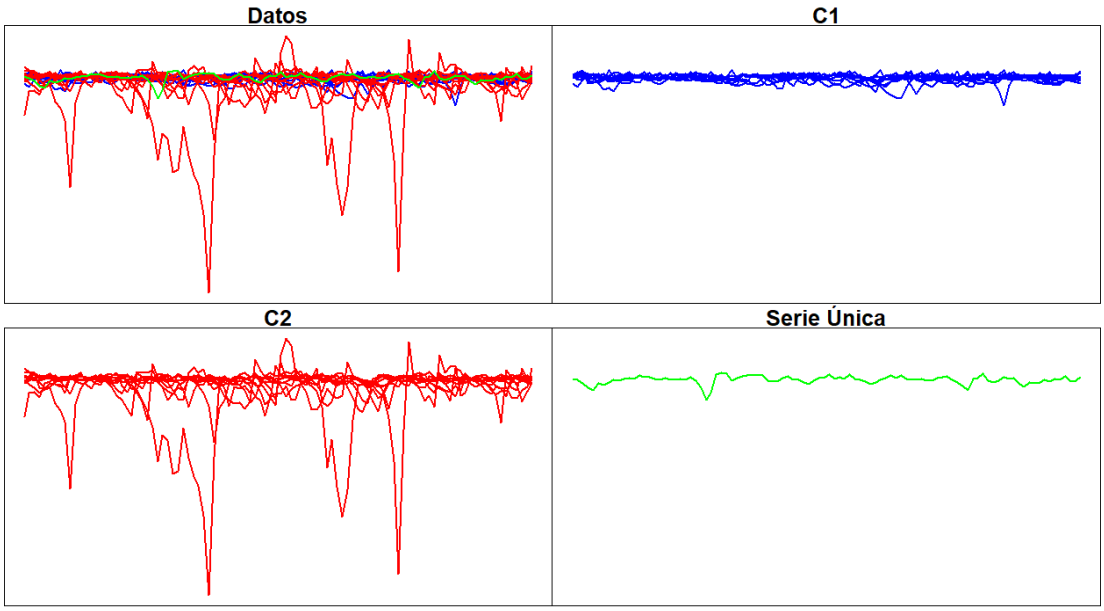


Figura 5.8: Realizaciones de una réplica del escenario de simulación 2.B.

Los correspondientes diagramas de caja con los grados de membresía usando series de longitud $T = 200$ se muestran en la Figura 5.9.

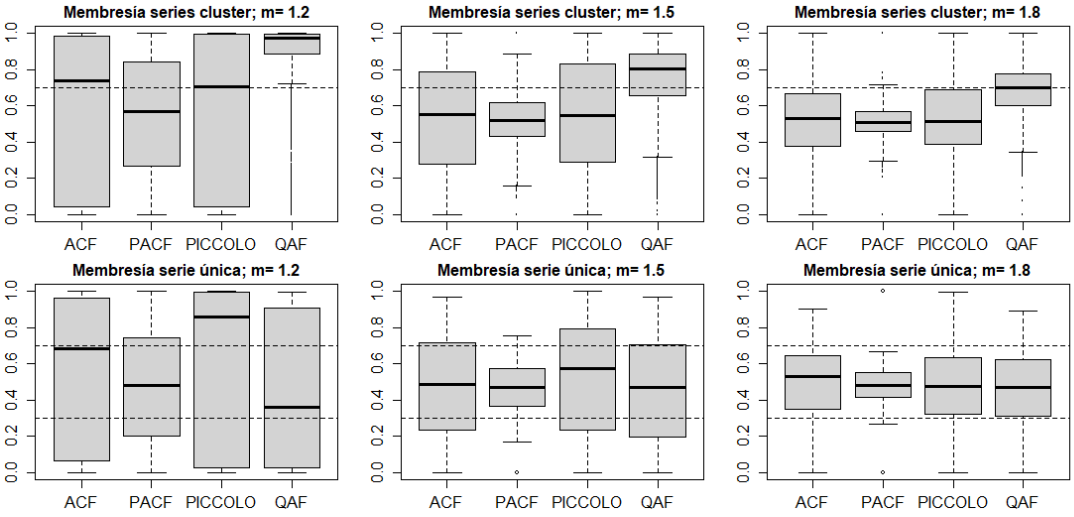


Figura 5.9: Diagramas de caja con grados de membresía retornados en el Escenario 2.B para $T = 200$. Para las series pertenecientes a los grupos (fila superior) se usan los grados de membresía asignados al cluster real de pertenencia, mientras para la serie equidistante (fila inferior) se usan los grados de membresía al Cluster C1.

En consonancia con el análisis del Escenario 2.A, la fila superior de la Figura 5.9 evidencia la total incapacidad de ACF, PACF y Piccolo para discriminar los clusters. Con $m = 1.2$ el

50% de los valores de membresía para el cluster real se mueven entre 0.1 y 0.9 con las métricas ACF y Piccolo. Tal rango intercuartílico es indicativo de la poca fiabilidad del resultado. PACF presenta menor dispersión, pero con una pobre mediana por debajo de 0.6. Por contra, QAF obtiene un excelente resultado con las series de los clusters cuando $m = 1.2$. Sin embargo, igual que con modelos lineales, $m = 1.2$ es un coeficiente de separabilidad excesivamente bajo para detectar series con conducta difusa, justificando así la incapacidad de las cuatro métricas para detectar que la serie adicional es equidistante de ambos clusters (gráfico de la izquierda en la fila inferior de la Figura 5.9). Aumentar el valor de m a 1.5 o 1.8 elimina este problema para las cuatro métricas, aunque solo QAF mantiene buenas tasas de correcta clasificación de las series de los clusters.

Las proyecciones 2DS en la Figura 5.10 corroboran la ventaja y buena conducta de la métrica QAF en orden a discrimnar entre los grupos e identificar la serie equidistante.

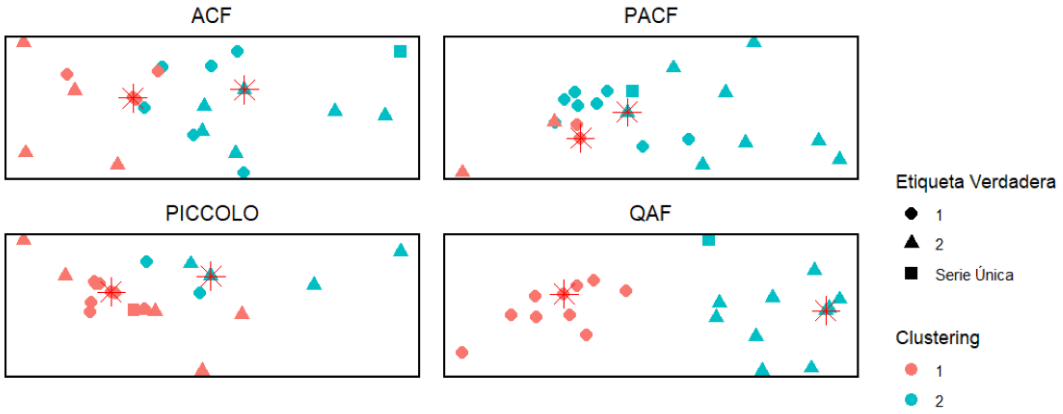


Figura 5.10: Series de una réplica específica del Escenario 2.B proyectadas en un plano bidimensional utilizando 2DS.

Escenario 3.A.

El último experimento considera un escenario muy general con 3 clusters generados por modelos no lineales con diferentes estructuras: un modelo autorregresivo exponencial (EXPAR), un modelo bilineal (BL) y un modelo no lineal general (ver Tabla 5.5). Como en escenarios anteriores, se muestra un ejemplo de realizaciones del Escenario 3.A en la Figura 5.11.

Los resultados de eficiencia cluster con las distintas métricas basados en tasas de clasificación correcta con un umbral 0.6 se proporcionan en la Tabla 5.7. La superioridad de QAF y el mal comportamiento del resto es de nuevo muy claro, quedando además refrendado si se consideran los valores ARI.F y Jaccard.F como índices de calidad cluster (ver Figura A.4 en el Apéndice A). El empleo de proyecciones 2DS con las matrices distancia de una de las réplicas ilustra esta conducta (ver Figura 5.12).

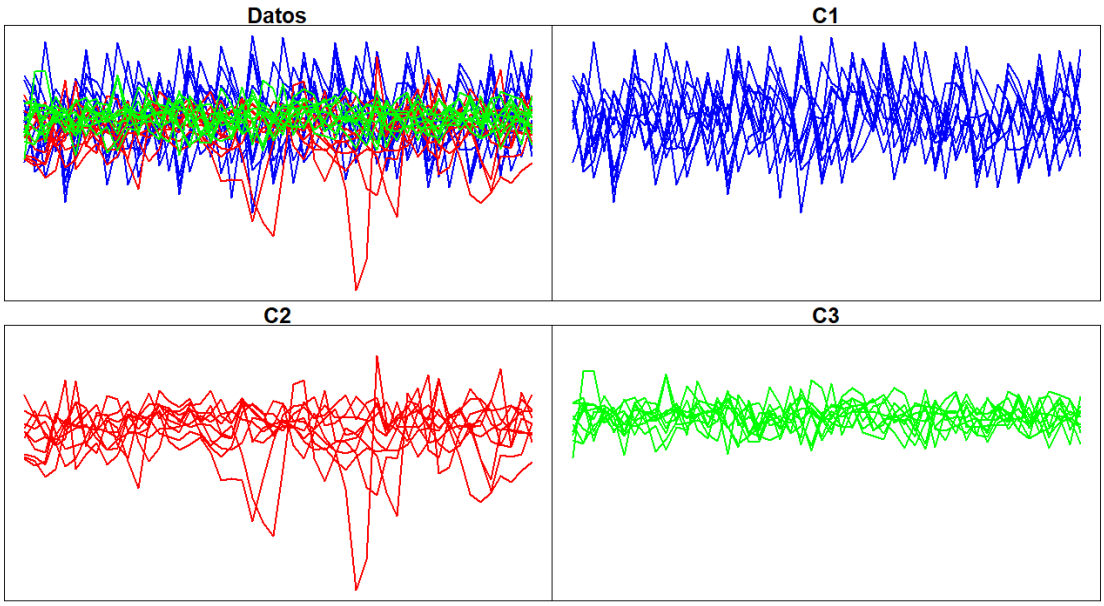


Figura 5.11: Realizaciones de una réplica del escenario de simulación 3.A.

Métrica	m	% Éxito (sd)
QAF	1.2	0.874 (0.111)
QAF	1.5	0.768 (0.095)
QAF	1.8	0.532 (0.115)
PICCOLO	1.2	0.461 (0.216)
ACF	1.2	0.387 (0.176)
PACF	1.2	0.373 (0.214)
PICCOLO	1.5	0.373 (0.199)
DTW	1.2	0.335 (0.188)
PICCOLO	1.8	0.327 (0.166)
ACF	1.5	0.162 (0.108)
PACF	1.5	0.130 (0.108)
ACF	1.8	0.082 (0.053)
PACF	1.8	0.071 (0.040)
EUCL	1.2	0.052 (0.047)
DTW	1.5	0.039 (0.016)
DTW	1.8	0.035 (0.007)
EUCL	1.5	0.034 (0.005)
EUCL	1.8	0.033 (0.000)

(a) T = 50

Métrica	m	% Éxito (sd)
QAF	1.2	0.997 (0.013)
QAF	1.5	0.988 (0.022)
QAF	1.8	0.926 (0.050)
PICCOLO	1.2	0.703 (0.220)
PACF	1.2	0.686 (0.248)
PICCOLO	1.5	0.627 (0.231)
PICCOLO	1.8	0.543 (0.216)
ACF	1.2	0.501 (0.152)
PACF	1.5	0.444 (0.209)
ACF	1.5	0.424 (0.158)
DTW	1.2	0.327 (0.209)
ACF	1.8	0.216 (0.132)
PACF	1.8	0.193 (0.109)
DTW	1.5	0.040 (0.013)
EUCL	1.2	0.034 (0.006)
DTW	1.8	0.034 (0.003)
EUCL	1.5	0.033 (0.000)
EUCL	1.8	0.033 (0.000)

(b) T = 200

Tabla 5.7: Porcentajes promedio de éxito del algoritmo FCMdC en el Escenario 3.A con diferentes métricas y niveles de solapamiento (m) para un umbral 0.6.

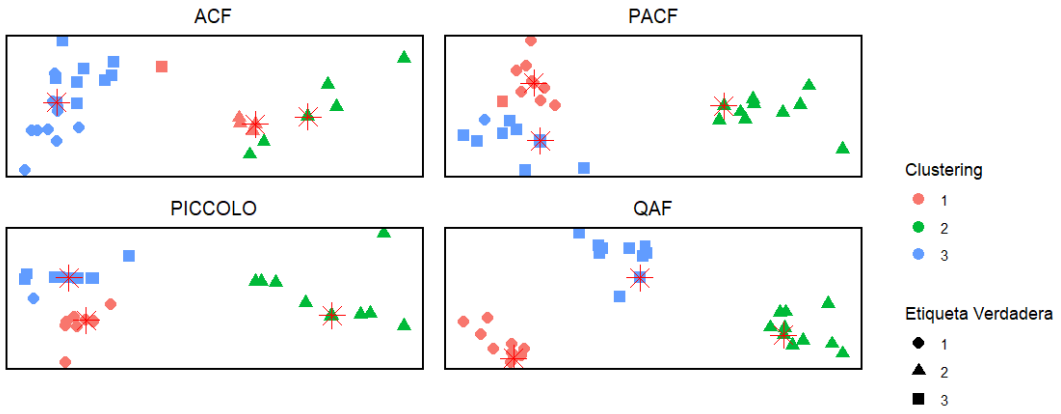


Figura 5.12: Series de una réplica arbitraria del Escenario 3.A proyectadas en un plano bidimensional utilizando 2DS.

5.4 Análisis del tiempo computacional

El análisis de la sección previa con datos simulados ha puesto de manifiesto la importancia de usar una u otra métrica en cluster difuso de series de tiempo según el tipo de patrones que se desean identificar. Ahora bien, en análisis cluster de series se trata a menudo con un volumen muy elevado de realizaciones, frecuentemente muy largas, lo que puede suponer un tiempo de computación excesivamente alto y limitar la aplicabilidad del procedimiento. De hecho, la eficiencia computacional podría ser un factor decisivo para elegir una u otra métrica si varias de ellas alcanzasen resultados satisfactorios.

Sobre la base de estos argumentos, se analiza en esta sección el tiempo de ejecución requerido para calcular la distancia entre dos series temporales. Nótese que una vez computada la matriz de distancias dos a dos, el algoritmo cluster sigue los mismos pasos para todas las métricas, de modo que es el cómputo de esta matriz el factor diferenciador para comparar la eficiencia computacional. El objetivo es por tanto identificar qué métricas son más eficientes en términos de tiempo de procesamiento y cuáles podrían resultar prohibitivamente lentas a medida que aumenta la longitud de las series.

Para un par de series AR(1) y utilizando diferentes valores para la longitud de las mismas, $T \in \{100, 500, 1000, 5000, 10000\}$, se han registrado los tiempos empleados con cada métrica para calcular la matriz de disimilitud dos a dos en un total de cien pruebas. En la Tabla 5.8 se muestran tiempos promedio y desviaciones típicas para cada métrica y cada longitud de serie.

El ordenador empleado fue un equipo de sobremesa LabMate3 con procesador Intel Core i7-7700 a 3.6 GHz y una memoria RAM de 32 GB. Los programas fueron escritos y ejecutados en RStudio, utilizando la versión 4.3.3 de R.

Es bien conocido que DTW es altamente costosa computacionalmente y este hecho se

Métrica	$T = 100$	$T = 500$	$T = 1000$	$T = 5000$	$T = 10000$
EUCL	0.001 (0.003)	0.001 (0.003)	0.001 (0.003)	0.003 (0.007)	0.006 (0.008)
DTW	0.002 (0.005)	0.020 (0.008)	0.072 (0.022)	2.256 (0.278)	8.534 (0.622)
ACF	0.001 (0.003)	0.001 (0.004)	0.001 (0.005)	0.002 (0.005)	0.003 (0.007)
PACF	0.001 (0.004)	0.001 (0.004)	0.001 (0.005)	0.003 (0.007)	0.006 (0.008)
PICCOLO	0.001 (0.005)	0.001 (0.004)	0.002 (0.008)	0.005 (0.008)	0.010 (0.008)
QAF	0.003 (0.007)	0.003 (0.008)	0.005 (0.010)	0.009 (0.009)	0.016 (0.008)

Tabla 5.8: Tiempos promedio de ejecución (en segundos) de cada métrica para obtener la matriz de disimilitud entre un par de series AR(1) sobre 100 ensayos y diferentes longitudes, T.

constata en los tiempos de ejecución obtenidos, observándose un drástico incremento con la longitud de las series. Las demás métricas requirieron tiempos de ejecución muy bajos, con un moderado incremento al crecer la longitud de las realizaciones.

Los tiempos reflejados en la Tabla 5.8 ilustran que, aunque DTW pueda llegar a ser efectiva en problemas de clustering basados en forma, su elevado costo computacional puede limitar su viabilidad en aplicaciones prácticas donde se manejen grandes volúmenes de datos. Por el contrario, el resto de métricas examinadas exhiben una alta escalabilidad.

Capítulo 6

Casos de estudio con datos reales

Tras haber analizado exhaustivamente el rendimiento de un rango de métricas en escenarios simulados, se procede en este capítulo a ilustrar su utilidad en un contexto práctico utilizando datos reales. Los casos de estudio considerados evidencian las ventajas de la flexibilidad del enfoque del cluster difuso y la importancia de emplear una u otra métrica según el tipo de series con las que se trata y los objetivos que persiguen con el análisis cluster.

6.1 Análisis cluster con series de demanda eléctrica

El primer caso de estudio tiene por objeto clasificar las horas del día de acuerdo a la evolución del cambio diario en la demanda eléctrica.

Se emplean datos de demanda de electricidad en España del año 2012, amablemente proporcionados por la Profesora Rebeca Peláez Suárez de la Universidad Carlos III de Madrid, y cuya fuente original es el Operador del Mercado Ibérico de Energía (OMIE) que, junto con Red Eléctrica de España (REE), constituyen las entidades reguladoras del Mercado Eléctrico en España. Una discusión más detallada de estos datos se proporciona en [44].

Se seleccionan las 24 series temporales registrando la demanda eléctrica en España durante los días laborables del año 2012 en cada hora del día, que se denotan por $X_{i,t}$, $i = 1, \ldots, 24$, $t = 1, \ldots, 261$, y se representan en la Figura 6.1.

Las series se transforman mediante $Y_{i,t} = \log(X_{i,t}/X_{i,t-1}) = \log X_{i,t} - \log X_{i,t-1}$, de forma que las nuevas series $Y_{i,t}$ representan la tasa logarítmica de variación diaria de la demanda eléctrica en la i-ésima hora y se representan en la Figura 6.2.

El objetivo es someter a análisis cluster a las 24 series $Y_{i,t}$ para identificar grupos de horas en el día con patrones específicos de variación diaria de demanda eléctrica. Nótese que:

- Es razonable un enfoque de cluster difuso porque cabe esperar que la variación de demanda eléctrica de unas horas a otras tenga una transición suave y no abrupta, de forma que algunas horas podrían estar a caballo de distintos clusters.

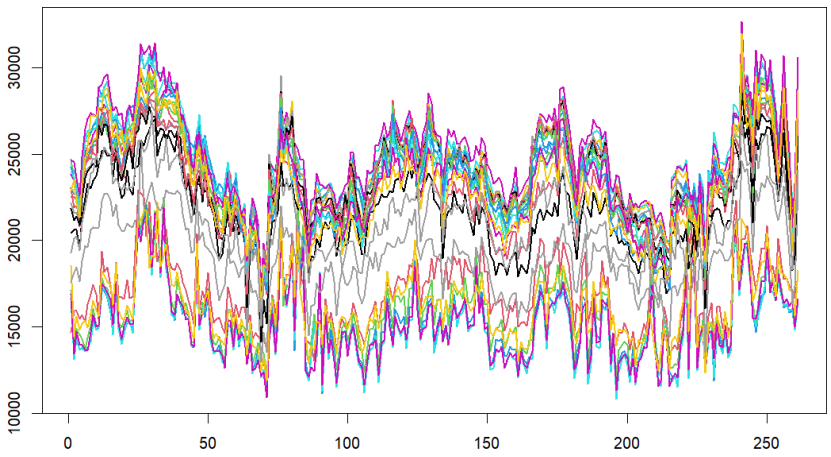


Figura 6.1: Series diarias de demanda eléctrica en España durante los días laborables del año 2012 para cada hora del día ($X_{i,t}$, $i = 1, \ldots, 24$).

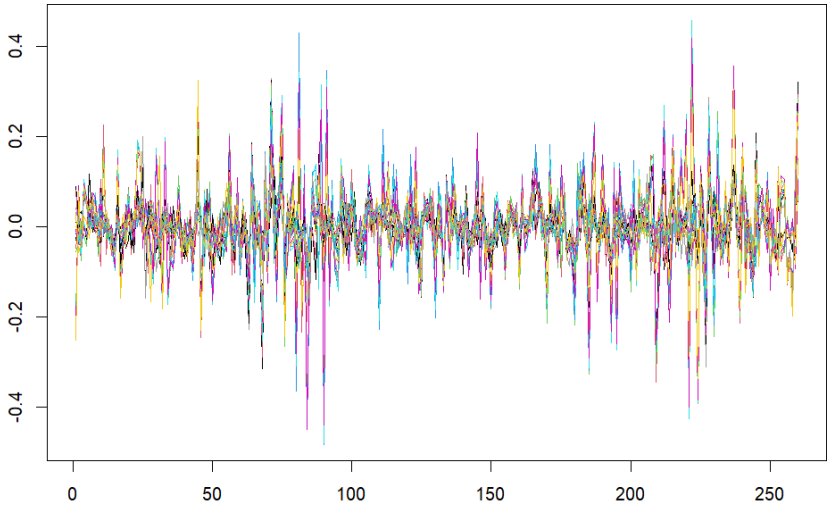


Figura 6.2: Series diarias de tasa logarítmica de variación diaria en la demanda eléctrica en España durante los días laborables del año 2012 para cada hora del día ($Y_{i,t} = \log(X_{i,t}/X_{i,t-1})$, $i = 1, \ldots, 24$).

- Los perfiles de las series en la Figura 6.2 sugieren que se trata de series estacionarias. Así, lo razonable es evaluar disimilitud entre estructuras dinámicas subyacentes. Emplear una métrica diseñada para discriminar en forma podría producir resultados inesperados.

De acuerdo con las consideracines previas, se decide realizar el análisis cluster con el algoritmo Fuzzy C-Medoids (FCMdC) basado en la distancia QAF que tan buen comportamiento mostró en los experimentos de simulación. Además, para ilustrar como una métrica en forma puede conducir a resultados menos razonables, se compararán los resultados con aquellos derivados de aplicar FCMdC con la métrica Euclídea.

Para seleccionar el número de clusters se emplea el *índice de silueta* (IS) [45], una medida de compacidad cluster que evalúa cómo de bien se agrupan los puntos dentro de un mismo grupo y qué tan separados están de otros grupos. La idea es obtener valores de IS con distintos números de grupos y seleccionar como valor óptimo K aquel que retorna el valor máximo del IS. Con la función `SIL.F`, de la librería **clust** de **R**, se obtuvieron valores de una extensión del índice de silueta al contexto difuso [46] que se representan en la Figura 6.3. Se concluye que, tanto con la distancia Euclídea como con QAF, el número de clusters apropiado es $K = 2$.

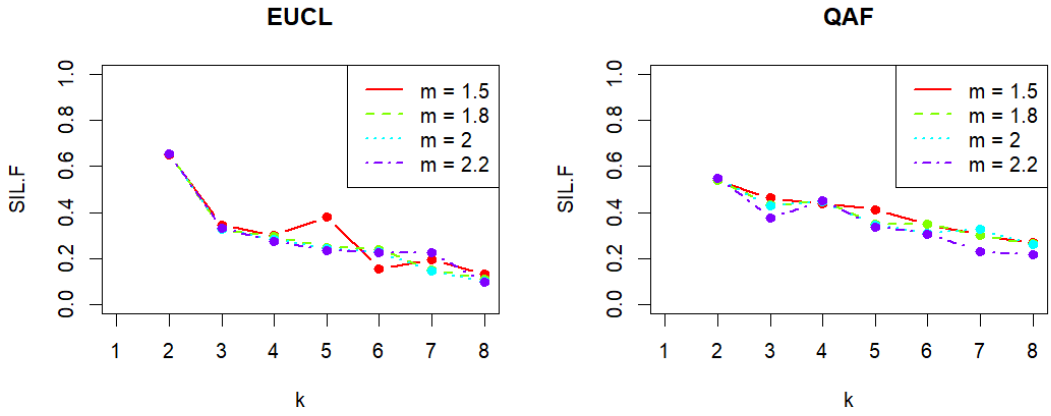


Figura 6.3: Valores del índice de silueta fuzzy para los datos de demanda eléctrica.

Las soluciones cluster tras aplicar FCMdC a las 24 series $Y_{i,t}$ con $K = 2, m = 2$ y usando las distancias Euclídea y QAF, se muestran en la Figura 6.4, donde se proporcionan los valores de membresía de cada serie (hora) a uno de los dos clusters.

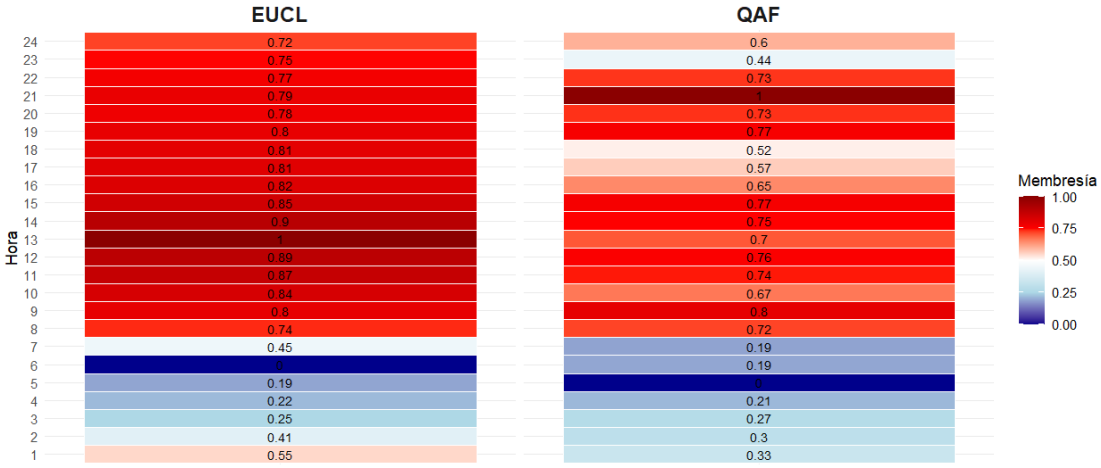


Figura 6.4: Fuzzy C-Medoids (FCMdC) con $K = m = 2$ de las 24 series $Y_{i,t}$: Valores de membresía a uno de los dos clusters con la distancia Euclídea y la distancia QAF.

Aún a pesar de usar $m = 2$, FCMdC con la distancia Euclídea rompe el conjunto de series de forma abrupta en dos grupos de horas muy compactos (de 08:00 a 24:00 y de 03:00 a 06:00), con niveles de pertenencia al cluster dominante por encima de 0.7. Solo las series correspondientes a la 01:00, 02:00 y 07:00 horas presentan una conducta a caballo de los dos clusters mencionados, lo que no parece en absoluto razonable en el caso de las dos primeras.

FCMdC con QAF conduce a una partición más difusa y acorde con lo esperado. Hay un cluster compacto con las horas de menor demanda y menor variación intradía (de 01:00 a 07:00), pero en el segundo cluster, con horas de mayor demanda, se observan series con niveles de membresía equirepartidos que confieren a esas franjas horarias una posición intermedia entre las conductas de ambos clusters. Este es el caso por ejemplo del "valle" en los niveles de membresía de las series de las 17:00 y 18:00 horas, donde comienza a ceder el pico de demanda de la actividad laboral, y de la transición suave hacia el cluster de baja demanda de los niveles de membresía para las horas finales del día, 23:00 y 24:00. Más allá de estas observaciones, las tonalidades menos acentuadas de rojo para el cluster de mayor demanda con QAF evidencian la transición más difusa de los niveles de membresía con esta métrica.

6.2 Análisis cluster con series de datos meteorológicos

El segundo caso estudio concierne al problema de identificar lugares con similares patrones de evolución de temperatura. Se emplean datos meteorológicos reales de la Agencia Estatal de Meteorología (AEMET), que consisten en series de temperaturas diarias promedio de varias estaciones meteorológicas en España a lo largo de un período extenso (1980-2009). Los datos se obtienen de la librería **fda.usc** [47] de **R** y, aunque la base de datos original contiene series de 73 estaciones, se trabaja aquí con una muestra de 25 para facilitar la interpretación y visualización de los resultados (ver Figura 6.5).

Las series son claramente no estacionarias y como el interés radica en agrupar estaciones con similar perfil de temperatura diaria promedio, se decide realizar cluster difuso (FCMdC) usando métricas basadas en forma. En concreto, se usan las distancias Euclídea y DTW. Para ilustrar la versatilidad del cluster fuzzy, previamente se ha sometido a las series a un cluster estándar, usando el algoritmo PAM y estas mismas métricas.

Se emplea de nuevo la versión fuzzy del índice de silueta para determinar el número apropiado de grupos. Como se muestra en la Figura 6.6, el valor óptimo se tiene en $K = 4$ clusters, tanto para la distancia Euclídea como para DTW y con independencia del valor del coeficiente de solapamiento m.

Las particiones con el algoritmo PAM se proporcionan en la Figura 6.7. Se obtienen los mismos resultados con las dos métricas, con la única excepción de la localización de la estación Zaragoza (Aeropuerto). Además de esta diferencia, resulta inesperada la ubicación de la

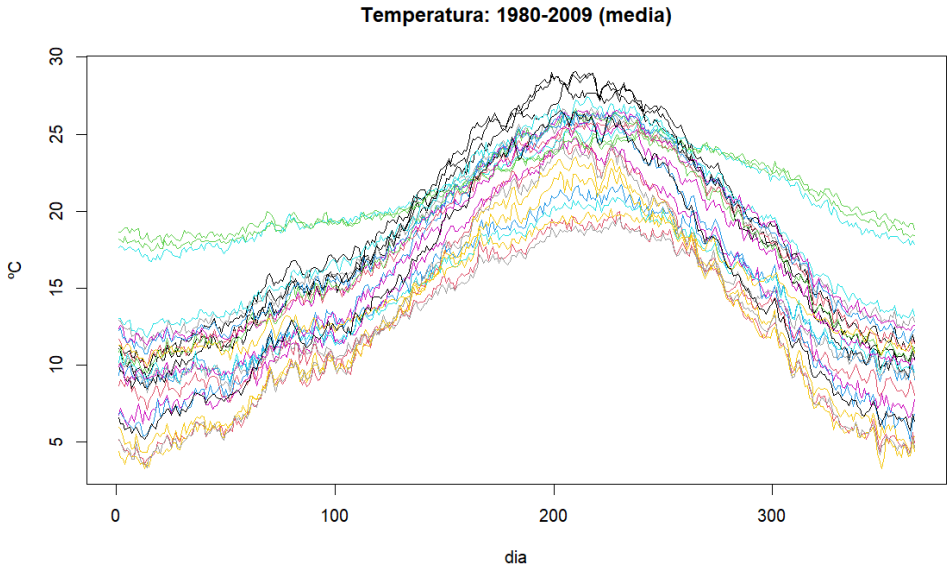


Figura 6.5: Series de temperatura diaria promedio de 25 estaciones meteorológicas españolas a lo largo del período 1980-2009.

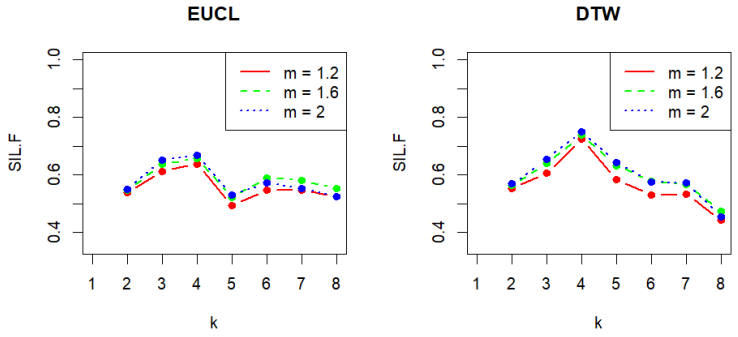


Figura 6.6: Valores del índice de silueta fuzzy para las 25 series de temperatura diaria promedio.

serie de temperatura de Girona/CostaBrava en el cluster formado por localidades del norte occidental de la península.

Las Tablas 6.1 y 6.2 muestran los niveles de membresía de la solución fuzzy para $m = 1.6$. Con la distancia Euclídea, Zaragoza (Aeropuerto) ya no aparece en el cluster de localidades próximas a la costa cantábrica, aunque sí lo sigue haciendo Girona/Costa Brava. Con DTW, Girona/Costa Brava desaparece de este cluster que pasa a estar formado únicamente por las localidades del norte occidental. La flexibilidad del enfoque fuzzy permite observar como estas dos localidades presentan niveles de membresía más repartidos entre varios clusters que el resto de series, lo cual es mucho más informativo que la salida del algoritmo PAM que no distingue en absoluto si la ubicación en un cluster es más o menos periférica.

(a) Distancia Euclídea (b) DTW

Figura 6.7: Particiones cluster con el algoritmo PAM y las distancias Euclídea (a) y DTW (b).

Nótese que, pese a usar $m = 1.6$ con ambas métricas, la distancia Euclídea conduce a un escenario con mayor incertidumbre, con valores de membresía más repartidos. Por ejemplo, los valores de membresía para Zaragoza y Girona están muy repartidos entre tres grupos. Por el contrario, DTW incrementa sustancialmente el grado de pertenencia de ambas localidades al cluster con centroide Cuenca (0.6375 para Girona y 0.6855 para Zaragoza) y asigna el resto de series a clusters únicos de forma más nítida, con la mayoría de niveles de pertenencia superiores a 0.8. La solución alcanzada con DTW parece por tanto mucho más razonable y acorde con lo esperado.

Estación	CUENCA	ASTURIAS/AVILÉS	LANZAROTE/AEROPUERTO	VALENCIA/AEROPUERTO
CUENCA	1	0	0	0
PAMPLONA/NOAIN	0.7238	0.1984	0.0239	0.0539
ZARAGOZA (AEROPUERTO)	0.3827	0.3129	0.0672	0.2372
VALLADOLID	0.8942	0.0696	0.0112	0.025
DAROCA	0.9485	0.0336	0.0054	0.0125
MADRID, RETIRO	0.4982	0.2732	0.0548	0.1739
TENERIFE/SUR	0.0109	0.0182	0.9315	0.0394
LANZAROTE/AEROPUERTO	0	0	1	0
LAS PALMAS DE GRAN CANARIA/GANDO	0.0061	0.0104	0.96	0.0235
MURCIA/ALCANTARILLA	0.0509	0.078	0.0639	0.8073
MÁLAGA/AEROPUERTO	0.0165	0.0311	0.0363	0.9161
SEVILLA/SAN PABLO	0.0687	0.0989	0.1345	0.6979
ALMERÍA/AEROPUERTO	0.0335	0.0589	0.1047	0.8029
PALMA DE MALLORCA, CMT	0.0136	0.026	0.0241	0.9362
CÓRDOBA/AEROPUERTO	0.1018	0.1343	0.1051	0.6589
MURCIA/SAN JAVIER	0.0293	0.0606	0.0355	0.8746
VALENCIA/AEROPUERTO	0.0408	0.076	0.0436	0.8396
ALICANTE/EL ALTET	0	0	0	1
PALMA DE MALLORCA/SON SAN JUAN	0.1448	0.2942	0.0763	0.4847
BILBAO/AEROPUERTO	0	1	0	0
GIRONA/COSTA BRAVA	0.4016	0.4218	0.0429	0.1336
OVIEDO	0.2128	0.6616	0.0411	0.0845
SANTANDER/PARAYAS	0.0407	0.9162	0.0128	0.0304
A CORUÑA	0.0934	0.7638	0.0454	0.0974
ASTURIAS/AVILÉS	0.1662	0.6977	0.0457	0.0905

Tabla 6.1: Grados de membresía con FMCdC usando la distancia Euclídea y $m = 1.6$.

La Figura 6.8 agrupa los perfiles de las series temporales según los grupos resultantes del

Estación	CUENCA	LANZAROTE/AEROPUERTO	VALENCIA/AEROPUERTO	ASTURIAS/AVILÉS
CUENCA	1	0	0	0
GIRONA/COSTA BRAVA	0.6375	0.0184	0.2002	0.1439
PAMPLONA/NOAIN	0.8275	0.0093	0.0548	0.1084
ZARAGOZA (AEROPUERTO)	0.6855	0.0201	0.2198	0.0746
VALLADOLID	0.917	0.0059	0.0333	0.0439
DAROCA	0.9278	0.0052	0.0334	0.0336
MADRID, RETIRO	0.7061	0.0187	0.2	0.0752
TENERIFE/SUR	0.0053	0.9718	0.0181	0.0047
LANZAROTE/AEROPUERTO	0	1	0	0
LAS PALMAS DE GRAN CANARIA/GANDO	0.0033	0.9816	0.0119	0.0031
MURCIA/ALCANTARILLA	0.0423	0.0271	0.8961	0.0346
MÁLAGA/AEROPUERTO	0.041	0.0791	0.8328	0.0471
SEVILLA/SAN PABLO	0.0517	0.058	0.8465	0.0438
ALMERÍA/AEROPUERTO	0.0472	0.1402	0.7613	0.0513
PALMA DE MALLORCA, CMT	0.0369	0.0536	0.8692	0.0402
CÓRDOBA/AEROPUERTO	0.0918	0.0397	0.8136	0.0549
MURCIA/SAN JAVIER	0.031	0.0236	0.9083	0.0371
VALENCIA/AEROPUERTO	0	0	1	0
ALICANTE/EL ALTET	0.0303	0.0398	0.8963	0.0337
PALMA DE MALLORCA/SON SAN JUAN	0.0664	0.0171	0.8562	0.0604
BILBAO/AEROPUERTO	0.0927	0.0162	0.148	0.7431
OVIEDO	0.0702	0.007	0.0482	0.8746
SANTANDER/PARAYAS	0.0503	0.0105	0.0852	0.854
A CORUÑA	0.0426	0.0149	0.0921	0.8504
ASTURIAS/AVILÉS	0	0	0	1

Tabla 6.2: Grados de membresía con FMCdC usando DTW y $m = 1.6$

proceso de clustering (con la distancia Euclídea en la fila superior y con DTW en la inferior). La simple inspección visual de estos gráficos confirma la uniformidad en forma de cada uno de los grupos y las diferencias entre ellos.

La mayor discrepancia se observa al ubicar la serie de Girona/Costa Brava en el grupo de las localidades del norte occidental (ver figura a la derecha de la fila superior). Esta serie muestra temperaturas notablemente más altas en los días intermedios del año y colas más pesadas que le resto de series en este grupo, resultando más afín a las series del Grupo 2 (con centroide Cuenca). Justamente este es el cambio que aplica DTW en la fila inferior, produciendo una mayor coherencia interna de los grupos.

6.2.1 Herramienta de visualización

Como este segundo caso de estudio trata con estaciones meteorológicas de las que se disponen sus coordenadas geográficas, se ha desarrollado una pequeña página web para visualizar los resultados. Se utiliza una combinación de tecnologías que facilitan la gestión, almacenamiento y visualización de los datos espaciales.

Específicamente, se emplea PostgreSQL con la extensión PostGIS para el almacenamiento de los datos espaciales. PostgreSQL es un sistema de gestión de bases de datos relacional que permite manejar grandes volúmenes de datos de manera eficiente, mientras que PostGIS añade capacidades geoespaciales, permitiendo almacenar y manipular datos espaciales directamente en la base de datos. La estructura de las tablas en la base de datos sigue la forma presentada en la Tabla 6.3. Los datos almacenados incluyen la identificación de la estación, el cluster al

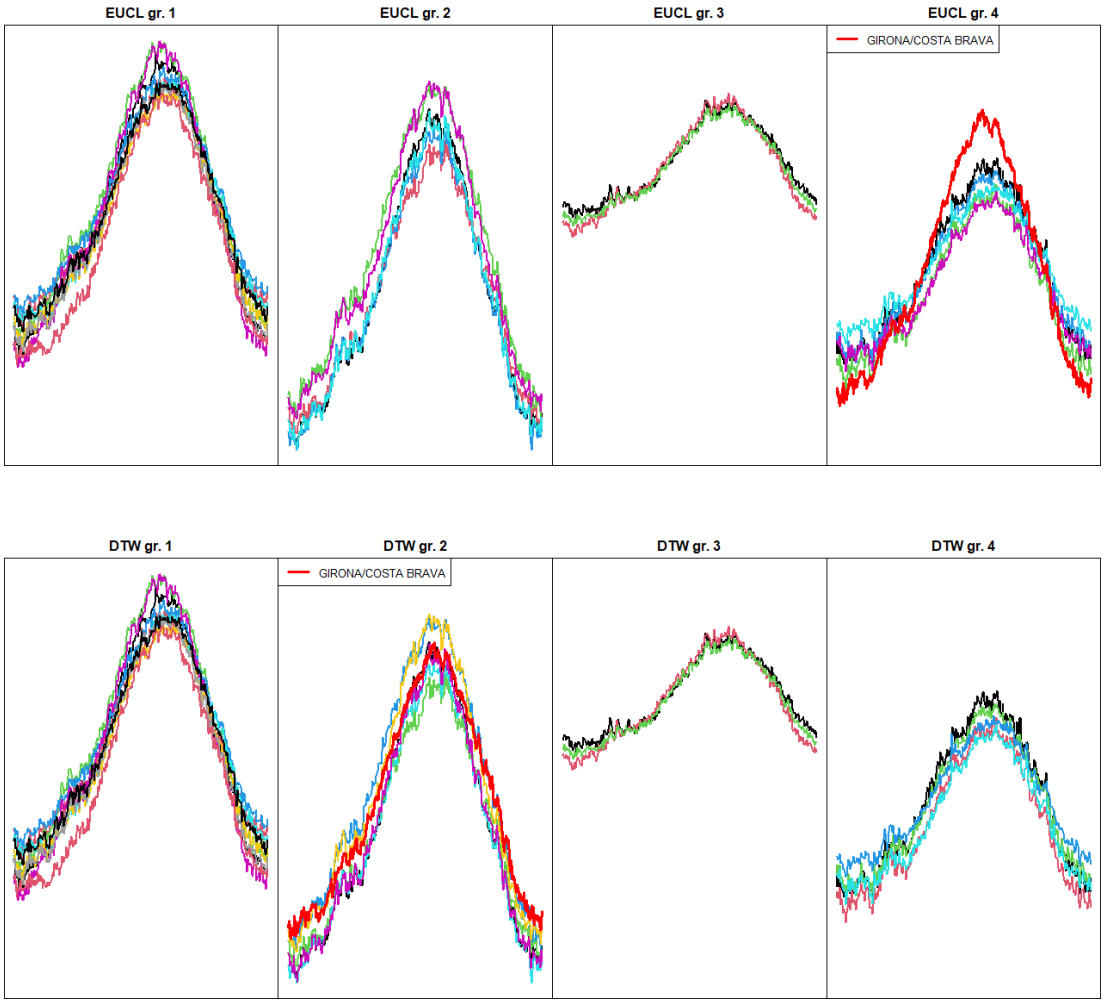


Figura 6.8: Perfiles de las series agrupados de acuerdo a la solución reportada por FCMdC con $m = 1.6$ y las distancias Euclídea (fila superior) y DTW (fila inferior).

que pertenece, el grado de pertenencia y la información espacial en formato POINT.

id	estación	cluster	membresía	longitud	latitud	geometry
1	MURCIA/ALCANTARILLA	4	0.8073	-1.2297	37.9578	POINT (-1.2297 37.9578)
2	CUENCA	1	1.0000	-2.1381	40.0667	POINT (-2.1381 40.0667)
3	TENERIFE/SUR	3	0.9315	-16.5608	28.0475	POINT (-16.5608 28.0475)
4	BILBAO/AEROPUERTO	2	1.0000	-2.9058	43.2981	POINT (-2.9058 43.2981)

Tabla 6.3: Estaciones meteorológicas: identificación, cluster, niveles de membresía y ubicación geográfica.

Para la visualización de los resultados en un mapa interactivo, se optó por GeoServer, una

plataforma de código abierto que permite servir datos espaciales y visualizarlos en la web. GeoServer se configuró para acceder a la base de datos PostgreSQL/PostGIS y proporcionar servicios WMS (Web Map Service), que permiten que los datos geográficos sean consumidos y visualizados en la página web. Finalmente, se utilizó Leaflet, una biblioteca JavaScript, para crear el mapa interactivo en la página web.

Las Figuras 6.9 y 6.10 ilustran el resultado de la herramienta de visualización creada. A través de los botones disponibles en la interfaz se pueden activar las diferentes capas que muestran los resultados del particionado para cada métrica. Las Tablas 6.1 y 6.2 con los grados de membresía son accesibles a través de los correspondientes hipervínculos.

Para obtener información de la visualización del mapa se han empleado diferentes estrategias gráficas. Por ejemplo, el grupo de pertenencia de cada estación meteorológica se indica a través de su color, de modo que puntos de igual color se clasifican en el mismo cluster. El tamaño de los puntos varía según el grado de pertenencia al cluster, de forma que cuánto más grande sea el círculo, mayor es el nivel de pertenencia devuelto por el procedimeinto cluster. Por último, se añade un borde de color negro a aquellos puntos definidos como medoides (grado de pertenencia igual a uno).

Esta herramienta permite un análisis rápido e intuitivo de los resultados del clustering para las diferentes métricas utilizadas. Obviamente, es particularmente llamativo el cambio de grupo (color) de la estación Girona/Costa Brava. La ventaja adicional de la herramienta es que permite observar la coherencia de los grupos creados con su ubicación geográfica, lo cual es esperable cuando se mide la evolución de temperaturas. Así, se distingue la formación de grupos en la costa norte occidental, la costa mediterránea, el interior de la península y el archipiélago canario. Estos agrupamientos reflejan patrones climáticos que son consistentes con la geografía y las características climáticas de cada región, confirmando la efectividad del enfoque utilizado para la clasificación.

El grado de certidumbre en la asignación al cluster también se percibe fácilmente con la inspección visual del mapa. Por ejemplo, todas las estaciones formando el grupo del archipiélago canario se grafican con puntos grandes, informando así que han sido asignadas a ese cluster con un alto grado de membresía y, por tanto, sin incertidumbre. Nótese que, en general, los puntos con la métrica DTW tienden a ser más grandes que aquellos con la distancia Euclídea, corroborando así la observación realizada anteriormente de que DTW conduce a una solución menos difusa.

Figura 6.9: Estaciones distancia euclídea

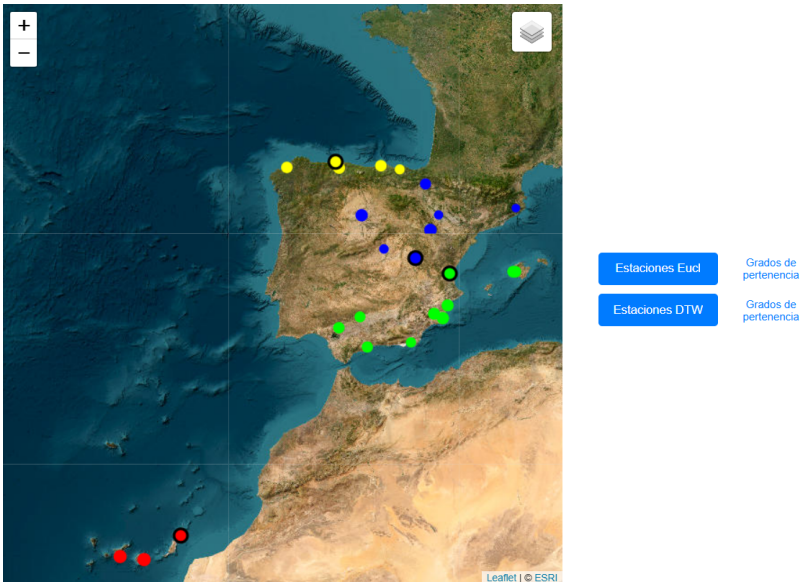


Figura 6.10: Estaciones DTW

Capítulo 7

Conclusiones

En el presente estudio se ha abordado el análisis cluster difuso de series temporales, un tópico de gran interés y aplicabilidad en muchas áreas. Más específicamente, se ha analizado el comportamiento del algoritmo Fuzzy C-Medoids (FCMdC) considerando diferentes métricas para evaluar disimilitud entre realizaciones de series temporales. Como se ha argumentado e ilustrado a través de sencillos ejemplos, definir una distancia entre series de tiempo, objetos de naturaleza dinámica, no es trivial y una selección adecuada de esta distancia puede resultar clave para el éxito o fracaso de la tarea cluster.

Tras presentar al algoritmo FCMdC como un problema de optimización envolviendo una distancia entre objetos, se ha introducido un abanico de métricas propuestas en la literatura para discriminar entre series según sus perfiles geométricos (distancias basadas en forma) o según sus modelos de dependencia subyacentes (distancias basadas en estructura). La elección de uno u otro criterio depende de la naturaleza de las series sometidas a cluster y del propósito del agrupamiento. Entre las métricas basadas en forma se han considerado la distancia Euclídea y la conocida como Dynamic Time Warping (DTW). Para discriminar entre procesos generadores estacionarios, se han descrito distancias libres de modelo basadas en comparar características seriales, a saber autocorrelaciones simples (ACF), autocorrelaciones parciales (PACF) y autocovarianzas cuantil (QAF), y una distancia basada en modelos autorregresivos (Piccolo). Se ha puesto especial énfasis en subrayar los puntos fuertes y las limitaciones de las métricas revisadas.

En este marco de trabajo, se procedió a examinar el rendimiento en términos de eficiencia cluster del algoritmo FCMdC con las métricas mencionadas a través de un amplio estudio de simulación diseñado para discriminar entre series generadas desde procesos estacionarios, incluyendo tanto modelos lineales como no lineales. El análisis cluster de series estacionarias es particularmente retante por la dificultad de capturar diferencias estructurales entre modelos a menudo muy parecidos y donde un gráfico de los perfiles de las series frecuentemente confunde más que ayuda. La eficiencia del clustering se evaluó considerando diferentes criterios

y los resultados alcanzados permiten establecer las siguientes conclusiones:

- A diferencia del análisis cluster estándar (hard cluster), el cluster fuzzy es lo suficientemente versátil para detectar la ubicación de series individuales en más de un cluster si una métrica apropiada es seleccionada. Lo hace además sin tener un efecto distorsionador sobre los grados de membresía del resto de series, aquellas pertenecientes a un único cluster.

- Al igual que en análisis cluster fuzzy de datos estáticos, seleccionar un coeficiente de solapamiento m adecuado es importante para tener resultados satisfactorios. Disponer de un criterio automático para su selección es un tema abierto y de gran interés.

- Como se esperaba, las distancias basadas en forma (Euclídea y DTW) se han mostrado de todo punto ineficaces para discriminar entre series de tiempo generadas de procesos estacionarios.

- Entre las distancias consideradas basadas en estructura, la distancia de Piccolo trabajó bien con procesos lineales, pero no así con modelos no lineales, donde sufrió de la mala especificación del modelo y puso en evidencia el riesgo inherente a usar métricas basadas en modelos con carácter general.

- Las métricas ACF y PACF mostraron un comportamiento razonable, pero claramente inferior a la métrica basada en autocovarianzas cuantil (QAF). La conclusión es que las autocovarianzas cuantil proporcionan una foto más precisa de las estructuras de dependencia subyacentes lo que permite incrementar su capacidad discriminatoria. QAF fue la única que mostró robustez al tipo de proceso generador, siendo incluso competitiva con la distancia de Piccolo en escenarios autorregresivos.

- Las conductas mencionadas se observaron con independencia de la longitud de las series, aunque obviamente las tasas de éxito en la clasificación fueron más elevadas con series más largas.

- La proyección de las distancias sobre planos 2DS (escalados métricos bidimensionales) permitió visualizar y corroborar la diferente capacidad discriminatoria de las métricas estudiadas.

- En análisis cluster de series es frecuente manejar grandes volúmenes de series largas, de modo que es importante examinar no solo la eficacia cluster sin también la complejidad computacional. El resultado del examen de tiempos de computación permitió concluir que todas las métricas analizadas resultan eficientes a excepción de DTW, que además sufre una disminucion significativa de la eficiencia a medida que aumenta la longitud de las series.

Además del análisis con datos simulados, se ha realizado análisis cluster con dos bases de datos reales, incluyendo respectivamente series de datos diarios de demanda eléctrica para cada hora del día y series diarias de temperatura promedio en distintas localizazaciones geográficas. Se seleccionaron estos dos casos de estudio por tratar con objetivos cluster diferentes: agrupar estructuras de dependencia en el caso de las series de demanda eléctrica y perfiles de forma en el caso de las series de temperatura. En ambos casos, los resultados alcanzados supusieron:

- Evidenciar la utilidad del enfoque fuzzy en la práctica. Con las series de demanda eléctrica fue factible examinar la transición suave de los grados de membresía para determinadas horas del día, en tanto que con las series de temperatura se detectaron grupos con perfiles de temperatura más alejados del patrón general definiendo el cluster de pertenencia.

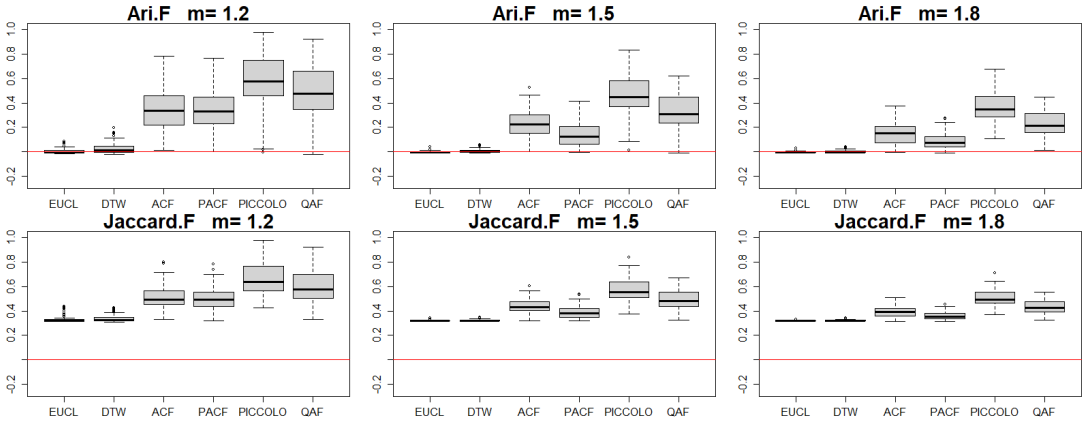
- Concluir la necesidad de utilizar una métrica acorde con el propósito cluster y, una vez fijado este objetivo, comprobar que unas métricas arrojan resultados más apropiados que otras. En los casos de estudio considerados, QAF mejoró claramente los resultados de la distancia Euclídea con los datos de demanda eléctrica, en tanto que DTW condujo a unos resultados más acordes con lo esperado en el caso de los datos meteorológicos.

En definitiva, el estudio llevado a cabo confirma el interés en el tópico abordado y pone de manifiesto la enorme utilidad que un enfoque fuzzy puede tener en muchas aplicaciones, así como la importancia de seleccionar una métrica apropiada para discriminar entre series. El estudio experimental aporta alguna luz sobre la conducta de algunas métricas, pero es muy importante subrayar que otras muchas han sido propuestas en la literatura. Se trata de un campo abierto de estudio con muchos retos por abordar todavía, tales como el problema de detectar el número de grupos subyacentes, determinar el coeficiente de solapamiento apropiado, medir robustez de los criterios de distancia para modelos complejos (por ejemplo series heterocedásticas) y realizar cluster de series multivariantes, entre otros.

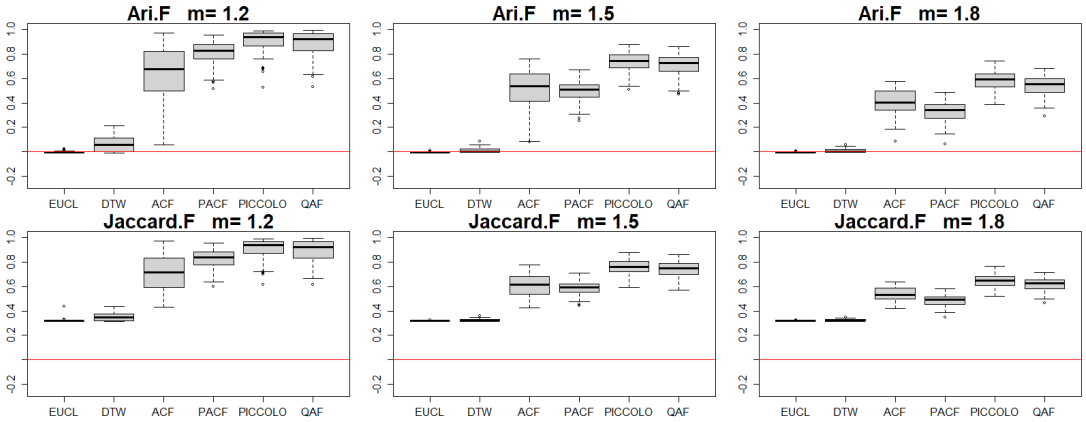
Apéndices

Apéndice A

Material adicional

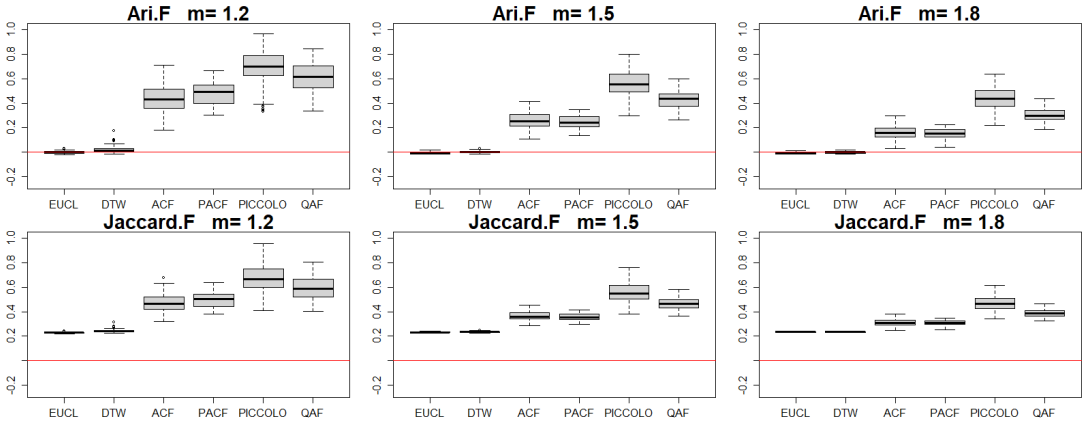


(a) T = 50

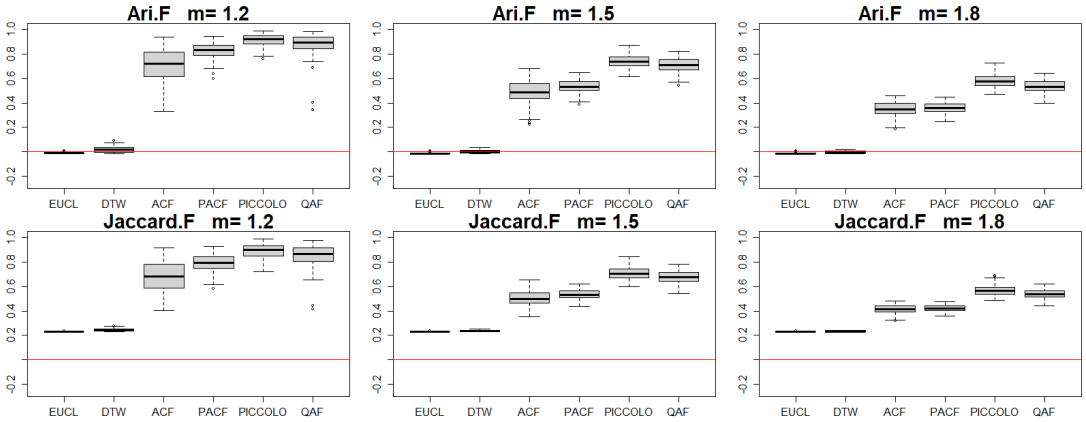


(b) T = 200

Figura A.1: Diagramas de caja de los índices de calidad cluster ARI.F y Jaccard.F basados en las 100 réplicas del algoritmo FCMdC en el Escenario 1.A.

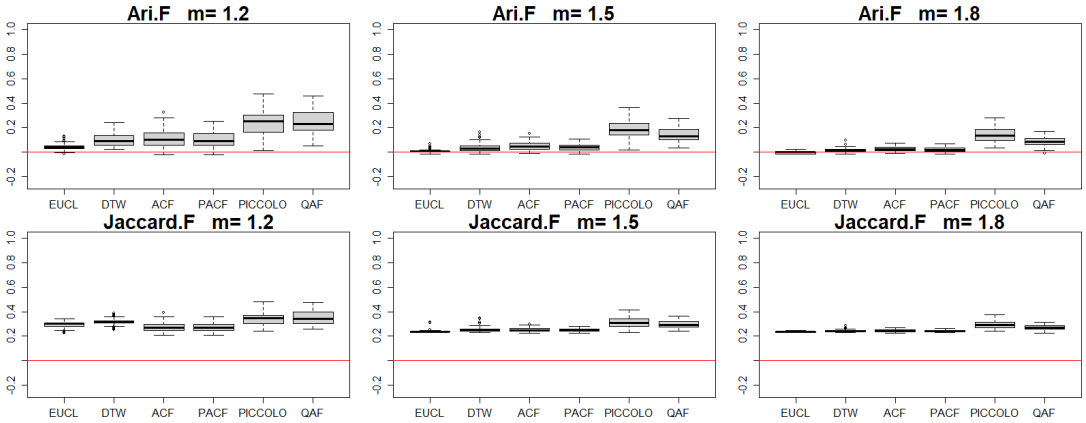


(a) T = 50

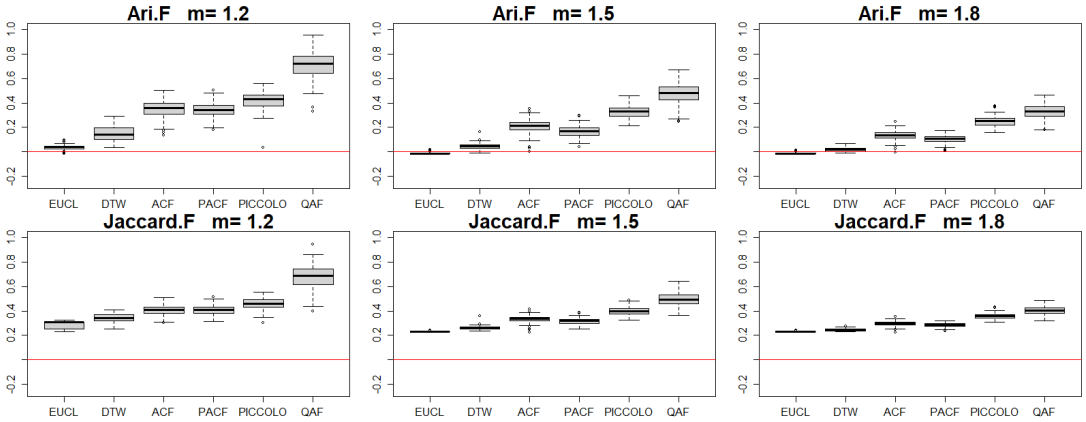


(b) T = 200

Figura A.2: Diagramas de caja de los índices de calidad cluster ARI.F y Jaccard.F basados en las 100 réplicas del algoritmo FCMdC en el Escenario 1.B.

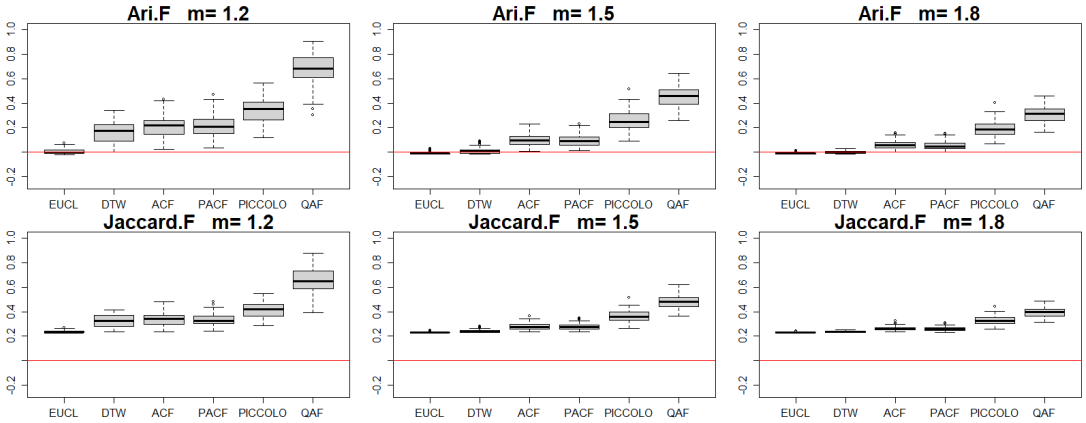


(a) T = 50

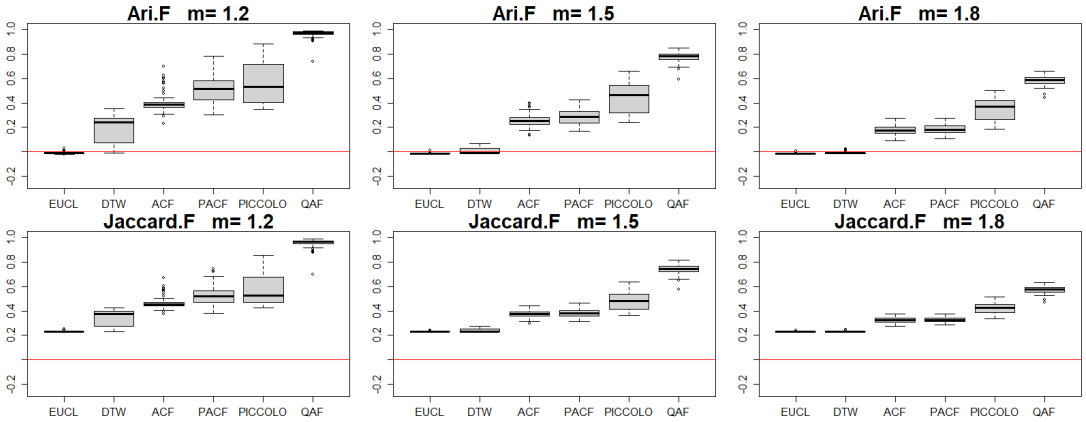


(b) T = 200

Figura A.3: Diagramas de caja de los índices de calidad cluster ARI.F y Jaccard.F basados en las 100 réplicas del algoritmo FCMdC en el Escenario 2.A.



(a) T = 50



(b) T = 200

Figura A.4: Diagramas de caja de los índices de calidad cluster ARI.F y Jaccard.F basados en las 100 réplicas del algoritmo FCMdC en el Escenario 3.A.

Bibliografía

[1] P. PBC, "Rstudio: Open source & professional software for data science teams," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://posit.co/downloads/

[2] Microsoft, "Visual studio code," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://code.visualstudio.com/

[3] DBeaver Corporation, "Dbeaver," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://dbeaver.io/

[4] GeoServer Project, "Geoserver," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://geoserver.org/

[5] Leaflet, "Leaflet: a javascript library for interactive maps," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://leafletjs.com/

[6] PostGIS Project, "Postgis," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://postgis.net/

[7] PostgreSQL Global Development Group, "Postgresql," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://www.postgresql.org/

[8] Glassdoor, "Junior data scientist sueldos," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://www.glassdoor.es/Sueldos/junior-data-scientist-sueldo-SRCH_KO0,21.htm

[9] ——, "Catedrático sueldos," accedido: 11 de noviembre de 2024. [En línea]. Disponible en: https://www.glassdoor.es/Sueldos/catedr%C3%A1tico-sueldo-SRCH_KO0,11.htm

[10] T. Liao, "Clustering of time series data: a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[11] T.-c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[12] S. Rani and G. Sikka, "Recent techniques of clustering of time series data: a survey," *International Journal of Computer Applications*, vol. 52, no. 15, pp. 1–9, 2012.

[13] J. Caiado, E. Maharaj, and P. D'Urso, "Time series clustering," in *Handbook of Cluster Analysis*, ser. Handbooks of Modern Statistical Methods, C. Hennig, M. Meila, F. Murtagh, and R. Rocci, Eds. London: Chapman and Hall/CRC, 2015, pp. 241–263.

[14] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition.* John Wiley & Sons, 1999.

[15] P. D'Urso and E. A. Maharaj, "Autocorrelation-based fuzzy clustering of time series," *Fuzzy Sets and Systems*, vol. 160, no. 24, pp. 3565–3589, 2009.

[16] P. D'Urso, L. De Giovanni, R. Massari, and D. Di Lallo, "Noise fuzzy clustering of time series by autoregressive metric," *METRON*, vol. 71, no. 3, pp. 217–243, 2013.

[17] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition.* Chichester, UK: Wiley, 1999.

[18] E. Schubert and P. J. Rousseeuw, "Faster k-medoids clustering: Improving the pam, clara, and clarans algorithms," in *SISAP 2020*, 2019, pp. 171–187.

[19] E. Anderson, "The irises of the gaspe peninsula," *Bulletin of the American Iris Society*, vol. 59, pp. 2–5, 1935.

[20] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[21] R. A. Becker, J. M. Chambers, and A. R. Wilks, *The New S Language.* Wadsworth & Brooks/Cole, 1988.

[22] P. Giordani, M. B. Ferraro, and A. Serafini, *fclust: Fuzzy Clustering*, 2024, versión 1.0-1, disponible en CRAN. [En línea]. Disponible en: https://cran.r-project.org/web/packages/fclust/index.html

[23] R. Krishnapuram, A. Joshi, O. Nasraoui, and L. Yi, "Low-complexity fuzzy relational clustering algorithms for web mining," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 595–607, 2001.

[24] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering: a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[25] P. Montero and J. A. Vilar, "Tsclust: An r package for time series clustering," *Journal of Statistical Software*, vol. 62, no. 1, Nov. 2014. [En línea]. Disponible en: http://www.jstatsoft.org/

[26] E. A. Maharaj, P. D'Urso, and J. Caiado, *Time Series Clustering and Classification.* London, UK: Chapman and Hall/CRC, 2019.

[27] M. Corduas, "Mining time series data: A selective survey," in *Data Analysis and Classification*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, F. Palumbo, C. N. Lauro, and M. J. Greenacre, Eds. Berlin: Springer-Verlag, 2010, pp. 355–362.

[28] J. Lin and Y. Li, "Finding structural similarity in time series data using bag-of-patterns representation," in *Proceedings of the 21st International Conference on Scientific and Statistical Database Management, SSDBM 2009.* Berlin: Springer-Verlag, 2009, pp. 461–477.

[29] P. D'Urso, L. De Giovanni, and R. Massari, "Garch-based robust clustering of time series," *Fuzzy Sets and Systems*, vol. 305, pp. 1–28, 2016.

[30] G. E. A. P. A. Batista, X. Wang, and E. J. Keogh, "A complexity-invariant distance measure for time series," in *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM11.* Mesa: SIAM, 2011, pp. 699–710.

[31] D. Sanko and J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison Wesley, 1983.

[32] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD Workshop*, 1994, pp. 359–370.

[33] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data Min. Knowl. Discov.*, vol. 13, no. 3, p. 335–364, 2006.

[34] Z. Bohte, D. Cepar, and K. Kosmelj, "Clustering of time series," in *COMPTSTAT 80, Proceedings in Computational Statistics*, M. Barritt and D. Wishart, Eds. Heidelberg: Physica Verlag, 1980, pp. 587–593.

[35] P. Galeano and D. Pena, "Multivariate analysis in vector time series," *Resenhas do Instituto de Matematica e Estatistica da Universidade de Sao Paulo*, vol. 4, no. 4, pp. 383–403, 2000.

[36] J. A. Vilar, B. Lafuente-Rego, and P. D'Urso, "Quantile autocovariances: A powerful tool for hard and soft partitional clustering of time series," *Fuzzy Sets and Systems*, vol. 340, pp. 38–72, June 2018. [En línea]. Disponible en: https://doi.org/10.1016/j.fss.2018.01.011

[37] D. Piccolo, "A distance measure for classifying arima models," *Journal of Time Series Analysis*, vol. 11, no. 2, pp. 153–164, 1990.

[38] M.-S. Yang, K.-L. Wu, J.-N. Hsieh, and J. Yu, "Alpha-cut implemented fuzzy clustering algorithms and switching regressions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 3, pp. 588–603, 2008.

[39] E. A. Maharaj and P. D'Urso, "Fuzzy clustering of time series in the frequency domain," *Information Sciences*, vol. 181, no. 7, pp. 1187–1211, 2011.

[40] F. d. A. de Carvalho, C. Tenrio, and N. Junior, "Partitional fuzzy clustering methods based on adaptive quadratic distances," *Fuzzy Sets and Systems*, vol. 157, no. 21, pp. 2833–2857, 2006.

[41] T. Kamdar and A. Joshi, "On creating adaptive web servers using weblog mining," Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, USA, Technical Report TR-CS-00-05, 2000.

[42] R. J. G. B. Campello, "A fuzzy extension of the rand index and other related indexes for clustering and classification assessment," *Pattern Recognition Letters*, vol. 28, pp. 833–841, 2007.

[43] B. Lafuente-Rego, P. D'Urso, and J. A. Vilar, "Robust fuzzy clustering based on quantile autocovariances," *Statistical Papers*, vol. 61, no. 6, pp. 2393–2448, December 2020. [En línea]. Disponible en: https://doi.org/10.1007/s00362-018-1053-6

[44] G. Aneiros, J. Vilar, and P. Raña, "Short-term forecast of daily curves of electricity demand and price," *International Journal of Electrical Power Energy Systems*, vol. 80, pp. 96–108, 2016. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0142061516000466

[45] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/0377042787901257

[46] R. Campello and E. Hruschka, "A fuzzy extension of the silhouette width criterion for cluster analysis," *Fuzzy Sets and Systems*, vol. 157, no. 21, pp. 2858–2875, 2006. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0165011406002892

[47] M. Febrero-Bande and M. Oviedo de la Fuente, "Statistical computing in functional data analysis: The R package fda.usc," *Journal of Statistical Software*, vol. 51, no. 4, pp. 1–28, 2012. [En línea]. Disponible en: https://www.jstatsoft.org/v51/i04/