Estudio de Métricas de Similitud Entre Series Temporales y Técnicas de Análisis de Conglomerados de Series Temporales

José Manuel Juan Barrientos

Supervisado por David Gálvez Ruiz

Facultad de Matemáticas

Noviembre 2022

A Jose, por estar siempre a mi lado,
a Alejandra, por ser mi apoyo,
a mi hermana, por quererme tanto,
y a mis padres, por acompañarme en este camino.

Resumen

En este trabajo se pretende proporcionar una visión sobre el Análisis de Conglomerados de series temporales a través de un estudio centrado en la determinación de la similitud entre estas.

Para ello, se comienza definiendo los conceptos que se usarán a lo largo del trabajo y se muestra la relevancia de hallar la similitud. Se continua con una extensa revisión a distintas formas de hallarla junto a una selección de medidas de bondad de ajuste. Se concluye con una ejemplificación de la teoría expuesta aplicando dos técnicas distintas a un caso real y comparando sus resultados.

Abstract

This work aims to provide an overview of time series Cluster Analysis through a study focused on determining the similarity between them.

For this, it begins by defining the concepts which will be used throughout the work and showing the relevance of finding the similarity. It continues with an extensive review of different ways to find it together with a selection of measures of goodness of fit. It concludes with an exemplification of the theory by applying two different techniques to a real case and comparing their results.

Índice general

1

Introducción

En diversos entornos como en el financiero, meteorológico, médico o psicológico se trata con grandes cantidades de datos que se necesitan interpretar. Estos datos suelen tener una forma particular, la de series temporales.

Una serie temporal se puede ver como una colección de datos tomados en intervalos regulares (o irregulares) de tiempo. Las emisiones de CO_2 anuales de España en los últimos 20 años, los resultados de un sujeto dentro de estudio psicológico donde se evalúan a los sujetos en distintos momentos o los usos que se ven estos artículos [1], [2], [3] son algunos ejemplos de series temporales. Existen muchos otros ejemplos, teniéndose por tanto una gran variedad dentro de las series temporales pues se pueden medir intervalos de tiempo regulares e irregulares, ser o no estacionarias, medir una o varias variables que pueden ser continuas, categóricas o una combinación de ambas y un amplio etcétera. Esta gran variedad si bien positiva en tanto que sus aplicaciones hace necesaria una delimitación sobre el tipo de series temporales que se tratarán en el estudio. Así se tratarán sólo las series temporales medidas en variables continuas con alguna mención a otros tipos.

La característica principal de las series temporales es que con ellas portan información sobre las características temporales de los datos. Explotar, analizar y extraer información sobre este aspecto juega un papel muy importante de la relevancia de esta forma de considerar los datos como se verá a lo largo del trabajo.

Este documento trata el Análisis de Conglomerados que clasifican las series temporales agrupándolas de manera no supervisada [4]. Para ello se parte de datos sin clasificar y se organizan en diferentes agrupaciones de forma que se maximice la similitud entre los elementos de un mismo grupo y se minimice la similitud de estos con los de otros posibles grupos. En concreto el trabajo se centra en las técnicas de conglomerados y en proporcionar formas de medir la similitud entre series temporales.

Este documento se organiza del siguiente modo: una primera parte donde se introducirán las definiciones y conceptos necesarios para el correcto desarrollo de los estudios. Se verá continuado por el estudio de métricas de similitud donde se presentarán las más destacadas y se abordará la *maldición de la dimensionalidad* [5]. Tras esto, se dará una clasificación de las técnicas de Análisis de Conglomerados según el tratamiento de las series temporales considerándose las basadas en la forma, en las características o en modelos estadísticos derivados de las series temporales. Se concluye esta primera parte exponiendo distintas medidas de evaluación que proporcionan información sobre la bondad del análisis realizado. En la segunda parte se ejemplificará la teoría vista mediante la aplicación de dos técnicas, DTW y MMO, a un caso real realizando un Análisis de Conglomerados con cada una de ellas y comparando los resultados entre sí para determinar la mejor elección.

2

Primeras nociones

Es importante destacar que la naturaleza de las series temporales hace necesarias algunas consideraciones que se mostrarán en esta sección.

2.1 Series temporales

En este documento se considerarán definidas como sigue.

Definición 2.1.1 (Serie temporal univariante)**.** Una serie temporal puede entenderse como un vector

$$x = (x_1, x_2, \ldots, x_T)$$

Donde x_t es un dato tomado en el momento t-ésimo y T es la cantidad total de observaciones.

De manera más general se tratan con las siguientes dos definiciones. Se tiene que:

Definición 2.1.2 (Serie temporal multivariante)**.** Dadas m variables medidas en T momentos distintos y en orden creciente de tiempo. Una serie temporal multivariante (STM) se define como:

$$X = (X_1, X_2, \ldots, X_m)$$

Donde X_j es una serie temporal univariante de longitud T y $X_j(t)$ es una observación de la variable j-ésima en el momento $t \quad \forall i \in \{1, 2, \ldots, m\}$.

Definición 2.1.3 (Serie temporal matricial)**.** Dadas n STMs de m variables medidas en T momentos distintos. Se define una serie temporal matricial como:

$$\mathbb{X} = \begin{pmatrix} X_{11} & \cdots & X_{1j} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ X_{i1} & \cdots & X_{ij} & \cdots & X_{im} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ X_{n1} & \cdots & X_{nj} & \cdots & X_{nm} \end{pmatrix}$$

donde el elemento X_{ij} es una serie temporal univariante de longitud T, la fila i-ésima una STM de m variables y la columna j-ésima da los valores de la j-ésima variable en las n STMs.

Se denotará $\mathbb{X}(t)$ a:

$$\mathbb{X}(t) = (X_{ij}(t))_{n\times m},$$

la serie temporal matricial en el momento t.

Observación. Se tiene que una serie temporal matricial se reduce a una multivariante cuando $n = 1$ y a una univariante si $n = m = 1$.

Dados los objetos anteriores se plantea el problema de agrupar las STMs para intentar hallar resultados significativos. Ver todas las posibles agrupaciones y elegir la mejor es muy costoso por la gran cantidad de posibles combinaciones, número de conglomerados y lo elevado que puede llegar a ser el número de objetos a clasificar. Es por esto que resulta tan importante el Análisis de Conglomerados.

2.2 Introducción al Análisis de Conglomerados

El análisis de conglomerados tiene como objetivo resolver el siguiente problema de clasificación:

Dada Ξ , una muestra de n objetos medidos en m variables, i.e., $X_i = (X_{i1}, X_{i2}, \ldots, X_{im})$ $\forall i \in \{1, 2, \ldots, n\}$. Se quieren clasificar en G grupos o conglomerados cumpliendo que:

$$\bigcup_{i=1}^{G} C_i = \Xi,$$
$$C_i \cap C_j = \emptyset \quad \forall i, j \in \{1, 2, \ldots, G\},\ i \neq j.$$

A la par que se desea esa división maximice la similitud entre los objetos de un mismo conglomerado, C_i, y minimice la similitud de esos para con los de los C_j restantes.

A los métodos usados para abordar esta problemática se les han otorgado diversos nombres como, Q-análisis, Taxonomía Numérica y Análisis de Clústers o Conglomerados según su área de aplicación (véanse Psicología, Biología y Estadística respectivamente). Si bien posee numerosas aplicaciones el uso del Análisis de Conglomerados (como se le referirá en este documento) se limita a 3 casos:

- **Partición de los datos:** dado un conjunto de datos sin clasificar, el cual se sospecha heterogéneo, se quiere dividir en grupos cumpliendo que cada individuo pertenezca a uno y solo uno y que las agrupaciones sean internamente homogéneas. El número de grupos puede venir prefijado antes de realizar el análisis (según el método a elegir).
- **Clasificación de variables:** en el mismo caso anterior, se pretende esta vez agrupar variables para reducir la dimensión del problema.
- **Construir jerarquías:** se busca estructurar los casos de forma jerárquica por similitud se puede ver como consecuencia de usar algoritmos jerárquicos para resolver los casos anteriores .

Por tanto, se pueden distinguir dos tipos de conglomerados: de datos y de variables. Así, dado el problema se puede disponer la muestra Ξ en una matriz (matriz de datos) como sigue:

$$\mathbb{X} = \begin{pmatrix} X_{11} & \cdots & X_{1j} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ X_{i1} & \cdots & X_{ij} & \cdots & X_{im} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ X_{n1} & \cdots & X_{nj} & \cdots & X_{nm} \end{pmatrix}$$

donde la i-ésima fila proporciona los valores del i-ésimo objeto en las m variables y la j-ésima columna los de la j-ésima variable a lo largo de todos los objetos de la muestra. De esta forma, $\mathbb{X}$ permitiría estudiar el primer caso y para el segundo solo haría falta tomar la traspuesta.

Observación. Si tomamos que Ξ sea una muestra de n series temporales m-variantes medidas en T intervalos de tiempo, $\mathbb{X}$ sería una serie temporal matricial. Es más, se puede ver como

$$\mathbb{X} = (X_i)_{i=1,\dots,n}$$

con X_k la k-ésima STM de la muestra.

El objetivo del Análisis de Conglomerados es dar una división de Ξ que maximice la similitud dentro de cada conglomerado a la par que minimice la similitud con el resto de conglomerados. Para ello, en vez de usar los datos dispuestos como en $\mathbb{X}$ se puede considerar una matriz $n \times n$ de distancia o similitud entre elementos de la muestra. Sea B dicha matriz se tendría que b_{ij} con $i, j = 1, 2, \dots, n$ es la distancia o similitud entre el elemento X_i y X_j de Ξ.

2.2.1 Distancia y (di)similitud

Se ha hablado de similitud entre los objetos de la muestra en tanto que es necesaria para cuantificar la cercanía (o parecido) entre estos y así poder agruparlos. A continuación, se expondrá que se entiende por distancia y por similitud en este contexto.

Definición 2.2.1 (Distancia)**.** Sea Y un conjunto no vacío. Se dice que la aplicación $d : Y \times Y \to [0, \infty)$ es una distancia o métrica si cumple que $\forall x, y, z \in Y$:

- *Reflexividad:*
$$d(x, y) = 0 \Leftrightarrow x = y.$$
- *Simetría:*
$$d(x, y) = d(y, x).$$
- *Desigualdad triangular:*
$$d(x, y) \leq d(x, z) \ + \ d(z, y).$$

La literatura falla a la hora de dar una definición precisa de similitud. Esta se usa como término paraguas: toda función que permita afirmar que dos objetos son similares, i.e. que las propiedades que se hayan medido se parezcan según el criterio establecido por esta, se considera similitud. Así, se entiende que a menor similitud menor parecido. Teniendo esto presente, se usará [6] para dar la definición en la que nos basaremos.

Definición 2.2.2 (Similitud)**.** Sea Y un conjunto no vacío. La aplicación $s : Y \times Y \to \mathbb{R}$ es una similitud o medida de similitud si es una función acotada superiormente cumpliendo que para $s_{max} = \text{máx Im}(s)$ y $\forall x, y \in Y$:

- *Reflexividad:*
$$s(x, y) = s_{max} \Leftrightarrow x = y.$$
- *Simetría:*
$$s(x, y) = s(y, x).$$

Entendiéndose que a menor disimilitud más parecido existe entre los objetos y de manera análoga a la similitud, obtenemos:

Definición 2.2.3 (Disimilitud)**.** Sea Y un conjunto no vacío. La aplicación $dis : Y \times Y \to \mathbb{R}$ es una disimilitud o medida de disimilitud si es una función acotada inferiormente cumpliendo que para $dis_{min} =$ mín $\mathrm{Im}(dis)$ y $\forall x, y \in Y$:

- *Reflexividad:*
$$dis(x, y) = dis_{min} \Leftrightarrow x = y.$$
- *Simetría:*
$$dis(x, y) = dis(y, x).$$

No toda medida de (di)similitud ha de encajar en las definiciones anteriores. Estas se han tomado por ser intuitivas y generales. Se pueden dar otras como las vistas en [6]. Será suficiente, en este caso, con las definiciones proporcionadas.

Observación. Se tiene que toda distancia es una medida de disimilitud.

Una vez establecido el concepto de (di)similitud se puede obtener B, la matriz de (di)similitud:

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1i} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ b_{i1} & & b_{ii} & & b_{in} \\ \vdots & & \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{ni} & \cdots & b_{nn} \end{pmatrix}$$

Matriz $n \times n$ donde b_{ij} es la medida bien de similitud o disimilitud entre el j-ésimo y el i-ésimo elemento de la muestra. Usando las definiciones proporcionada se tiene que B cumple:

1. $b_{ii} = s_{max}$ (dis_{min} si es de disimilitud) $\forall i = 1, \ldots, n.$
2. Es simétrica:
$$b_{ij} = b_{ji} \ \forall i, j = 1, \ldots, n.$$

Si además cumple que es semi-definida positiva como matriz de una similitud s cumpliendo:

$$0 \leq s(x, y) \leq 1.$$

Se demuestra en [7] que
$$d(x, y) = \sqrt{1 - s(x, y)}$$
define una distancia.

Observación. Es importante notar que se trabaja con STMs. Si bien una serie temporal univariante puede tratarse como un vector de T componentes, una STM sería una matriz $m \times T$ y han de definirse las medidas teniendo esto en cuenta. Esto se verá con detenimiento en 3.

Ya conocido lo que queremos maximizar (minimizar en el caso de ser disimilitud) se quiere agrupar los objetos de la muestra siguiendo el criterio expuesto al inicio de la sección 2.2. Para ello hay diferentes técnicas y algoritmos.

2.2.2 Proceso de Análisis de Conglomerados y algoritmos

Realizar este análisis es un proceso que consta de varias etapas. De manera general se tienen:

1. Seleccionar las variables significativas que describan el objeto de estudio.
2. Seleccionar la medida de (di)similitud a usar para crear las agrupaciones.
3. Seleccionar el algoritmo a usar.
4. Valoración de los resultados con análisis de robustez y consistencia.

Todas las etapas se contemplan con detenimiento a lo largo de este documento. La primera etapa se verá en 6. La segunda etapa se estudiará en profundidad en 3 y también en 4. Los algoritmos se expondrán a continuación. Finalmente la valoración de los resultados se trata en 5.

Observación. Es importante notar que a la combinación de las etapas 2 y 3, elegir una medida y con esta aplicar cualquiera (o uno solo) de los algoritmos que se verán, es lo que denotaremos como *técnica* de Análisis de Conglomerados. Esto es lo que se expondrá en 4 donde se clasificarán según hagan uso de la (di)similitud.

Por algoritmo se entiende la manipulación de los objetos per se para agruparlos. Así, se tienen que hay dos tipos:

- **Jerárquicos.** Son los algoritmos que en cada paso o bien dividen un conglomerado en dos o bien aúnan dos en uno en tanto que se optimice la medida usada. Hay dos tipos:
  - *Aglomerativos.* Se considera cada elemento un conglomerado. Partiendo de esto, en cada iteración se unen los dos conglomerados más similares hasta que queda uno solo que contiene a toda la muestra.
  - *Divisivos.* Es el proceso inverso. Partiendo de un único conglomerado con toda la muestra, en cada iteración se subdivide hasta que cada objeto es un conglomerado.
- **No jerárquicos.** Estos algoritmos hacen una única división. Primero se fija el número, k, de conglomerados ha formar y se toma una partición inicial de la muestra en k grupos. En cada iteración se reasignan los elementos hasta que el algoritmo se estabiliza. La división resultante es la obtenida en la última iteración del algoritmo.

Observación. En el caso de las series temporales existe también algoritmos que usan segmentos o puntos concretos de las series temporales en vez de su totalidad [4]. En este documento solo se verán aquellos que usan las series temporales al completo.

3

Estudio de métricas de (di)similitud en series temporales

En la sección anterior se ha visto que la medida de (di)similitud es vertebral en el Análisis de Conglomerados al necesitar una forma de determinar el parecido (o la diferencia) entre elementos y en base a esta formar los grupos. La motivación de este estudio parte de lo anterior pues aunque se aplique el mismo algoritmo cambiar la medida usada podría llevar a cambiar la asignación de etiquetas y por ende los resultados. Los conglomerados resultantes podrían ser distintos tanto en número como en los elementos que los conforman. Por esto la elección de la medida de similitud adecuada es esencial.

La respuesta a las preguntas «¿qué medida usar?,¿cuál es la mejor?» es compleja y no encuentra una respuesta única ni completamente certera. Esta decisión depende de factores como el tipo de datos o el objetivo con el que se hace el análisis. Se verán en esta sección las principales medidas, sus usos, ventajas y desventajas.

3.1 Series temporales univariantes

Las medidas difieren entre las series temporales univariantes y multivariante por lo que dividir este estudio en los dos casos es obligado.

Es importante notar que se suponen las variables medidas son continuas y en consecuencia serán las medidas mostradas. Si bien, esto es una generalización que deja atrás muchos resultados interesantes, se hace necesaria dada la extensión de la literatura en este ámbito. Aún así, se harán menciones a alguno de los más relevantes para este documento.

3.1.1 Distancias métricas

Como se ha mencionado anteriormente toda distancia métrica es una medida de disimilitud. Estas medidas se pueden aplicar a la series temporales (de una variable continua) si se consideran como vectores de $\mathbb{R}^T$ con T la longitud de la serie.

Una vez elegida la medida para determinar los grupos se necesita saber que series temporal están más cerca (tiene menor disimilitud) entre sí. De aquí nace el problema de «El vecino más cercano» consistiendo

en dado un punto fijo (una serie temporal) encontrar otra lo más cercana posible con respecto a las demás.

Es importante notar que las series temporales pueden alcanzar valores de T muy elevados como los obtenidos en aplicaciones climatológicas por lo que se hace patente la *maldición de la dimensionalidad* [5]. Este es un nombre usualmente atribuido a los problemas que se puedan originar al trabajar en altas dimensiones. En el caso que nos concierne, el problema del vecino más cercano, se tiene que al usar las normas L_p $\forall p \in [1, \infty]$ a mayor p mayor pérdida de significación de la solución al problema [8].

Con lo anterior presente, consideremos una matriz $\mathbb{X}$ de n series temporales medidas en T momentos distintos. Sean $x = (x_1, x_2, \ldots, x_T)$ e $y = (y_1, y_2, \ldots, y_T)$ dos elementos de $\mathbb{X}$ y sean x_t, y_t los valores en el momento t-ésimo de las respectivas series. Las distancias más usadas son las que siguen:

- Distancia Minkowski: $d_p(x, y) = \left(\sum_{t=1}^{T}(x_t - y_t)^p\right)^{1/p} \ \forall p \in [1, \infty)$.
- Distancia de Chebyshev: $d_\infty(x, y) = \text{máx}\{|x_t - y_t|,\ t = 1, \ldots, T\}$.
- Distancia Mahalanobis: $d_M(x, y) = \left((x - y)\Sigma^{-1}(x - y)^\top\right)^{1/2}$.
- Distancia cuerda: $d_{cu}(x, y) = \left(2 - 2\frac{\sum_{t=1}^{T} x_t y_t}{\|x\|_2 \|y\|_2}\right)^{1/2}$.
- Distancia coseno: $d_{cos} = \frac{\sum_{t=1}^{T} x_t y_t}{\|x\|_2 \|y\|_2}$.

Donde Σ es la matriz de covarianza de los datos y $\|\cdot\|_2$ la norma euclídea.

Partiendo del hecho que en las medidas anteriores es necesario que todas las series tengan la misma longitud. Las distancia Minkowski y Chebyshev provienen de las normas L_p y por tanto heredan el problema de la pérdida de significado en altas dimensiones anterior. Si bien las mejores opciones serían d_1 y d_2, la distancia Manhattan y euclídea respectivamente. Presentarían menos problemas al aumentar de dimensión aunque siguen siendo muy ineficientes para los altos valores que puede tomar T, llegando a ser superadas en eficiencia por un escáner secuencial [9]. Además ocurre que la variable con mayores valores domina al resto y las correlaciones lineales entre estas pueden acortar distancias dando lugar a medidas erróneas [10]. Estos dos últimos problemas se pueden afrontar haciendo uso de la distancia cuerda al normalizar los vectores (define la distancia como la cuerda que les une en una hiperesfera unidad) y con la distancia Mahalanobis que usa la matriz de covarianzas asociando diferentes pesos a las variables. Sin embargo esto no soluciona el problema con la dimensión de T y la distancia Mahalanobis requiere de la normalidad de los datos y de clústeres hiperelipsoidales [11].

Por otro lado la distancia coseno no presenta estos problemas pues es independiente de la longitud del vector e invariante ante rotaciones. Sin embargo, no es invariante frente a transformaciones lineales.

En las desventajas asociadas a las medidas anteriores no se ha contemplado la más importante. Al pretender que una serie temporal es solamente un vector se elimina una de sus características más importantes: son datos temporales. Esto nos permite volver a la pregunta de «¿qué medida usar?» y ver que las anteriores no serían la respuesta más acertada ya que no tienen en cuenta el tipo de datos con los que se tratan.

3.1.2 Similitudes y disimilitudes

Eliminar la obligación de cumplir la desigualdad triangular da lugar a poder definir nuevas medidas que sí tienen en cuenta la temporalidad de los datos siendo así más robustas ante series de distintos tamaño, transformaciones como el desplazamiento en el tiempo de una serie respecto a otra, etc.

El primer ejemplo que se verá con más detalle en la sección 4 es Dynamic Time Warping [12]:

Dynamic Time Warping

Dadas dos series temporales $x = (x_1, x_2, \ldots, x_T)$ e $y = x = (y_1, y_2, \ldots, y_S)$, de longitudes T y S respectivamente pudiendo ser $T \neq S$ se crea una matriz de distancias $T \times S$, $D_{x,y}$:

$$D_{x,y} = \begin{pmatrix} d_p(x_T, y_1) & d_p(x_T, y_2) & \cdots & d_p(x_T, y_S) \\ \vdots & \vdots & \ddots & \vdots \\ d_p(x_2, y_1) & d_p(x_2, y_2) & \cdots & d_p(x_2, y_S) \\ d_p(x_1, y_1) & d_p(x_1, y_2) & \cdots & d_p(x_1, y_S) \end{pmatrix}$$

donde en d_p se toma comúnmente la distancia euclídea ($p = 2$). Una vez obtenida esta matriz se busca un camino, $W = \{w_1, \ldots, w_K\}$, de elementos contiguos de la matriz $D_{x,y}$ cumpliendo, generalmente:

1. $w_1 = (1, 1)$ y $w_K = (T, S)$.
2. Si $w_k = (i, j)$ y $w_{k-1} = (i', j')$ entonces:

$$0 \leq i - i' \leq 1$$
$$0 \leq j - j' \leq 1.$$

Es decir se busca un camino continuo y monótono que coincida en los extremos para asegurar una alineación adecuada.

Finalmente la distancia entre la series x e y viene dada por el camino óptimo que cumple lo anterior y minimiza la función:

$$DTW(x, y) = \text{mín} \left(\sum_{i=1}^{K} w_i \right). \tag{3.1}$$

Si bien esta medida de disimilitud hace uso de la distancia euclídea el resultado obtenido es muy diferente gracias al camino W. Siendo en tanto muy útil para encontrar series temporales con formas parecidas incluso si ha habido un desplazamiento de estas en el tiempo o un cambio de escala. Sin embargo tiene un elevado coste computacional, $O(TS)$ [12].

DTW al contrario que la distancias que le han precedido, entra dentro de un tipo de medidas llamadas elásticas al permitir flexibilidad [13] como se observa en las siguiente gráficas.

(a) Dynamic Time Warping (b) Distancia euclídea

Figura 3.1: Diferencia entre DTW y la distancia euclídea donde las líneas grises representan el mapeo entre los puntos de las dos series temporales observándose como DTW aproxima mejor la forma.

Si bien soluciona en cierto modo el problema de las transformaciones de las series temporales (en concreto el desplazamiento y/o el cambio de escala), sigue siendo costoso para altos valores de T y S. Para afrontar este problema las medidas que se verán se encargan de extraer una representación de las series temporales (disminuyendo así la dimensión) y sobre estas medir. Este procedimiento es parecido al que se explicará en 4 pues no se miden las series per sé sino una representación (o modelo) de estas.

El primer ejemplo de este tipo de medidas lleva usándose desde 1993 [14] y se basa en concebir las series temporales como ondas. De este modo, todos los ámbitos que traten con datos como señales de radio, sonido o incluso datos climatológicos se pueden ver beneficiados por esta consideración. A este método se le conoce como la Transformada Discreta de Fourier, pues hace uso de esta para aproximar las series temporales.

Transformada Discreta de Fourier

Dada una serie temporal, $x = (x_1, x_2, \ldots, x_n)$ su transformada de Fouirer es [15]:

$$X_f = \frac{1}{\sqrt{n}\sum_{t=1}^{n} e^{-i2\pi ft/n}} \; f = 1, 2, \ldots, n.$$

Donde $i = \sqrt{-1}$, la unidad imaginaria. Una vez transformadas las series se tiene el teorema de Parseval:

Teorema (Paserval)**.** Sea X la Transformada Discreta de Fourier de la serie x. Entonces:

$$\sum_{t=1}^{n} |x_t|^2 = \sum_{f=1}^{n} |X_f|^2.$$

De este teorema se deduce que, definiendo $x = z - y$, la diferencia de dos series temporales, la distancia euclídea entre ellas coincide con la diferencia de los valores de sus transformadas. Así se define la distancia entre dos series temporales, como la suma de cuadrados, que aplicando lo anterior resulta en:

$$TDF(z, y) = \sqrt{\sum_{f=1}^{n} |X_f|^2}.$$

Esta técnica reduce la dimensión pues es con $f < 5$ ya se considera una buena aproximación. Además es invariante ante desplazamientos en el tiempo de las series y el cálculo de la transformada tiene complejidad

$O(nlog(n))$ donde n es la longitud de las series [15]. Por lo tanto se resuelve el problema que nos planteaba el DTW sobre la complejidad.

Existen otros métodos de reducción de dimensión de las series temporales como el SVD [16] que ya no se pueden considerar medidas pero que también resuelven el problema de la maldición de la dimensionalidad [13].

Si bien puede parecer que tener en cuenta la naturaleza de las series temporales desemboca en medidas complejas, no siempre es así. Hay medidas como las que siguen que desafían esto:

Correlación temporal

La correlación temporal es un coeficiente que se inspira en el coeficiente de correlación de Pearson (que también se puede usar como medida de series temporales) pero no presenta el problema de sobre estimación que este sí al tener en cuenta la dependencia temporal entre las variables [17]. Dadas dos series $x = (x_1, x_2, \ldots, x_T)$ e $y = x = (y_1, y_2, \ldots, y_T)$ se define su correlación temporal se como sigue [17]:

$$CORT(x, y) = \frac{\sum_{t=1}^{T-1}(x_{t+1} - x_t)(y_{t+1} - y_t)}{\sqrt{\sum_{t=1}^{T-1}(x_{t+1} - x_t)^2}\sqrt{\sum_{t=1}^{T-1}(y_{t+1} - y_t)^2}}.$$

Así, $CORT(x, y) \in [-1, 1]$ donde si el coeficiente es 1 entonces las series muestran un comportamiento dinámico similar, en el caso de ser -1 este es similar pero en la dirección opuesta y si es 0, muestran distintos comportamientos.

Una vez que se establece este coeficiente la medida de disimilitud necesita de otra medida, normalmente la euclídea o el DTW, para asociarle un peso a los valores que estas proporcionan a través de una función de ajuste adaptativa. Así, la distancia sería:

$$d_{CORT}(x, y) = \phi(CORT(x, y))d(x, y).$$

La función propuesta en [17] es:

$$\phi_k(u) = \frac{2}{1 + e^{k*u}} \ \forall k \geq 0.$$

Cumpliéndose que a medida que $CORT(x, y)$ vaya incrementando desde 0 hasta 1 la disimilitud vaya disminuyendo y viceversa.

Coeficiente de correlación Pearson

Del mismo modo a la distancia anterior se puede definir una disimilaridad usando el coeficiente de correlación de Pearson. Sea una serie temporal $x = (x_1, x_2, \ldots, x_T)$ su media se define como:

$$\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t.$$

Teniendo esto en cuenta y tomando otra serie temporal y de igual longitud. El coeficiente de correlación

de Pearson es

$$COR(x,y) = \frac{\sum_{t=1}^{T}(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^{T}(x_t - \bar{x})^2}\sqrt{\sum_{t=1}^{T}(y_t - \bar{y})^2}}.$$

Se ve en [18] la definición de una disimilitud:

$$d_{COR}(x,y) = \sqrt{\left(\frac{1 - COR(x,y)}{1 + COR(x,y)}\right)^{\beta}}, \text{con } \beta > 0.$$

Importante notar que d_{COR} tiende a infinito cuando el coeficiente de correlación se acerca a -1 así que esta medida sirve para aplicaciones donde no se vea la anticorrelación como aceptable [18].

Finalmente, cambiando de manera de entender las series, si las vemos como funciones discretas de una variable se le puede aplicar la medida siguiente.

Distancia Fréchet

Introducida en primera instancia para medir la proximidad sobre curvas continuas en [19] se define para el caso discreto. Sean dos series temporales no necesariamente de la misma longitud $x = (x_1, x_2, \ldots, x_T)$ e $y = (y_1, y_2, \ldots, y_S)$. Se define M como el conjunto de todos los posibles emparejamientos de los valores de x e y dos a dos de longitud m, donde si $r \in M$, un emparejamiento, este será de la forma

$$r = ((x_{a_1}, y_{b_1}), \ldots, (x_{a_m}, y_{b_m}))$$

cumpliendo que $a_i, \in \{1, \ldots, T\}$, $b_i, \in \{1, \ldots, S\}$ tales que $a_1 = b_1 = 1$, $a_m = T$, $b_m = S$ y para todo $i = 1, \ldots, m-1$ es $a_{i+1} = a_i$ o $a_{i+1} = a_i + 1$ y $b_{i+1} = b_i$ o $b_{i+1} = b_i + 1$. Esto obliga a respetar el orden de los elementos. La distancia se define por tanto como

$$d_F(x,y) = \min_{r \in M}\left(\sum_{i=1}^{m}|x_{a_i} - y_{b_i}|\right).$$

En [19] se demuestra que esta disimilitud es de hecho una métrica. Además de una cota superior de la distancia de Fréchet original y una buena aproximación suya de complejidad $O(TS)$.

3.2 Series temporales multivariantes (STM)

Por la naturaleza de las medidas anteriores, salvo DTW, no se pueden aplicar a STM pues según se han definido son vectores de vectores a los que hacerles la media entre otras operaciones resultan en tareas no tan sencillas. El primer ejemplo que se verá será el DTW donde se hace muy patente este cambio de paradigma.

Dynamic Time Warping

Dadas dos STM $Y = (Y_1, Y_2, \ldots, Y_s)$ y $X = (X_1, X_2, \ldots, X_t)$, se tiene que cada componente es una serie temporal que suponemos longitud T para las de X y las de Y. Por tanto la matriz $D_{x,y}$ anterior cambia

siendo ahora:

$$D_{X,Y} = \begin{pmatrix} d_p(X_t, Y_1) & d_p(X_t, Y_2) & \cdots & d_p(X_t, X_s) \\ \vdots & \vdots & \ddots & \vdots \\ d_p(X_2, Y_1) & d_p(X_2, Y_2) & \cdots & d_p(X_2, Y_s) \\ d_p(X_1, Y_1) & d_p(X_1, Y_2) & \cdots & d_p(X_1, Y_s) \end{pmatrix}$$

donde la $d_p(X_i, Y_j)$ es la distancia entre dos vectores (las dos series temporales correspondientes). El camino W y la función a minimizar continúan la misma.

Distancia Edit en secuencias reales

Esta distancia nace de la necesidad de seguimiento de objetos ya sean en 2 o 3 dimensiones [20]. Así una serie temporal en este ámbito tendrá la forma $X = \big((x_1, t_1), \ldots, ((x_n, t_n)\big)$, donde x_i puede ser un vector de dos o tres dimensiones y t_i es el tiempo asociado que se considera discreto. Así la distancia entre dos trayectorias, Y y X de longitudes m y n respectivamente se define recursivamente como [20]:

$$EDR(X,Y) = \begin{cases} n & \text{si } m = 0 \\ m & \text{si } n = 0 \\ \text{mín}\{EDR(X_{-1}, Y_{-1}) + s, & \\ \quad EDR(X_{-1}, Y) + 1, & \text{en otro caso} \\ \quad EDR(X, Y_{-1}) + 1\} & \end{cases}$$

Donde X_{-1} es la trayectoria X quitándole el primer elemento y s es un coste que depende de los valores x_1, y_1 de manera que $s = 0$ si $|x_{1,1} - y_{1,1}| \leq \varepsilon$ y $|x_{1,2} - y_{1,2}| \leq \varepsilon$ para un ε a elegir y $s = 1$ en caso contrario.

Esta disimilitud mide la cantidad de veces que se necesita, insertar, borrar o desplazar la trayectoria X para transformarla. Esta medida presenta una mejora en el tratamiento de los efectos del ruido y por tanto en el de valores atípicos respecto a DTW y al igual que este no es sensible ante desplazamientos.

Esta medida sale de la distancia Edit junto con la distancia Edit con penalización real que se puede ver en [21]. Tanto la distancia que se ha descrito como la que se ha mencionado se pueden generalizar para tantas dimensiones como sean necesarias.

Subsecuencia común más larga

Siguiendo en el ámbito del seguimiento de objetos se encuentra esta medida que se presenta más adecuada que DTW para este trabajo [22]. Esta disimilitud trabaja con diferentes tasas de muestreo, longitudes y con valores atípicos, además de permitir que haya elementos sin emparejar. Dados dos trayectorias $Y = (Y_1, Y_2, \ldots, Y_m)$ y $X = (X_1, X_2, \ldots, X_n)$, $\delta \in \mathbb{Z}$ y $\varepsilon \in (0, 1)$. Se define la distancia como:

$$SSCL_{\delta,\varepsilon}(X,Y) = \begin{cases} 0 & \text{si X o Y son vacíos} \\ 1 + SSLC_{\delta,\varepsilon}(X_{-1}, Y_{-1}) & \text{si X e Y concuerdan y } |n - m| \leq \delta \\ \text{máx}\{SSLC_{\delta,\varepsilon}(X_{-1}, Y), SSLC_{\delta,\varepsilon}(X, Y_{-1})\} & \text{en caso contrario} \end{cases}$$

Donde X_{-1} vuelve a ser la STM X sin el último valor y que dos series concuerden significa que $|X_n - Y_m| < \varepsilon$.

Aquí ε es elegido por el usuario para dictaminar el umbral de lo que se considera parecido y δ se elige para decir cuanto se puede desplazar horizontalmente para acomodar los posibles desplazamientos de una serie respecto a otra.

Con esta última disimilitud se concluye el estudio de las distintas medidas para series temporales. Solo se han presentado las más usadas, para más ejemplos se puede consultar [23] donde se ven en particular las que se han visto e incluso las normas L_p multivariantes. A lo largo de esta sección se ha pretendido ir señalando sus características para hacer más fácil las respuestas a las preguntas iniciales de «¿qué medida usar?,¿cuál es la mejor?» sin embargo, estas siguen siendo complicadas. Hay varios artículos con el objetivo de comparar distintas medidas (tanto las vistas aquí como otras más) sobre una misma base de datos que pueden ayudar a contestar a las cuestiones como en [24] con un caso simulado o también [13].

Ciertas medidas o más bien técnicas que se han dejado sin mencionar por no encajar del todo en esta sección se verán en la siguiente pues se tratarán las distintas formas de extraer información para ver similitudes sin que haya necesariamente una similitud o disimilitud interviniendo (como era el caso de SVD). Se le dará importancia a modelos y las distancias quedan relegadas a un segundo plano.

3.3 Más allá de las variables continuas

Como ya se había mencionado al principio de la sección, solo se han visto las medidas que necesitan para variables continuas. Evidentemente, no todas las series temporales vienen dadas por variables continuas, muchas pueden venir dadas por variables discretas, categóricas, cualitativas, etc. Ya es una tarea ardua lidiar con este tipo de variables en bases de datos usuales. Si bien como hemos visto en el caso de las series temporales univariantes, esas mismas distancias les podrían ser aplicadas (en algunos casos) pero se olvidarían de la temporalidad de los datos y en el caso de STM las medidas les ocurre lo que al SVD, se basan en un modelo muchas veces estadístico [25]. Es decir, son (di)similitudes complejas con muchas consideraciones a hacer.

Tomando más conciencia aún sobre la complejidad de tratar con datos reales y diferentes tipos de variables, es de obligada mención el hecho de que no siempre hay solo variables continuas o categóricas, muchas veces se mezclan en la misma base de datos. Este es en gran parte el caso en estudios médicos. Para variables mixtas (que es como se le denomina a lo anterior) se propuso ya en 1971 una muy buena distancia, la de Gower [26]. Sin embargo, esta distancia no se puede usar en las series temporales por como está definida y como se define este tipo de series [27]. Ante este problema una de las soluciones más interesantes es la de usar los Modelos Markovianos Ocultos, una aproximación basada en un modelo estadístico a partir de cadenas de Markov ocultas, que se suele usar en los casos médicos [28].

En definitiva, hay muchas opciones donde elegir a la hora de ver cómo decidir que se entiende por parecerse. Muchos tipos de variables y para lidiar con ellos muchas soluciones distintas. La literatura es amplia en este aspecto gracias al potencial de las series temporales.

4

Estudio de técnicas de conglomerados

Acorde a 2.2.2 proporcionando una visión más amplía: se entiende una *técnica de conglomerados* como el método y algoritmo de aglomeración elegidos para obtener la información necesaria de los datos y separarlos en conglomerados. Siguiendo esta línea, la división que se hará en esta sección será parecida a la propuesta en [4]. Usualmente se clasifican como en [29] pero no es un enfoque adecuado para este documento aunque se hará una consideración sobre los distintos algoritmos que se suelen usar.

En este documento se dividirán las *técnicas de conglomerados* en tres categorías, las basadas en las formas de las series temporales, en características extraídas de estas y en modelos estadísticos sobre ellas. Esta división se hace acorde al uso de la medida de (di)similitud. En el primer caso la medida se usa para comparar la forma de las series per se o bien una aproximación de estas. En el segundo caso se usa para comparar un vector de características extraídas de las series y en el último para ver qué modelo estadístico se ajusta mejor en tanto que a generar las series temporales se refiere.

Se expondrán ejemplos de las principales técnicas en cada categoría.

4.1 Técnicas basadas en la forma

Las técnicas agrupadas bajo esta categoría además incluyen y/o son las mismas que las denotadas en la literatura como *raw-data-based* o *similarity-based* [30, 31], es decir, técnicas basadas en los datos sin procesar o en similitud. Se ha elegido denotarlas por técnicas basadas en la forma (de las series temporales) al considerarse un nombre mejor ajustado a los ejemplos que aquí se verán y a lo que se considera en este trabajo a este tipo de técnica.

Definición 4.1.1 (Técnica basada en la forma)**.** Se denotan técnicas basadas en la forma aquellas que comparan las formas de las series. Se consideran dos tipos: aquellas que usan los datos sin procesar comparando mediante una medida de (di)similitud y aquellas que usan modelos para aproximar las formas de la series comparando estas aproximaciones mediante medidas o divergencias.

Una *divergencia* es una medida de disimilitud que no cumple la propiedad simétrica aunque siempre se puede simetrizar.

Siguiendo la definición se verán dos ejemplos de técnicas, una para cada tipo.

4.1.1 Dynamic Time Warping (DTW)

Ya en 3.1.2 se hizo un primer acercamiento a DTW. Por el orden de complejidad de esta disimilitud se suele aplicar para series temporales cortas. Sin embargo, hay numerosos métodos usados para hacer de DTW un método más eficiente en tanto que al hallar el camino W se refiere. Se verán estos métodos junto con una definición más exhaustiva de DTW.

Dada que la diferencia entre usar DTW para series univariante o multivariantes radica en como se interpreta la matriz distancia entre dos series, se explicarán los conceptos para series univariantes para aligerar notación siendo estos siempre aplicables al caso multivariante.

Sean $u = (u_1, u_2, \ldots, u_T)$ e $v = (v_1, v_2, \ldots, v_S)$ dos series temporales y su respectiva matriz de distancias $D_{u,v}$.

$$D_{u,v} = \begin{pmatrix} d_p(u_T, v_1) & d_p(u_T, v_2) & \cdots & d_p(u_T, v_S) \\ \vdots & \vdots & \ddots & \vdots \\ d_p(u_2, v_1) & d_p(u_2, v_2) & \cdots & d_p(u_2, v_S) \\ d_p(u_1, v_1) & d_p(u_1, v_2) & \cdots & d_p(u_1, v_S) \end{pmatrix}.$$

Si bien la definición de la matriz puede parecer inusual proviene de [32], donde primero se define DTW, para clarificar las diferencias temporales entre las series. Es importante notar que el objetivo con el que se crea esta medida no es otro que el reconocimiento de voz y por tanto busca deformar una serie en otra eliminando así las fluctuaciones temporales propias del lenguaje hablado. Tomar la matriz de distancias de esta manera permite verla como una cuadrícula sobre el plano donde la primera columna y la última fila serían los ejes cartesianos, la serie u el eje de abscisas y v el de ordenadas.

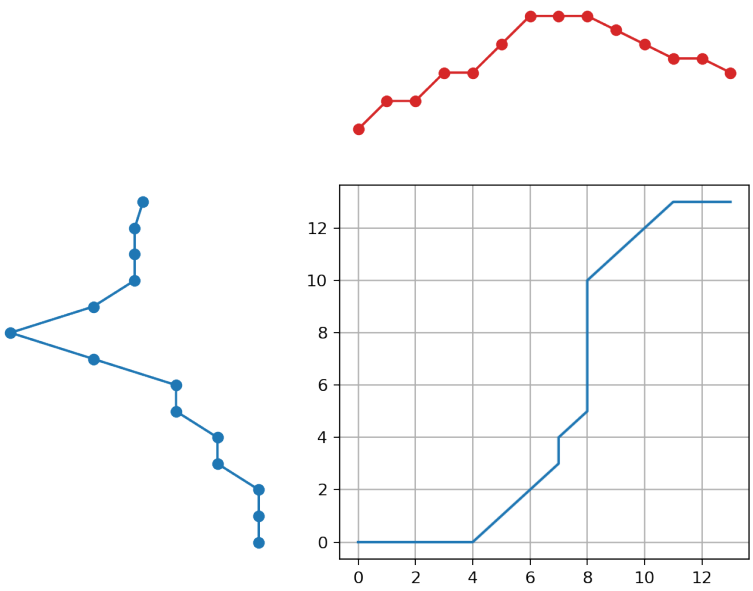


Figura 4.1: Visualización de las series temporales u (en rojo) y v (en azul) junto al camino de deformación entre ellas.

Sería entonces W el camino de deformación de la serie u a la serie v, obteniéndose una diagonal en el caso de no existir diferencias entre ellas.

Para obtener el camino se hace uso de programación dinámica donde se pretende encontrar el camino que optimice:

$$DTW(x,y) = \text{mín}\left(\frac{\sum_{i=1}^{K} w_i \cdot p_i}{\sum_{i=1}^{k} p_i}\right).$$

Donde p_i son pesos asociados a los pasos dados en el camino. Si todos los posibles pasos tienen el mismo coste la definición anterior se resume a la función 3.1, en caso contrario se usa para medir características como la flexibilidad.

Hay una gran cantidad de caminos de deformación posibles, muchos de los cuales si bien pueden tener un coste reducido representan una deformación sin coherencia con lo que se busca. Por esto, se imponen ciertas restricciones sobre W.

Restricciones sobre el camino de deformación

Sean $w_k = (i,j)$ y $w_{k-1} = (i',j')$ dos elementos del camino $W = \{w_1, w_2, \ldots, w_K\}$. Estos elementos han de cumplir las siguientes restricciones:

1. *Monotonía.* Se busca que el camino no retroceda en el tiempo, es decir, que no se repita una característica:
$$i' \leq i \text{ y } j' \leq j.$$

2. *Continuidad.* Asegura que no se omitan características importantes exigiendo:
$$i - i' \leq 1 \text{ y } j - j' \leq 1.$$

3. *Condiciones sobre los bordes.* Evita que la alineación solo considere una parte de las series:
$$w_1 = (1,1) \text{ y } w_K = (T,S).$$

4. *Ventana de deformación.* Se asume que un buen camino de deformación no se ha de alejar demasiado de la diagonal pues en caso contrario podría saltarse características dispares y estancarse en las que sean muy similares:
$$|i - j| \leq r \text{ donde } r > 0 \text{ es la longitud de la ventana.}$$

5. *Restricción sobre la pendiente.* Con esta restricción se busca evitar tanto que el camino sea excesivamente creciente dando lugar a que un patrón corto en u se corresponda con uno demasiado largo en v, como que sea demasiado suave ocurriendo el caso contrario. Así se restringirá el número de pasos consecutivos dados en la dirección de "abscisas" u "ordenadas". Es decir, para $w_1 = (i,j)$, $w_k = (i',j')$, dos elementos de W diferenciados $k = p + q$ pasos donde se han dado p pasos en la dirección de ordenadas y q en la de abscisas, se ha de cumplir que:
$$j - j' \leq n \text{ y } i - i' \leq m.$$
Suele ocurrir que se toma $m = n$.

Se obtiene de lo anterior un camino de deformación que alinea completamente una serie a la otra de manera continua evitando retrocesos en el tiempo, sin saltarse patrones relevantes o estancarse en estos.

Como se puede observar la elección de los valores de la ventana de deformación y la pendiente se deja libre. Cuales son los valores adecuados puede variar dependiendo de los datos y tomar distintos valores y hacer pruebas con un subconjunto de los datos puede ser de gran utilidad.

En el caso de la ventana de deformación se tiene de herencia del uso original el usar una ventana donde r es el 10 % de la longitud de las series temporales. Sin embargo, en [33] se discute que este valor no es necesariamente adecuado y que menores valores de r pueden ser beneficiosos llegando a proponer tomar r como el 5 % de la longitud de las series.

La ventana más comúnmente usada es la ventana Sakoe-Chiba [32]. Esta puede derivar en una imposibilidad de cumplir la restricción sobre los bordes en el caso de tener dos series temporales de distinto tamaño. Ante esto se proponen dos soluciones:

- En [34] se propone usar una ventana constituida por los puntos a lo largo de la diagonal serrada que une a (1,1) y (T, S) (tiene pendiente $\frac{S}{T}$ en vez de 1) que se encuentran contenidos en

  $$[(i, j - r), (i, j + r)]$$

  con (i, j) en la diagonal anterior.

- En [33] se argumenta que no existe un diferencia estadísticamente significativa entre usar las series temporales con longitudes dispares o usar interpolaciones de estas de manera que todas se ajusten a la longitud de la mayor de todas. Así, usando esta interpolación, no sería necesaria la consideración de una ventana que siga otra diagonal que la propia de una matriz cuadrada.

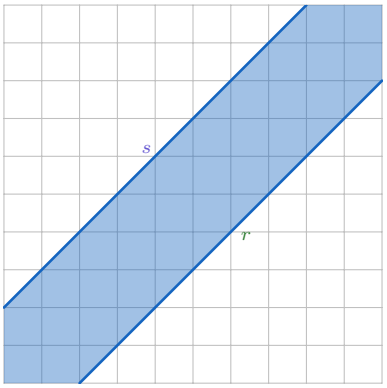


Figura 4.2: Banda de Sakoe-Chiba.

Otro tipo de ventana muy usada sobre todo en el campo de reconocimiento de voz es el paralelogramo de Itakura [35] el cual no se ve afectado por la posible diferencia en las longitudes de las series. Tomando dos series u y v de longitudes T y S respectivamente y dado un valor p que será el parámetro del paralelogramo. Se tiene que, suponiendo la matriz $D_{u,v}$ un rectángulo en el plano delimitado por los ejes cartesianos y las rectas $x = S$, $y = T$, el paralelogramo viene dado por los puntos contenidos en el área que encierran las rectas

$$\begin{aligned} r_1 : y &= \frac{x}{p} \\ r_2 : y &= x \cdot S \\ r_3 : y &= \frac{x}{p} + T - \frac{S}{p} \\ r_4 : y &= x \cdot p + T - p \cdot S \end{aligned}$$

Con respecto a la pendiente de la recta esta restricción ya va implícita al usar una ventana que no permita avanzar demasiado en las direcciones verticales y horizontales. Además al usar los pesos asociados a los pasos que se mencionaban al principio de manera que se penalice dar pasos en esas direcciones también se restringe la pendiente del camino de deformación.

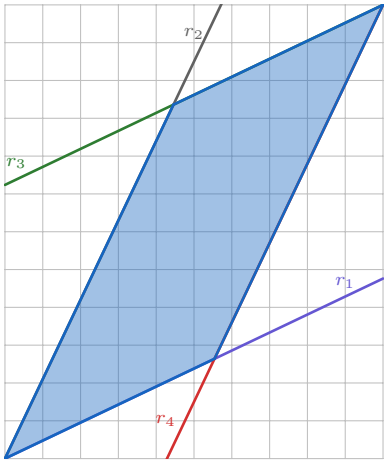


Figura 4.3: Paralelogramo de Itakura con $p = 2$.

Elección de los pesos asociados

Hay varias elecciones de los pesos como las mostradas en [34]. Las más usada son dos tipos de patrones simétricos:

- El primero, le asigna a todos los pasos el mismo valor, 1, lo que hace que se propicie los pasos diagonales.
- El segundo le asigna a la diagonal un peso de 2 y a desplazarse en la vertical u horizontal un peso de 1. Así le otorga el mismo peso a dar un paso en diagonal que ha darlo primero en vertical y luego en horizontal o viceversa.

Figura 4.4: Diferencia entre los dos tipos de patrones simétricos más comunes.

Límites inferiores

Si bien las restricciones anteriores proporcionan una mejoría en la eficiencia de DTW principalmente tienen de objetivo proporcionar el tipo de camino de deformación deseado. Para disminuir la complejidad hasta llegar al punto que esta pueda llegar a ser $O(T)$ [33] con T la longitud de la serie más larga

(asumiendo el uso de interpolación en caso de series de longitudes dispares). Los tipos de restricciones más usadas se pueden ver en [36] siendo estas la propuesta por Keogh y la propuesta en [37] que hace uso de la anterior y la mejora.

4.1.2 Modelos Markovianos Ocultos. MMO

Esta técnica si bien en base a un modelo se considera dentro de esta categoría ya que construye una aproximación a las formas de las series temporales.

Un MMO es un tipo de modelo que se basa en las cadenas de Markov [38]. La definición que se usará en este documento es la que sigue.

Definición 4.1.2 (Cadena de Markov)**.** Dado un proceso estocástico discreto $\{X_t\}_{t\in\mathbb{N}}$ que toma valores en un conjunto finito o numerable E llamado conjunto de estados. Se tiene que esta secuencia de variables aleatorias es una cadena de Markov si cumple la propiedad de Markov:

$$P\left(X_{n+1}=i|X_n=j,X_{n-1}=i_{n-1},\ldots,X_1=i_1\right)=P\left(X_{n+1}=j|X_n=i\right)$$

donde $i,j,i_{n-1},\ldots,i_1\ \in E$.

A lo anterior en la literatura se le conoce como cadena de Markov discreta. Este tipo de proceso estocástico se caracteriza por su falta de memoria pues la probabilidad que se de un estado u otro al dar un paso en el tiempo (de n a $n+1$) depende únicamente del estado anterior. Aquí las variables proporcionan la probabilidad que en el tiempo $n+1$ se de el estado j dado que en el tiempo anterior ocurriera el estado i. Estas probabilidades se pueden organizar en una matriz llamada matriz estocástica o de transición donde p_{ij} es la probabilidad anteriormente mencionada.

$$P=\begin{pmatrix} p_{11} & p_{12} & \cdots \\ p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

cumpliendo

$$\sum_{i=1}^{\infty} p_{ij}=1\ \forall j\in E.$$

Se ha de mencionar que se está asumiendo que las probabilidades de transición son independientes del instante de tiempo en la que se produzcan. Este puede no ser siempre el caso pues la matriz anterior puede depender del tiempo o covariables.

Una vez conocido el concepto de una cadena de Markov, un MMO se puede ver como un doble proceso estocástico donde hay un primer proceso oculto, los estados, compuesto por una cadena de Markov que solo se puede observar a través del segundo proceso, una variables cuyas distribuciones de probabilidad (probabilidades de emisión) vienen regidas por estos estados ocultos. Es decir, dadas unas variables aleatorias, su comportamiento viene regido por una cadena de Markov que no se puede observar.

Esto queda más claro a través de ejemplos. En [39] se dan varios distintos al que se proporcionará a continuación:

Uno de los motivos por lo que este tipo de técnica se usa en los contextos sanitarios es porque un individuo se puede considerar en uno de dos estados (sano o enfermo) de una cadena markoviana. Sin

embargo, este estado no se puede observar directamente, se necesita de ciertas variables como la temperatura corporal o un análisis de sangre cuyos resultados vendrían determinados por los estados de la cadena de Markov (si el paciente está sano o enfermo). Así, la probabilidad de una analítica con resultados preocupantes es menor si el paciente está sano que si estuviera enfermo.

Una vez se ha dado una intuición, se define un MMO como [38]:

Definición 4.1.3 (Modelo Markoviano Oculto)**.** Se entiende por Modelo Markoviano Oculto a un proceso estocástico consistentes de dos partes diferenciadas, una oculta y una observable, pero relacionadas entre sí en tanto que la observable depende de la oculta.

- La parte oculta consiste de una cadena de Markov finita $\{X_t\}_{t=1}^{K}$ con $E = \{e_1, \ldots, e_N\}$ estados y una matriz de transición P. Junto de la distribución inicial asociada, denotada como

$$\pi_i = P(X_1 = e_i)\ \forall e_i \in E.$$

- La parte observable está formada por el conjunto de variables $\{Y_t\}_{t=1}^{K}$ con $V = \{v_1, \ldots, v_M\}$ el espacio de posibles valores que toman y una matriz de emisión Q donde

$$q_{ij} = P(Y_t = v_j | X_t = e_i),\ \text{con } 1 \leq i \leq N,\ 1 \leq j \leq M.$$

Las matrices de transición y emisión pueden verse como las matrices de un grafo dirigido.

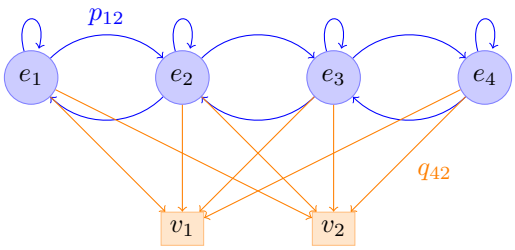
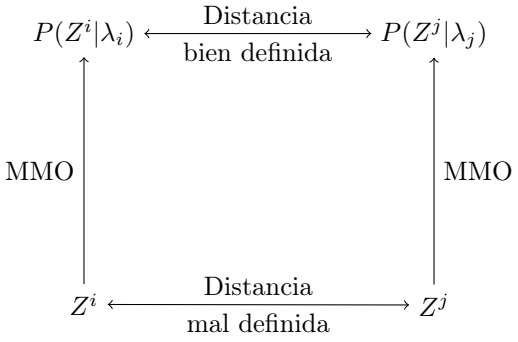


Figura 4.5: Un Modelo Markoviano Oculto con 4 estados que sólo emiten dos posibles valores.

En el diagrama anterior las flechas azules serían la matriz P donde la probabilidad de pasar del estado e_1 a e_2 es p_{12} y la de pasar del estado e_1 a e_3 sería 0. La matriz de emisión vendría dada por las flechas naranjas donde q_{42} es la probabilidad de que el estado e_4 produzca la observación v_2.

Ya entendida la base sobre la que se fundamenta la técnica, se ha de justificar porqué se entiende como basada en la forma de las series temporales.

Esta técnica está construida de manera tal que puede trabajar con variables categóricas y continuas. Es el primer caso el que más destaca ya que no se pueden entender como puntos en un espacio multidimensional como hasta ahora se ha hecho. Suponiendo así una serie temporal multivariante de variables categóricas el enfoque que se le da es el de una sucesión de observaciones de las variables Y_t con $t = 1, \ldots, T$ la longitud de la serie. Una vez se considera esto se le asocia el MMO más probable de generar esa sucesión de observaciones. Pudiendo pasar así de comparar series temporales a comparar las formas de las funciones de probabilidad de los MMO que, con mayor probabilidad, generan esas series temporales. Por esto, a pesar de asumir un modelo estadístico, se sigue considerando una técnica basada en la forma pues el MMO lo que hace es aproximar la serie temporal como tal. Es decir, se hace lo siguiente:



Siendo Z^i, Z^j dos series temporales y $P(Z^i|\lambda_i)$, $P(Z^j|\lambda_j)$ sus respectivos modelos asociados.

Para el caso de variables continuas, como en el reconocimiento de voz, existen las cadenas de Markov continuas y los Modelos Markovianos Ocultos Continuos (MMOC).

En esencia una cadena de Markov continua se basa en el mismo principio que la discreta [40].

Definición 4.1.4 (Cadena de Markov continua)**.** Dado un proceso estocástico continuo $\{X(t)\}_{t\geq 0}$ donde t es una variable aleatoria que indica el tiempo y un conjunto finito o numerable de estados E. Se dice que es una cadena de Markov continua si cumple la propiedad de Markov para cualquier conjunto finito $0 \leq t_1 < t_2 < \cdots < t_n < t_{n+1}$ de tiempos:

$$P\left(X(t_{n+1}) = j | X(t_n) = i, X(t_{n-1}) = i_{n-1}, \ldots, X(t_1) = i_1\right) = P\left(X(t_{n+1}) = j | X(t_n) = i\right),$$

donde $j, i, i_n, i_{n-1}, \ldots, i_1 \ \in E$.

Teniendo esto en cuenta un MMOC hace uso de lo anterior para lidiar con observaciones continuas. El concepto de MMO es el mismo pero ahora las observaciones son continuas por tanto, se suele asumir que dado un estado las observaciones son fruto de una mixtura de funciones gaussianas [41]. Así

$$q_j(x) = \sum_{k=1}^{K} c_{jk} N(x, \mu_{jk}, \Sigma_{jk}) \ 1 \leq j \leq N$$

es el modelo que aproxima la observación x dado el estado $j \in E$, $N(x, \mu_{jk}, \Sigma_{jk})$ es la función de densidad de una normal de media μ_{jk} y matriz de varianza Σ_{jk} y los c_{jk} son los pesos cumpliendo que

$$\sum_{k=1}^{K} c_{jk} = 1$$

para que se cumpla que $q_j(x)$ es una función de densidad.

Con esto es importante señalar que la duración de los estados puede venir modelada por una distribución de la familia exponencial [38]. Así se ha de entender que dada una serie temporal que depende de una variable continua, la forma de esta va a ser modelada por un mixtura de normales asociadas a cada estado donde cada uno de ellos ocurre dada una distribución de la familia exponencial (esta sería la probabilidad de emisión del caso discreto). Muchas veces esto hace que la estimación de parámetros se ralentice mucho decantándose en ese caso por una distribución uniforme de los tiempos entre estados [41].

Una vez establecido el modelo que se desea usar, es necesario estimar los parámetros. Para esta labor hay numerosas opciones de acuerdo al tipo de MMO que se elija pero lo más común es máxima verosimilitud

de los parámetros [31, 42, 43]. Es importante notar que el número de estados tiene que ser proporcionado para estos algoritmos por lo que se suele probar con varios y tomar la mejor opción.

Ahora se tiene un modelo para cada serie temporal que se denotará $P(Z^i|\lambda_i)$ donde Z^i es una serie temporal multivariante y λ_i son los parámetros asociados al modelo obtenido para esa serie (matriz de emisión, matriz de transición y distribución inicial). Se puede leer como la probabilidad de que dado el modelo λ_i este produzca la serie Z^i. De $P(Z^i|\lambda_i)$ se obtiene una función de densidad o de probabilidad (depende de si el MMO es continuo o discreto) por lo que se necesita de una medida que compare funciones de densidad mediante su forma. La medida más usada es una divergencia, la llamada divergencia de Kullback-Leibler [44].

$$d_{KL}(P(x), Q(x)) = \int_{\mathbb{R}} P(x)\log\left(\frac{P(x)}{Q(x)}\right) dx.$$

Observación. También existe para el caso discreto [45].

En la notación de este texto, dada una serie cualquier Z^i y dos modelos λ_i, λ_j donde el primero es el inicialmente asociado a Z^i y el segundo es el que se quiere comparar, que a su vez está asociado a Z^j. Se tiene que la distancia Kullback-Leibler sería

$$d_{KL}(P(Z^i|\lambda_i), P(Z^i|\lambda_j)) = \int_{\mathbb{R}} P(Z^i|\lambda_i)\log\left(\frac{P(Z^i|\lambda_i)}{P(Z^i|\lambda_j)}\right) dZ^i.$$

Esto se puede interpretar como la diferencia entre usar el modelo asociado a la serie Z^i, λ_i, y usar el modelo λ_j para generar Z^i. Es decir, diferencia las formas de las series pues a mayor diferencia entre las series más diferente serán los modelos que las aproximan y por tanto va a ser menos probable que un modelo genere a la otra serie y viceversa.

El uso de esta medida puede verse en [28] donde se da una aproximación pues es computacionalmente costosa calcularla. También proporciona la forma simétrica de esta divergencia para así poder tratarla como disimilitud.

Ya vistos un ejemplo de cada uno de los tipos de enfoques de las técnicas basadas en la forma se concluye esta sección.

4.2 Técnicas basadas en características

En esta categoría se encuentran técnicas apropiadas para series temporales largas ya que en vez de tratar con la totalidad de las series o un modelo, definen la disimilitud a través de características extraídas de ellas, reduciendo así la dimensión del objeto de estudio.

Definición 4.2.1 (Técnica basada en características)**.** Se denominan técnicas basadas en las características a aquellas técnicas que extraen características (estadísticas o no) de las series temporales para aunarlas en un vector de tamaño menor que el de la serie sobre el cual se define la medida a usar.

Dado que se necesitan que las propiedades sean significativas para las series que se trabajan este tipo de técnica es muy dependiente al tipo de datos a los que se vaya a aplicar, es decir, son dependientes de la aplicación. Esto se verá en los ejemplos provistos.

En [46] se hace un análisis de conglomerados de las medidas por hora del consumo de 1035 usuarios durante 84 días. La longitud de las series temporales es muy elevada por lo que se inclinan por hacer una

extracción de características a través de 7 medidas calculadas semanalmente: media, desviación estándar, asimetría, curtosis, caos, energía y periocidad.

Las 4 primeras son características estadísticas ya conocidas, las 3 últimas son medidas asociadas a series temporales.

- La primera, *caos*, se cuantifica a través del máximo exponente de Lyapunov [47] que cuantifica si un sistema es o no caótico, es decir, si es sensible a pequeñas variaciones en las condiciones iniciales. Así, tomando una serie temporal $x = (x(1), x(2), \ldots, x(T))$ y una variación en las condiciones iniciales $x_\varepsilon(0)$ de manera que $|x(0) - x_\varepsilon(0)| = \varepsilon$ se tiene que viene dado por [48]

$$\lim_{t\to\infty}\left(\lim_{\varepsilon\to 0}\frac{1}{t}\ln\left(\frac{|x(t)-x_\varepsilon(t)|}{\varepsilon}\right)\right).$$

- La *energía* se basa en la transformada de Fourier 3.1.2 para tener una característica que mida la periocidad pero basada en ver la serie como una señal:

$$\frac{\sum_{f=1}^{m}|X_f|^2}{|w|}$$

  donde w es la longitud de la duración de la semana sobre la que se estén extrayendo las medidas. La transformada de Fourier permite pasar de un dominio temporal a un dominio frecuentista donde estudiar esta característica resulta más sencillo.

- Finalmente, la *periodicidad* sigue en el análisis de frecuencia anterior. Para determinar esta característica se pasa del dominio temporal al frecuentista (a través de la transformada de Fourier [49]) y se hace uso de un *periodogram*, un estimador de la densidad espectral (que mide la autocorrelación de una serie temporal o el poder de una frecuencia), midiendo la correlación entre las serie temporal y ondas de funciones seno y coseno dada una (o un conjunto de) frecuencia(s) [50]. Así se estima, si existe, la frecuencia con mayor poder sobre la serie y de ella se obtiene el periodo.

Las características anteriores entran dentro de un conjunto de características globales que se usan de manera más general para extraer propiedades de las series pues pueden ser usadas en gran variedad de datos de manera significativa. Destacando algunas de las nombradas en [51].

- *Tendencia*. Se define como un cambio a largo plazo en la media de los valores, ya sea creciente o decreciente. Una forma de estimarla suele ser usar el spline de regresión penalizado [52] para así ver si la serie tiende a crecer o decrecer.

- *Estacionalidad*. Se entiende por patrones que se repiten dado un tiempo fijo. Así si la forma de la serie se repite cada año, semana, u hora la serie está influenciada por una componente de estacionalidad. Para estimarla se estudia las autocorrelaciones de la serie buscando un coeficiente de autocorrelación (parcial o total) elevado dado el desfase estacional [53]. Es importante diferenciar estacionalidad y periodo ya que la primera se refiere al patrón que es repetido y la segunda al tiempo ocurrido entre patrones recurrentes. Así si una serie repite una forma de manera anual tendrá esa forma como estacionalidad y un periodo de un año.

- *Autocorrelación*. Dado un intervalo de tiempo $[t, t + k]$ para un k variable comprueba si existe correlación entre los valores x_t y x_{t+k}. Esta correlación puede darse solo en ese intervalo de tiempo, o darse distintas en diferentes intervalos consecutivos a lo largo de la serie temporal.

Muchas de las propiedades mencionadas proviene de una rama del análisis de las series temporales llamado análisis espectral donde se pretende extraer comportamientos periódicos.

En [46, 54] se ejemplifica el uso de las medidas ya mencionadas tanto para el caso univariante como multivariante. J. Timmer et al [55] proporcionan una clara división entre características del dominio temporal y frecuentista. Si bien los anteriores artículos muestran la versatilidad de las características, J. Jeong et al [56] usa análisis de información mutua para extraer características sobre las series temporales de electrogastogramas a pacientes con Alzheimer mostrando como el tipo de datos que se pretendan analizar influye mucho en las técnicas usadas.

A la hora de usar una técnica basada en características la literatura es extensa y se hace de vital importancia conocer la naturaleza de los datos. Una vez se ha superado este paso, generalmente se introducen en un vector y se usa la distancia euclídea junto a uno de los algoritmos usuales de clasificación para realizar el Análisis de Conglomerados.

4.3 Técnicas basadas en modelos

Es muy importante diferenciar el enfoque de estas técnicas al dado en el uso de modelos para técnicas basadas en la forma. Este último aproxima la forma de cada serie con un modelo y usa este para definir una nueva distancia a partir de la cual realizar el análisis mientras que las técnicas que se presentan asumen un único modelo con varias componentes que genera a los datos y la clasificación ocurre según que componente genere con mayor probabilidad los datos.

Definición 4.3.1 (Técnica basada en modelo)**.** Se denotan técnicas basadas en modelos a aquellas que asumen que los datos están generados por un modelo estadístico que se pretende recuperar a través de estos. Los modelos constan de varias componentes asociadas cada una a un conglomerado y se pretende encontrar los datos que mejor se ajusten a cada componente generando así la división.

Una de las técnicas mas usadas tanto a la hora de decidirse por un modelo como a la hora de como hallar sus parámetros es el que se conoce en la literatura como EMGMM, *Expectation Maximization Gaussian Mixture Model*. Un modelo que asume una mixtura de normales sobre los datos y usa el algoritmo EM para aproximar sus parámetros.

4.3.1 Modelo de Mixturas Gausianas. MMG

Un modelo de mixtura es un modelo donde se supone los datos generados por un conjunto de distintas funciones de densidad, $\{f_1, \ldots, f_K\}$, de manera que dado un vector aleatorio, $\underline{X} = (X_1, X_2, \ldots, X_t)$ de t variables aleatorias se tiene que

$$f(\underline{x}) = \sum_{k=1}^{K} \tau_k f_k(\underline{x})$$

con $\underline{x} = (x_1, x_2, \cdots, x_t)$ es su función de densidad cumpliendo que

$$\tau_k \geq 0 \ \forall k \quad \text{y} \quad \sum_{k=1}^{K} \tau_k = 1.$$

Ya establecido el modelo y el número de funciones de densidad que lo componen el objetivo es dado unos datos que se asumen independientes e idénticamente distribuidos, $\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n\}$, estimar los parámetros de manera que se maximice la verosimilitud. Así en el caso de MMG se tiene que la función de verosimilitud viene dada por [57]:

$$L(\theta_1, \ldots, \theta_K; \tau_1, \ldots, \tau_K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \tau_k \phi_k(\underline{x}_i|\theta_k).$$

Donde $\theta_k = (\mu_k, \Sigma_k)$ son los parámetros correspondientes a la k-ésima componente de la mixtura, τ_k denota la probabilidad de que el elemento $\underline{x}_i$ pertenezca a la k-ésima componente y

$$\phi_k(\underline{x}_i|\mu_k, \Sigma_k) = \frac{e^{\frac{-1}{2}(\underline{x}_i - \mu_k)^\top \Sigma_k^{-1}(\underline{x}_i - \mu_k)}}{\sqrt{2\pi|\Sigma_k|}}.$$

Se tiene que por la naturaleza de este modelo los conglomerados estarán concentrados en las medias de las distribuciones con mayor concentración alrededor de estos puntos. Además las formas de estos vendrán dadas por las matrices de varianzas y covarianzas. Ante esto se propone en [57] una manera general de enfocarlo realizando la descomposición espectral de las varianzas de manera que

$$\Sigma_k = D_k \Lambda_k D_k^\top.$$

Tras esto se toma λ_k como el primer autovalor de Σ_k y redefiniendo $\Lambda_k = \lambda_k A_k$ donde A_k es una matriz diagonal con elementos proporcionales a los autovalores. Obteniéndose

$$\Sigma_k = D_k \lambda_k A_k D_k^\top.$$

De manera que la matriz de autovectores D_k determina la orientación del k-ésimo conglomerado, A_k su forma y λ_k su volumen.

Otorgándole y quitándole libertad a los distintos elementos que forman la descomposición, tratándolos como parámetros independientes, se obtienen formas diferentes. En [58] se clasifican los distintos modelos. Empezando por el más sencillo $\Sigma_k = \lambda I$ que indica que todos los conglomerados son circulares de idénticas dimensiones. Si se permite que λ_k varíe en cada conglomerado entonces estos serán de dimensiones cambiantes. Tomando $\Sigma_k = \lambda A$ es el primer caso pero no teniendo que ser la forma necesariamente circular. De igual manera se puede dejar variar λ en los K conglomerados obteniéndose mismas formas pero distintos volúmenes. O variar A dando lugar a mismo volumen diferentes formas. En el caso de variar ambos se mantendría la orientación que estos siguen. Llegando a las formas más complejas se tiene ya la descomposición completa $\Sigma_k = D^\top \lambda A D$ por lo que ahora puede cambiar la orientación, no ha de ser necesariamente la de los ejes. Dejando D_k variar se obtiene que cada conglomerado puede tener una orientación propia. El resto de combinaciones son las del caso anterior sumándole el hecho de la orientación.

Una vez se hace la elección del modelo que se va a suponer cumplen los datos, se necesita estimar los parámetros de las K funciones normales. Para ello se usa el algoritmo EM, Esperanza Maximización. Esta es la elección usual ante los modelos de mixturas.

Algoritmo EM.

Este algoritmo [59] en el caso que ocupa a este documento se puede ver como sigue.

Dados los datos iid anteriores $\{\underline{x}_1, \ldots, \underline{x}_n\}$ se tiene que estos son los datos observados pero están incompletos. Los datos completos se entiende por $\underline{y}_i = (\underline{x}_i, \underline{z}_i)$ donde $\underline{z}_i = (z_{i1}, \ldots, z_{iK})$ es un vector cumpliendo que

$$z_{ik} = \begin{cases} 1 & \text{si } \underline{x}_i \text{ pertenece al } k\text{-ésimo conglomerado} \\ 0 & \text{en caso contrario} \end{cases}$$

Es decir, son las etiquetas que se buscan en el Análisis de Conglomerados. Por tanto la función de verosimilitud que se quiere maximizar cambia. Se puede definir el vector $\underline{Z} = (Z_1, \ldots, Z_k)$ como un vector aleatorio que sigue una distribución multinomial con probabilidades $\tau_1, \ldots, \tau_K$. De esta manera, la función que se quiere maximizar es la función de densidad conjunta de $\underline{X}$ y $\underline{Z}$:

$$f(\underline{x}_i, \underline{z}_i) = f(\underline{x}_i|\underline{z}_i) \cdot p(\underline{z}_i),$$

es decir

$$f(\underline{y}_i) = \prod_{k=1}^{K} (\phi_k(\underline{x}_i; \mu_k, \Sigma_k))^{z_i k} \prod_{k=1}^{K} \tau_k^{z_{ik}}.$$

A esta función de densidad se le aplica la log-verosimilitud sobre la que actuará el algoritmo EM:

$$l(\theta, \tau|y) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log(\tau_k \phi_k(\underline{x}_i\ \theta_k)).$$

Donde $\theta = (\theta_1, \ldots, \theta_K)$ y $\tau = (\tau_1, \ldots, \tau_K)$ El algoritmo consta de dos pasos.

- *Esperanza.* Este paso estima los datos no observados. Dada una aproximación de los parámetros $\hat{\theta}$ y $\hat{\tau}$, estima la probabilidad condicionada de que $\underline{y}_i$ pertenezca al k-ésimo conglomerado.

$$\hat{z}_{ik} = \frac{\hat{\tau}_k \phi_k(\underline{y}_i|\hat{\theta}_k)}{\sum_{j=1}^{K} \hat{\tau}_k \phi_j(\underline{y}_i|\hat{\theta}_j)} = E(z_{ik}|\underline{y}_i, \hat{\theta}_k).$$

- *Maximización.* Este paso maximiza la verosimilitud de la función

$$l(\theta, \tau|\underline{y})$$

  tomando $z_{ik} = \hat{z}_{ik}$. Así se obtiene la aproximación de los parámetros necesarios para el paso E.

Como es evidente un paso necesita del otro por lo que para iniciar el algoritmo se puede o bien empezar por el paso M proporcionando una clasificación inicial o bien empezando por el paso E proporcionando una estimación inicial de los parámetros.

Los pasos anteriores se repiten hasta obtener convergencia. En [60] se demuestra que repetir los pasos E y M llevan a un máximo local de la función de log-verosimilitud de los datos observados. Su función de verosimilitud es

$$L_O(\underline{x}|\theta) = \int L_C(\underline{y}|\theta)$$

donde

$$L_C(\underline{y}_i|\theta) = \prod_{i=1}^{n} f(\underline{y}_i|\theta)$$

es la función de verosimilitud de los datos completos, con f la función de distribución de la mixtura. Una vez alcanzado ese máximo se tendrían los parámetros que mejor hacen que el modelo y los datos se ajusten.

Modelo óptimo

Una vez obtenido los parámetros se tiene que a priori se había impuesto forma y número de conglomerados en los que debían encajar. Lo usual es probar con distinto número de conglomerados, K, y distintos modelos en cada elección de número de conglomerados. Para discernir si un modelo es mejor que otro se usan criterios como el BIC, *Bayesian Information Criterion*. Atendiendo a este valor se decide el mejor modelo. Una vez obtenido este asociamos cada elemento a su grupo correspondiente de manera que si $\underline{x}_i$ pertenece a la componente j-ésima, z_{ij} es el máximo valor de todos los z_{ik}.

4.3.2 Otros modelos de mixturas

Además de el modelo que se ha expuesto, las mixturas es una herramienta muy usada. Se puede aplicar con un enfoque diferente mediante mixturas de ARMA y MMO.

Las mixturas que se muestran en [61] se basan en un modelo de autorregresión y media móvil, ARMA por sus siglas en inglés. Este modelo se denota ARMA(p, q) dado por la composición de un modelo AR(p) y otro MA(q) [62]. Así, una serie temporal $x = (x_1, \ldots, x_T)$ vendría modelada por

$$x_t = \phi_0 + \sum_{j=1}^{p} \phi_j x_{t-j} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \; t = 1, \ldots, T.$$

Donde la primera parte conforma el modelo AR(p) de autorregresión,

$$x_t = \phi_0 + \sum_{j=1}^{p} \phi_j x_{t-j}$$

y la segunda parte el modelo de media móvil MA(q)

$$x_t = e_t + \sum_{j=1}^{q} \theta_j e_{t-j}$$

donde los e_i son un conjunto de valores independiente e idénticamente distribuidos conforme a una normal de varianza σ.

Así se aproxima la forma de la serie temporal sin embargo se puede definir una mixtura a partir de esta aproximación si se supone que cada conglomerado viene definido por un modelo ARMA distinto. Ocurriría igual que en el caso de MMG, se extraerían los parámetros $\{\phi_0, \ldots, \phi_p, \theta_1, \ldots, \theta_q, \sigma\}$ para el caso de cada conglomerado y se asociaría cada serie temporal al modelo que mejor se ajustara. Para más detalles en [61] se proporciona la descripción detallada de la mixtura como se ha mostrado en el caso MMG.

Las mixturas de MMO (Modelos Markovianos Ocultos) se basan en la misma premisa, asociar un modelo a cada conglomerado que estará conformado por las series temporales que vengan generadas por este con mayor verosimilitud. En [63] se ve esto con más detalle.

En ambos caso el algoritmo para obtener los parámetros sigue siendo el EM.

Observación. Además de las mixturas hay otros muchos tipos de técnicas basadas en modelo como se muestra en [64].

4.3.3 Algoritmos de conglomeración

Una vez que el método de definir una distancia o asociar un modelo se ha elegido el siguiente gran paso es la elección del algoritmo que se aplicará bajo esas condiciones. Escoger entre los múltiples tipos de algoritmos es una tarea no trivial ya que estos determinarán qué se entiende por distancia entre los conglomerados y cómo se separan o unen los objetos, entre otras cosas teniendo un gran impacto en el resultado final.

Si bien una visión extensa de estos algoritmos no es de interés en el caso que nos ocupa, es esencial tenerlos presente. Los principales algoritmos incluyendo el que se usará más adelante en la parte práctica se pueden ver en [65].

5

Medidas de evaluación

Como ya se ha expuesto el Análisis de Conglomerados es un método no supervisado de clasificación que busca encontrar una buena y significativa partición de los datos. Sin embargo, distintas técnicas y algoritmos de clasificación se pueden aplicar a una misma base de datos pudiendo variar los resultados obtenidos. Además, la efectividad de los anteriores va ligada al tipo de datos, los parámetros elegidos, etcétera. Con esto se plantea el problema de poder discernir que partición es mejor.

No suficiente con lo anterior, como se ha visto en las técnicas basadas en mixturas, muchas veces se ha de determinar a priori el número, K, de conglomerados necesitando por tanto una medida sobre como de bien se ajustan los datos a una división en K grupos.

Por tanto, una vez realizado un análisis sobre los datos se hace necesario saber si tanto la cantidad de conglomerados es la adecuada como la partición con dicha cantidad es buena, es decir, si la estructura es la mejor. Para enfrentarse a este problema y seleccionar la mejor opción se usan las medidas de evaluación.

Si bien en la literatura se da una clasificación en tres tipos distintos [66, 67]: interno, externo y relativo. En este documento se consideraran solo dos tipos externos e interno. Se entenderá por seguir un criterio interno que las medidas usan sólo la información intrínseca a los datos (e.g. matriz de distancia). Por otro lado si el criterio es externo la información se extrae de la verdadera clasificación de los datos que se supone conocida. En último lugar una medida relativa se considera un criterio interno usado con el objetivo de elegir la mejor entre distintas estructuras predeterminadas. Se omitirán las medidas de evaluación externas ya que no se adaptan al problema.

5.1 Medidas de evaluación internas.

Muchas de estas medidas tienen como base la relación entre cercanía de los objetos dentro de cada conglomerado y la lejanía de los conglomerados entre ellos, es decir, compacidad y separación. Un enfoque distinto y más actual son las medidas basadas en la estabilidad de las estructuras.

- **Índice de Dunn.** Este índice es propuesto por J.C. Dunn en 1974 [68] para encontrar conglomerados bien separados y compactos. Para ello usa la relación entre la distancia mínima entre conglomerados y el mayor diámetro de estos.

$$Dunn = \min_{1\leq i,j\leq G} \left\{ \frac{d(C_i, C_j)}{\max_{l=1,\ldots,G} diam(C_l)} \right\}.$$

Donde $d(C_i, C_j) = \text{mín}_{x \in C_i, y \in C_j} d(x, y)$ con $d(x, y)$ una distancia o disimilitud y $diam(C_l)$, el diámetro de un conglomerado, considerada una medida de dispersión, viene dada por $diam(C_l) = \text{máx}_{x,y \in C_l} d(x, y)$. Según Dunn, una estructura es buena si existe una separación notable entre conglomerados y los diámetros de estos son pequeños, es decir, se busca aumentar la distancia interconglomerado y reducir la intraconglomerados por lo que hallar el índice de Dunn óptimo es equivalente a hallar el máximo.

- **Índices similares al de Dunn.** El índice anterior presenta dos problemas: el coste computacional de los cálculos necesarios y la alta sensibilidad al ruido. Esta última puede contribuir a un aumento notable del diámetro de un conglomerado. Ante esto en [69] se presentan varias soluciones usando teoría de grafos.

  Para entender los índices que proponen Pal y Biswas es necesario definir ciertos conceptos previos.

  Definición 5.1.1 (Grafo)**.** Un grafo $G = (V, A)$ es un par donde V es un conjunto $V = \{v_1, \ldots, v_m\}$ denotado el conjunto de vértices y $A = \{a_1, a_2, \ldots, a_n\}$ es el conjunto de aristas. Una arista $a_k = \{v_i, v_j\}$ conecta a los vértices v_i y v_j.

  Definición 5.1.2 (Grafo ponderado)**.** Un grafo ponderado es la tupla ordenada $G = (V, A, W)$ de manera que $G = (V, A)$ es un grafo y $W = \{w_1, \ldots, w_n\}$ son los pesos asociados a cada una de las aristas de G con w_i siendo el correspondiente a a_i.

  Definición 5.1.3 (Grafo completo)**.** Se dice de $G = (V, A)$ es un grafo completo si $\forall v_i, v_j \in V$ con $v_i \neq v_j$ $\exists a_k = \{v_i, v_j\} \in A$.

  Definición 5.1.4 (Subgrafo)**.** Un grafo $G_s = (S, A_s)$ es un subgrafo de $G = (V, A)$, $G_s \subset G$, si $S \subset V$ y $A_s \subset A$ siendo $A_s = \{a_k = \{v_i, v_j\} | a_k \in A, v_i, v_j \in S\}$.

  Definición 5.1.5 (Camino)**.** Un camino es una secuencia de aristas $(a_1, a_2, \ldots, a_{q-1})$ que une $(v_1, v_2, \ldots, v_q)$ vértices distintos de manera que $e_i = \{v_i, v_{i+1}\}$ y v_1, v_q solo son adyacentes a la primera y última arista respectivamente.

  Definición 5.1.6 (Árbol)**.** Un árbol es un grafo $T = (V, A)$ donde todo par de vértices está conectado por un único camino.

  Así dada una base de datos, Ξ, dividida en G grupos $C_1, C_2, \ldots, C_G$ se puede ver cada grupo C_k como un grafo completo ponderado $G_k = (V_k, A_k, W_k)$ con V_k los datos de C_k y las aristas ponderadas mediante la disimilitud entre los datos. Sobre estos grafos completos (o nubes de punto) se usan tres tipos de estructuras dando lugar a través de cada una de ellas a un índice parecido al de Dunn pero con menos sensibilidad al ruido. Las estructuras usadas son las que siguen.

  Definición 5.1.7 (Árbol recubridor mínimo)**.** Un árbol recubridor mínimo (ARM) sobre un grafo ponderado $G = (V, A, W)$ es un árbol $T = (V_t, A_t, W_t)$ tal que $T \subset G$, $V_t = V$ y el coste asociado a T sea mínimo. Se define este coste como

  $$c_T = \sum_{w_t \in W_t} w_t.$$

  Definición 5.1.8 (Grafo de vecindad relativa)**.** Propuesto por Toussaint en 1980 [70]. Sean x_{ki}, x_{kj} dos elementos de $C_k = \{x_{k1}, \ldots, x_{km}\}$. Estarán conectados en el grafo de vecindad relativa (GVR) si cumplen que son vecinos relativos, es decir,

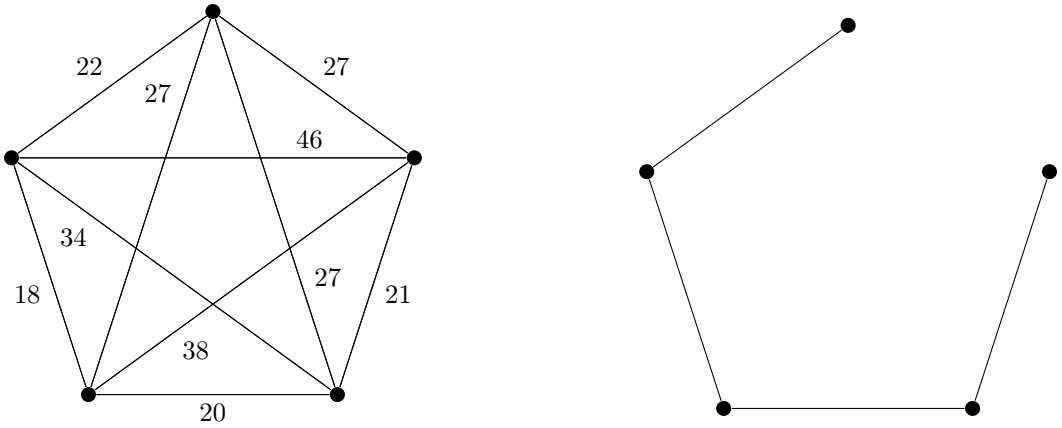
  $$d(x_{ki}, x_{kj}) \leq \text{máx}\{d(x_{ki}, x_{kh}), d(x_{kj}, x_{kh}) \ \forall h, h \neq i, h \neq j\}$$

  donde $d(\cdot, \cdot)$ es una disimilitud.

  Definición 5.1.9 (Grafo de Gabriel)**.** Debe su nombre a K. Ruben Gabriel que propone este grafo en 1969 junto a Sokal [71]. Sean x_{ki}, x_{kj} dos elementos de $C_k = \{x_{k1}, \ldots, x_{km}\}$. Estarán conectados en el grafo de Gabriel (GG) si cumplen que

  $$d^2(x_{ki}, x_{kj}) < d^2(x_{ki}, x_{kh}) + d^2(x_{kj}, x_{kh}) \ \forall h, h \neq i, h \neq j$$

  donde $d^2(\cdot, \cdot)$ es la distancia euclídea.



(a) Grafo ponderado inicial.

(b) Árbol recubridor mínimo.

(c) Grafo de vecindad relativa.

(d) Grafo de Gabriel.

Figura 5.1: Ejemplo de los grafos anteriores dado uno ponderado.

La modificación del índice de Dunn se hace a través de los grafos discutidos cambiando el concepto de diámetro de un conglomerado. Así para el caso de ARM se parte del grafo completo ponderado asociado al k-ésimo conglomerado $G_k = (V_k, A_k, W_k)$, GVR y GG necesitan únicamente de la nube de puntos. En cada caso se realiza el mismo procedimiento, se verá el asociado a GVR.

Sea $GVR = (V_k^{GVR}, A_k^{GVR}, W_k^{GVR})$ el GVR obtenido a partir de la nube de puntos del conglomerado C_k. Se define el diámetro de C_k como

$$diam^{GVR}(C_k) = \text{máx}\{w^{GVR} \in W_k^{GVR}\}.$$

Así el índice de Dunn modificado se define como

$$Dunn^{GVR} = \min_{1\leq i,j\leq G} \left\{ \frac{d(C_i, C_j)}{\text{máx}_{l=1,\ldots,G}\, diam^{GVR}(C_k)} \right\}.$$

Análogamente se definen $Dunn^{GG}$ y $Dunn^{ARM}$ cuyas interpretaciones son análogas al índice de Dunn buscando maximizar el valor con respecto al número total, G, de conglomerados.

- **Índice de Davies-Bouldin.** David L. Davies y Donald W. Bouldin introducen este índice, abreviado como índice DB, en 1979 [72] con la idea que una buena estructura debe poseer alta separación interconglomerado, compacidad y homogeneidad intraconglomerado. Al contrario que el índice de Dunn, el DB encuentra el valor óptimo del número de conglomerados minimizando como se verá a continuación.

Para comprender el índice DB es necesario primero saber lo que es una medida de dispersión y la definición que Davies y Bouldin proporcionan de medida de similitud entre conglomerados.

Definición 5.1.10 (Medida de dispersión)**.** Sea un conglomerado C conformado por los elementos $x_1, x_2, \ldots, x_n \in \mathbb{R}^p$. Se dice que la aplicación $s : C \to \mathbb{R}$ es un medida de dispersión si cumple:

- *No negatividad:*
$$s(x_1, \ldots, x_n) \geq 0.$$
- *Reflexividad:*
$$s(x_1, \ldots, x_n) = 0 \Leftrightarrow x_i = x_j \ \forall x_i, x_j \in C.$$

Definición 5.1.11 (Medida de similitud entre conglomerados)**.** Sean unos datos $\Xi \subset \mathbb{R}^p$ divididos en G grupos disjuntos $C_1, C_2, \ldots, C_G$, sea $s_i \ \forall i = 1, 2, \ldots, G$ una medida de dispersión de los distintos conglomerados y d_{ij} la distancia entre los grupos C_i y C_j tomada como la distancia entre dos vectores representativos de cada uno de ellos. Se dice que la aplicación $R(s_i, s_j, d_{ij}) : \mathbb{R}^3 \to \mathbb{R}$ es una medida de similitud entre los conglomerados C_i y C_j, denotada de manera abreviada como R_{ij}, si cumple las siguientes propiedades:

1. $R(s_i, s_j, d_{ij}) \geq 0$.
2. $R(s_i, s_j, d_{ij}) = R(s_j, s_i, d_{ji})$.
3. $R(s_i, s_j, d_{ij}) = 0 \Leftrightarrow s_i = s_j = 0$.
4. Si $s_j = s_k$ y $d_{ij} < d_{ik}$ entonces $R(s_i, s_j, d_{ij}) > R(s_i, s_k, d_{ik})$.
5. Si $d_{ij} = d_{ik}$ y $s_i > s_j$ entonces $R(s_i, s_j, d_{ij}) > R(s_i, s_k, d_{ik})$.

Cumpliendo las características anteriores Davies y Bouldin proporcionan una medida de similitud

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

la cual cumple que al buscar alta compacidad y separación interconglomerados, es decir, valores pequeños de la medida de dispersión y elevados de d_{ij} por lo que valores pequeños de R_{ij} indicarán que la división entre los conglomerado C_i y C_j es buena.

Siguiendo este razonamiento definen el índice como

$$DB = \frac{1}{G} \sum_{i=1}^{G} \max_{j \neq i} \{R_{ij}\}$$

la media de las similitudes de los conglomerados con el más parecido a ellos, que por lo anterior hallará su valor óptimo en el mínimo.

En [72] se proporcionan ejemplos de medida de dispersión y de distancia intraconglomerado a usar para calcular el índice.

- **Índices similares a DB.** Pal y Biswas al igual que con el índice de Dunn proponen en [69] índices basados en el índice DB pero haciendo uso de ARM, GVR y GG para definir la medida de dispersión de los conglomerados como el diámetro según cada grafo.

- **Índice de Caliński-Harabasz.** T. Caliński y J. Harabasz proporcionan este índice en 1974 [73] basándose en la matriz de dispersión de los datos. Sea $\Xi \subset \mathbb{R}^p$ el conjunto de N datos divididos en G grupos disjuntos $C_1, C_2, \ldots, C_G$ el índice CH viene dado por

$$CH = \frac{traza(B)}{traza(W)} \cdot \frac{N - G}{G - 1}$$

Siendo W la matriz de dispersión intraconglomerados

$$W = \sum_{i=1}^{G} \sum_{l=1}^{N_k} (x_{il} - \bar{x}_i)(x_{il} - \bar{x}_i)^\top$$

con N_k el número de elementos del grupo C_k, x_{il} el l-ésimo elemento del i-ésimo conglomerado y $\bar{x}_i$ el vector media muestral del conglomerado C_i.

La matriz B es la matriz de dispersión interconglomerado

$$B = \sum_{i=1}^{G} N_i(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^\top$$

donde $\bar{x}$ es el vector media muestral de Ξ.

Así una buena estructura será aquella que maximice el índice al aumentar la distancia entre los distintos grupos (aumentando B) y disminuya la distancia entre los objetos que los conforman (reduciéndose W).

- **Índice de silueta.** Sea $\Xi \subset \mathbb{R}^p$ el conjunto de N datos divididos en G grupos disjuntos $C_1, \ldots, C_G$ se define la silueta de una estructura como [74]

$$Silueta = \frac{1}{N} \sum_{j=1}^{N} s(j) \in [-1, 1]$$

donde

$$s(i) = \frac{b_k(i) - a_k(i)}{\text{máx}\{b_k(i), a_k(i)\}} \in [-1, 1]$$

siendo $a_k(i)$ una medida de la compacidad y $b_k(i)$ de la buena separación y clasificación de un objeto i al grupo C_k:

$$a_k(i) = \frac{1}{N_k - 1} \sum_{j \in C_k / i} d(i, j),$$

$$b_k(i) = \min_{l \neq k} \left\{ \sum_{j \in C_l} \frac{d(i, j)}{N_l} \right\}.$$

Una propiedad de la función $s(i)$ es que sirve como indicador de la buena clasificación: en caso de ser negativa indica que el objeto se encuentra mal clasificado en un conglomerado que no le corresponde, si $s(i) = 0$ no se discierne de manera clara si pertenece a otro conglomerado y valores positivos implican buena clasificación a más cercanos a 1 mejor. El valor óptimo se obtiene maximizando $Silueta$ ya que esto implicaría un estructura con conglomerados muy compactos y separados entre sí.

- **Criterios de información bayesiano y de Akaike. CIB y CIA.** CIB nace en 1986 como una aproximación del factor de Bayes por su complejidad a la hora de calcularse [75]. Se usa para determinar tanto la complejidad del modelo como el número de conglomerados en modelos de mixturas gausianas como ya se ha visto. El criterio se define como

$$CIB = -2ln(L) + \nu ln(N).$$

Siendo N el número de elementos en la base de datos, L la verosimilitud de los parámetros que pretenden generar el modelo a estudiar y ν es el número de parámetros a estimar, es decir, libres.

En 1984 se propone una estimación del *criterio de información de Akaike* que si bien es parecida al anterior,

$$CIA = -2ln(L) + 2\nu,$$

se trata de una estimación de la densidad de la verosimilitud del modelo.

En ambos casos un mejor modelo será aquel que tenga un valor de CIA o CIB menor.

- **Medida de estabilidad.** En 2001 Ben-Hur et al. proponen un método basado en estabilidad [76] para hallar el número de conglomerados óptimo en Análisis de Conglomerados que usen un algoritmo jerárquico. Este método, y todos los basados en estabilidad, presenta ventajas respecto a los anteriores pues no necesitan asumir distribuciones sobre los datos o que los conglomerados posean una forma determinada. Además, se puede usar para determinar si un conjunto de datos realmente posee una estructura.

  El método planteado se basa en la consideración de una estructura estable como aquella que si se plantea con distintas muestras aleatorias de los datos (mayores al 50 % del total) los resultados son similares entre sí.

  Para entender el concepto dado por Ben-Hur et al. de similitud entre distintas muestras se necesitan de dos definiciones previas. Sea $\Xi \subset \mathbb{R}^p$ el conjunto de N datos divididos en G grupos disjuntos $C_1, C_2, \ldots, C_G$:

  Definición 5.1.12 (Etiquetado)**.** Un etiquetado L es una partición de Ξ en G subgrupos o conglomerados que se representará mediante la matriz C dada por

  $$C_{ij} = \begin{cases} 1 & \text{si } x_i \text{ y } x_j \text{ pertenecen al mismo conglomerado y } i \neq j \\ 0 & \text{en caso contrario} \end{cases}$$

  Definición 5.1.13 (Producto)**.** Sean L_1 y L_2 dos etiquetados con matrices C^1 y C^2. Se define el producto de ambas como

  $$\langle L_1, L_2 \rangle = \left\langle C^1, C^2 \right\rangle = \sum_{i,j} C^1_{ij} C^2_{ij}.$$

  En [76] se usa que el producto anterior cumple la desigualdad de Cauchy-Schwartz para definir la similitud

  Definición 5.1.14 (Similitud)**.** Sean L_1, L_2 dos etiquetados. Se define la similitud entre estos como

  $$cor(L_1, L_2) = \frac{\langle L_1, L_2 \rangle}{\sqrt{\langle L_1, L_2 \rangle \langle L_1, L_2 \rangle}}.$$

  El método seguido, para cada $k = 1, \ldots, G$ (distintos números de conglomerados) toma varias muestras aleatorias de entorno al 80 % del total de los datos y les calcula la similitud anterior dos a dos. Se entiende que a más cercana a 1 más similares son y por tanto la estructura será más estable. Para interpretar este método se puede hacer mediante una gráfica donde se representa las distribuciones de acumulación de las similitudes para los distintos valores de k. El óptimo se obtendrá como el último valor antes de pasar de distribuciones muy cercanas a 1 a otras más dispersas como se muestra en las figuras 3 y 6 en [76].

  Ben-Hur et al. proporcionan otra forma de elegir el óptimo más precisa basada en el mismo principio anterior. Además para elegir el número óptimo de conglomerados este método se puede usar para otros parámetros [77].

6

Aplicación a un caso real

El grupo *The World Bank* participa en la *Open Data Initiative* proporcionando acceso a extensas bases de datos sobre el desarrollo de los distintos países medidos de diversas maneras. De entre estas se hará uso de *World Development Indicators* [78] aunque no al completo al ser demasiado extensa.

Se trabajará, por tanto, con una base de datos de variables continuas. Siguiendo con la intención de este documento se compararán dos técnicas basadas en la forma, Dynamic Time Warping y Modelos Markovianos Ocultos, donde el algoritmo de conglomeración usado en cada técnica será el mismo. Esta comparación se hace con el único objetivo de ilustrar la toma de decisión entre dos técnicas diferentes para realizar un Análisis de Conglomerados. MMO se puede aplicar a la base de datos como se verá más adelante ya que las variables económicas se pueden ver como realizaciones de variables aleatorias.

Para realizar el Análisis de Conglomerados se hará uso del software R [79], un entorno y lenguaje de programación gratuito disponible desde CRAN en `https://cran.r-project.org/`. Este software es uno de lo más usados en la investigación y la minería de datos entre otras disciplinas. Además R se puede ampliar a través del propio software con los numerosos paquetes y librerías disponibles. En el caso que concierne se usarán diversas librería las cuales se irán nombrando a medida que se haga uso de ellas.

6.1 Implementación de software

La base de datos a usar está compuesta por una selección de 216 regiones y países medidos en 17 variables desde 1985 hasta 2021. Para cargar este y distintos archivos xlsx a lo largo de esta sección usaremos la librería `readxl` [80]. Además, dado que son bases de datos de series temporales multivariantes para su tratamiento se usará la librería `tidyverse` [81], pensada para la ciencia de datos, aúna varios paquetes compatibles entre ellos facilitando la lectura, modificación y visualización de los datos. A su vez, las anteriores son compatibles entre sí.

6.1.1 Preprocesado

Se procede a cargar la base de datos y obtener información de esta para darle un formato adecuado.

```
library(readxl)
library(tidyverse)

datos = read_excel("Data_Extract_From_World_Development_Indicators.xlsx",
```

```
                    col_types = c(rep("text",4), rep("numeric", 37)))

head(datos)
dim(datos)
```

```
# A tibble: 6 x 41
  `Country Name` Count~1 Serie~2 Serie~3 1985 ~4 1986 ~5 1987 ~6 1988 ~7 1989 ~8
  <chr>          <chr>   <chr>   <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 Afghanistan    AFG     Adoles~ SP.ADO~    159.    160.    161.    162.    162.
2 Afghanistan    AFG     Agricu~ NV.AGR~     NA      NA      NA      NA      NA
3 Afghanistan    AFG     CO2 em~ EN.ATM~     NA      NA      NA      NA      NA
4 Afghanistan    AFG     Domest~ FS.AST~     NA      NA      NA      NA      NA
5 Afghanistan    AFG     Energy~ EG.USE~     NA      NA      NA      NA      NA
6 Afghanistan    AFG     Export~ NE.EXP~     NA      NA      NA      NA      NA
# ... with 32 more variables: `1990 [YR1987]` <dbl>, `1991 [YR1987]` <dbl>,
#   `1992 [YR1987]` <dbl>, `1993 [YR1993]` <dbl>, `1994 [YR1994]` <dbl>,
#   `1995 [YR1995]` <dbl>, `1996 [YR1996]` <dbl>, `1997 [YR1997]` <dbl>,
#   `1998 [YR1998]` <dbl>, `1999 [YR1999]` <dbl>, `2000 [YR2000]` <dbl>,
#   `2001 [YR2001]` <dbl>, `2002 [YR2002]` <dbl>, `2003 [YR2003]` <dbl>,
#   `2004 [YR2004]` <dbl>, `2005 [YR2005]` <dbl>, `2006 [YR2006]` <dbl>,
#   `2007 [YR2007]` <dbl>, `2008 [YR2008]` <dbl>, `2009 [YR2009]` <dbl>, ...

> dim(datos)
[1] 3677   41
```

Se observa que la segunda y cuarta columna contiene información repetida por lo que se eliminarán. Comprobando las dimensiones de la base de datos podemos observar que no coincide con las esperadas ya que al tener 216 países medidas en 17 variables debería haber 3772 filas. Ante esto se extrae más información comprobando que las últimas filas no contienen datos, se eliminan las filas y columnas pertinentes y se cambia el nombre de las columnas para hacerlas manejables.

```
tail(datos)
datos = datos[,-c(2,4)]
datos = datos[-c(3673:3677),]
colnames(datos) <- c("País", "Variables", as.character(c(1985:2021)))
```

```
  # A tibble: 6 x 41
  `Country Name` Count~1 Serie~2 Serie~3 1985 ~4 1986 ~5 1987 ~6 1988 ~7 1989 ~8
  <chr>          <chr>   <chr>   <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 Zimbabwe       ZWE     Employ~ SL.AGR~      NA      NA      NA      NA      NA
2 NA             NA      NA      NA           NA      NA      NA      NA      NA
3 NA             NA      NA      NA           NA      NA      NA      NA      NA
4 NA             NA      NA      NA           NA      NA      NA      NA      NA
5 Data from dat~ NA      NA      NA           NA      NA      NA      NA      NA
6 Last Updated:~ NA      NA      NA           NA      NA      NA      NA      NA
# ... with 32 more variables: `1990 [YR1987]` <dbl>, `1991 [YR1987]` <dbl>,
#   `1992 [YR1987]` <dbl>, `1993 [YR1993]` <dbl>, `1994 [YR1994]` <dbl>,
#   `1995 [YR1995]` <dbl>, `1996 [YR1996]` <dbl>, `1997 [YR1997]` <dbl>,
#   `1998 [YR1998]` <dbl>, `1999 [YR1999]` <dbl>, `2000 [YR2000]` <dbl>,
#   `2001 [YR2001]` <dbl>, `2002 [YR2002]` <dbl>, `2003 [YR2003]` <dbl>,
#   `2004 [YR2004]` <dbl>, `2005 [YR2005]` <dbl>, `2006 [YR2006]` <dbl>,
#   `2007 [YR2007]` <dbl>, `2008 [YR2008]` <dbl>, `2009 [YR2009]` <dbl>, ...
```

Una vez con un formato adecuado y visto que pareciera presentar una gran cantidad de valores que no se encuentran, comprobamos si es así eligiendo algunas columnas de la base de datos.

```
summary(datos[,c(1:11)])
```

```
   Pais             Variables             1985              1986          
Length:3672        Length:3672        Min.   :  -17.02   Min.   :  -31.905  
Class :character   Class :character   1st Qu.:   10.66   1st Qu.:    9.913  
Mode  :character   Mode  :character   Median :   37.00   Median :   35.491  
                                      Mean   :  228.11   Mean   :  219.839  
                                      3rd Qu.:   71.23   3rd Qu.:   71.015  
                                      Max.   :15225.34   Max.   :15080.930  
                                      NA's   :2294       NA's   :2290       
     1987               1988               1989               1990          
Min.   :  -17.15   Min.   :  -13.38   Min.   :  -42.45   Min.   :  -20.860  
1st Qu.:   10.47   1st Qu.:   11.25   1st Qu.:   10.15   1st Qu.:    7.488  
Median :   34.82   Median :   35.26   Median :   35.07   Median :   30.899  
Mean   :  218.00   Mean   :  230.46   Mean   :  229.33   Mean   :  234.037  
3rd Qu.:   71.36   3rd Qu.:   70.70   3rd Qu.:   71.29   3rd Qu.:   69.458  
Max.   :14654.09   Max.   :14582.10   Max.   :14355.97   Max.   :13703.176  
NA's   :2264       NA's   :2268       NA's   :2256       NA's   :1914       
     1991               1992               1993          
Min.   :  -64.047   Min.   :  -44.900   Min.   :  -29.300  
1st Qu.:    7.462   1st Qu.:    7.154   1st Qu.:    7.083  
Median :   30.139   Median :   30.493   Median :   30.486  
Mean   :  200.578   Mean   :  197.960   Mean   :  202.080  
3rd Qu.:   68.012   3rd Qu.:   67.662   3rd Qu.:   67.633  
Max.   :14818.456   Max.   :15242.954   Max.   :15624.708  
NA's   :1717        NA's   :1679        NA's   :1676       
```

Se comprueba que el número de valores que faltan es muy elevado en todas las columnas. Ante esto se toma la decisión de ver aquellas variables y países a los que les falten demasiados valores haciendo que estimar esos valores mediante un proceso de imputación no sea representativo. Para ello se necesita procesar los datos como se muestra en la siguiente sección.

6.1.2 Limpieza de los datos

Al tratar con series temporales multivariantes se necesita modificar cada serie de manera individual por lo que se trabajará con una lista donde cada elemento es una serie transformándola de las maneras necesarias hasta obtener una lista de matrices de datos, y otra con sus traspuestas.

```
#### NOMBRE DE LAS VARIABLES ####

nombres_variables = rep(0, 17)
for (i in c(1:17)) {
  nombres_variables[i] <- datos[i,2]
}


#### NOMBRE DE LOS PAÍSES ####

v =  seq(1, 3672, by=17)

países = list()

for (i in v) {
  países = append(países, datos[i,1])
}
```

```
#### LISTA DE PAÍSES COMO TIBBLES ####

lista_países = vector("list", 216)

for (i in c(1:216) ) {
  valores <- datos[datos$País== países[i],]
  nombre <- as.character(países[i])
  lista_países[[i]] <- assign(nombre, valores)
}

#### PAÍSES COMO TIBBLES SIN SU NOMBRE ####


elim_nombre <- function(var1) {
  var1[,c(2:39)]

}

lista_países = lapply(lista_países, elim_nombre)

#### TRASPUESTAS DE LAS TIBBLES ####


trasponer <- function(var1) {
  var1 %>% pivot_longer(cols= -1) %>% pivot_wider(names_from = "Variables", values_from
     = "value") %>% rename( Year = name)
}

lista_países.tras = lapply(lista_países, trasponer)


#### DATOS COMO MATRICES ####

lista_países.matriz = lapply(lista_países, data.matrix)
lista_países.tras.matriz = lapply(lista_países.tras, data.matrix)
```

Una vez obtenidas las listas precisas se hará uso de la librería `imputeTS` [82] dedicada a las series temporales y a la detección, visualización e imputación de los valores que faltan en estas. En este caso solo se usará la función `statsNA()` que se aplica a series temporales univariantes obteniéndose una lista con información sobre los valores que faltan de la cual se extraerá el número de estos valores por cada serie temporal univariante (variable) que compone a cada serie temporal multivariante. Si el total en una serie temporal univariante sobrepasa 29 se considerará que esa variable (en el caso de ese país) se deberá eliminar. Con los resultados se establece el número total de países en los que se debería eliminar cada variable en base a lo que se decidirá que si pasa de los 40 países no se usará esa variable en el Análisis de Conglomerados.

```
#### LISTA DE VARIABLES A ELIMINAR EN CADA PAÍS ####

var_elim_países = vector("list", 216)

for (i in c(1:216)) {
  lista = list()
  for (j in c(1:17)) {
    país <- lista_países.matriz[[i]]
    valor <- statsNA(país[j,], print_only = FALSE)[2]
    if(valor > 29) {
      lista = append(lista, j)
    } else {
      lista = append(lista, 0)
    }
    var_elim_países[[i]] <- lista
  }
```

```
}

#### NUMERO DE VARIABLES A ELIMINAR EN CADA PAÍS ####

vectvar_elim <- rep(0, 17)

for (i in c(1:216)) {

  lista <-var_elim_países[[i]]
  names(lista) <- c(1:17)

  for (j in c(1:17)) {
    if(j %in% lista) {

      vectvar_elim[j] <- vectvar_elim[j] +1
    }
  }
}

#Se guarda como tibble.

VarElim <- tibble(Variables = as.character(nombres_variables), Países = vectvar_elim)


#### GRAFICA DE BARRAS PARA VISUALIZAR LAS VARIABLES A ELIMINAR ####

hexpalette <- c("#115f9a", "#1984c5", "#22a7f0", "#48b5c4", "#76c68f", "#a6d75b", "#
    c9e52f", "#d0ee11",
                "#d0f400", "#d0ee11", "#c9e52f", "#a6d75b", "#76c68f", "#48b5c4", "#22
    a7f0", "#1984c5", "#115f9a")

ggplot(data=VarElim, aes(x = Variables, y = Países, fill = Variables)) +
  geom_bar(stat="identity") + theme(axis.text.x=element_blank(), axis.ticks.x=element_
    blank())+
  scale_fill_manual( values = hexpalette) + geom_hline(yintercept = 40) +
  ylab("Número de países")
```

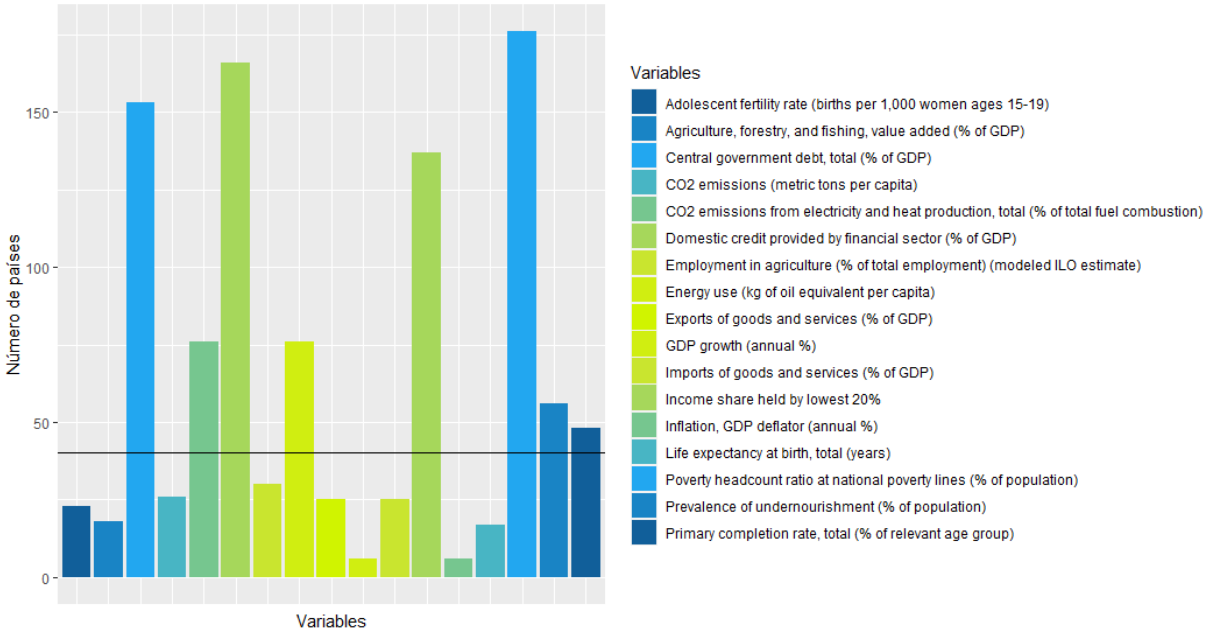


Figura 6.1: Gráfica de barras donde se representa el número de países en los que se ha decidido eliminar esa variable. La atraviesa una recta marcando el límite de 40 países.

Mediante la gráfica 6.1 se puede observar que eliminar ciertas variables es necesario ya que a pesar de no ser muy estricta la restricción para descartar una variable hay algunas como *Central government debt* o *Poverty headcount ratio at national poverty lines* que se eliminan en la mayoría de países. Se ha elegido el límite en 40 países al darse una clara división a esa altura.

Una vez visto lo anterior se procede a suprimir las variables precisas de la base de datos por lo que se hace necesario volver a obtener las listas de países como matrices (series temporales multivariantes).

```
#### SE DETERMINAN LAS VARIABLES A ELIMINAR ####


indices = c()

for (i in c(1:17)) {
  if(vectvar_elim[i] > 39) {
    indices = append(indices, i)
  }
}

nombresvar_eliminar = nombres_variables[indices]


#### SE ELIMINAN LAS VARIABLES DE LA BASE DE DATOS ####

n = length(nombresvar_eliminar)
for (i in c(1:n)) {
  datos = datos[!datos$Variables == nombresvar_eliminar[i],]
}

#### NOMBRE DE LAS VARIABLES QUE SE CONSERVAN ####

nombres_variables2 = c()
for (i in c(1:17-n)) {
  nombres_variables2[i] <- datos[i,2]
}

#### LISTA DE PAÍSES COMO TIBBLES ####

lista_países2 = vector("list", 216)

for (i in c(1:216) ) {
  valores <- datos[datos$País== países[i],]
  nombre <- as.character(países[i])
  lista_países2[[i]] <- assign(nombre, valores)
}

#### PAÍSES COMO TIBBLES SIN SU NOMBRE ####

lista_países2 = lapply(lista_países2, elim_nombre)

#### TRASPUESTAS DE LAS TIBBLES ####

lista_países2.tras = lapply(lista_países2, trasponer)

#### PAÍSES COMO MATRICES ####

lista_países2.matriz = lapply(lista_países2, data.matrix)

lista_países2.tras.matriz = lapply(lista_países2.tras, data.matrix)
```

Con la nueva base de datos se pretende eliminar ahora los países a los que le falten demasiados datos siendo esta vez más estrictos con un máximo de 19 valores.

```
#### SE ELIMINAN LOS PAÍSES DE LOS QUE FALTAN DEMASIADOS DATOS ####

orden_país_eliminado = c()
```

```
for (i in c(1:216)) {

  for (j in c(1:9)) {
    país <- lista_países2.matriz[[i]]
    valor <- statsNA(país[j,], print_only = FALSE)

    if( valor[2] >19) {
      orden_país_eliminado = append(orden_país_eliminado, i)
      break
    }
  }
}


nombres_países_eliminar = países[orden_país_eliminado];nombres_países_eliminar

m = length(nombres_países_eliminar)
for (i in c(1:m)) {
  datos = datos[!datos$País == nombres_países_eliminar[i],]
}
```

Una vez eliminados estos países la base de datos resultante contiene a 9 variables y 155 países. Con el objetivo de reducir el número de valores que faltan (detonados `NA`), se observa la presencia de estos y se suprimen los años con un número demasiado elevado. Teniendo en cuenta que la base de datos dispone ahora de 1395 filas, se descartarán los años a los que le faltan más de 300 valores. Además, se procede a hacer una comprobación sobre la base de datos para determinar que no existen filas vacías.

```
#### DETECCIÓN Y SUPRESIÓN DE LOS AÑOS CON GRAN PRESENCIA DE NA ####

valores_faltan = summary(datos[,c(3:39)])[7,]; valores_faltan

datos2 = datos[, -c(c(3:9), c(38,39))]

contador = 0
for (i in c(1:1395)) {
  serie = data.matrix(datos2[i,])[,-c(1,2)]
  valorNA = statsNA(serie, print_only = FALSE)[2]
  if (valorNA == 30) {
    contador = contador +1
    print(i)
  }
}
print(contador)

library(writexl)

write_xlsx(datos2, path =  "DatosProcesados.xlsx")
```

```
   > valores_faltan = summary(datos[,c(3:39)])[7,]; valores_faltan
          1985            1986            1987            1988            1989
"NA's   :523  " "NA's   :518  " "NA's   :510  " "NA's   :502  " "NA's   :487  "
          1990            1991            1992            1993            1994
"NA's   :304  " "NA's   :124  " "NA's   :116  " "NA's   :104  "  "NA's   :88  "
          1995            1996            1997            1998            1999
 "NA's   :49  "  "NA's   :32  "  "NA's   :27  "  "NA's   :26  "  "NA's   :25  "
          2000            2001            2002            2003            2004
 "NA's   :12  "   "NA's   :5  "   "NA's   :1  "   "NA's   :1  "   "NA's   :1  "
          2005            2006            2007            2008            2009
  "NA's   :4  "   "NA's   :4  "   "NA's   :4  "   "NA's   :4  "   "NA's   :4  "
          2010            2011            2012            2013            2014
  "NA's   :5  "   "NA's   :5  "   "NA's   :9  "   "NA's   :9  "   "NA's   :9  "
```

```
          2015            2016            2017            2018            2019
 "NA's   :16  "  "NA's   :16  "  "NA's   :16  "  "NA's   :16  "  "NA's   :24  "
          2020            2021
"NA's   :352  " "NA's   :754  "
> print(contador)
[1] 0
```

Ya considerada la base de datos preparada se guarda como archivo .xlsx haciendo uso del paquete `writexl` [83] para usar este archivo en el siguiente apartado de imputación.

6.1.3 Imputación

El método elegido hace uso de imputación mediante Bosques Aleatorios pues al partir sólo de la base de datos se supondrá que los mecanismos que propician los datos que faltan caen bajo la clasificación de *Missing Completely at Random, MCAR* dada en [84]. Esta categoría implica que los datos que no se encuentran son un subconjunto aleatorio del total. Es decir, el mecanismo que provoca la ausencia de datos es independiente de los valores (tanto observados como no) de la base de datos. Por tanto, encaja en el ejemplo que se trabaja.

Los Bosques Aleatorios entran dentro del Aprendizaje Supervisado donde se busca hacer predicciones partiendo de datos etiquetados. Se entiende la etiqueta como la relación entre una variable respuesta y las variables explicativas. En el caso expuesto se entienden los nombres de los países como la variable respuesta y las variables que se miden de cada uno de ellos como las explicativas. De este modo, los datos se pueden ver como etiquetados según los nombres de los países. En términos ya vistos, cada país sería un conglomerado. Se entenderá un *predictor* como una función que tomando los datos y parámetros a determinar produce una predicción de la variable respuesta.

Son Leo Breiman y Adele Cutler los que proponen este nuevo algoritmo de aprendizaje *Ensemble*. Este tipo de algoritmos se caracterizan por usar múltiples métodos más sencillos y combinar sus predicciones para obtener mejores resultados. En este caso se hace uso de *Bootstrap Aggregating* más conocido como bagging propuesto en 1994 por el mismo Leo Breiman. Bagging consiste en tomar K muestras aleatorias con reemplazamiento de tamaño el de la muestra (esto se conoce como *bootstrap*), aplicar el predictor sobre cada una de ellas y finalmente combinar los K predictores en uno único. Los predictores usados en los Bosques Aleatorios son los llamados *árboles de decisión*, árboles binarios (de cada vértice salen dos aristas) donde cada vértice o nodo contiene información sobre las variables explicativas que se usa para tomar una decisión. En el ejemplo que se trabaja los nodos se conocen como umbrales al ser las variables explicativas continuas. Finalmente, el algoritmo de Bosques Aleatorios consiste en construir K árboles de decisión mediante bagging. Cada uno de ellos usa un subconjunto aleatorio de las variables explicativas para realizar las clasificaciones, siendo la clasificación final elegida por voto mayoritario

Este algoritmo se puede usar tanto para clasificación como para regresión y su uso para la imputación se puede ver en [85, 86]. Los Bosques Aleatorios presentan también la gran ventaja de permitir una elevada complejidad sin sobreajuste. Es por esto, que se ha decidido usar este algoritmo en el ejemplo que se presenta.

La librería que se usará es `randomForest` [87] en concreto la función `rfImpute()` donde, como se ha indicado anteriormente, la variable respuesta serán los países. Así, la función anterior procede con una primera aproximación sustituyendo todos los valores que faltan por las medias de la columnas siendo cada columna en este caso un vector correspondiendo a una de las variables a lo largo de todos los países. Tras esto se procede a construir un Bosque Aleatorio y sustituir los valores imputados por las medias ponderadas de los valores que sí estaban observados, repitiéndose este proceso de construcción del bosque hasta 6 veces o las indicadas. La ponderación se basa en la matriz de proximidad del Bosque Aleatorio

donde si al recorrer un árbol las observaciones terminan en el mismo nodo la proximidad se aumenta en 1, normalizando después dividiendo entre el número total de árboles.

Se procede por tanto cargando la base de datos DatosProcesados.

```
datos <- read_excel("DatosProcesados.xlsx",
         col_types = c(rep("text",2), rep("numeric", 28)))
```

Se procede con el procesamiento ya visto anteriormente para poder manejar la base de datos de la manera adecuada.

```
##### NOMBRE DE LAS VARIABLES ####

nombres_variables = rep(0, 9)
for (i in c(1:9)) {
  nombres_variables[i] <- datos[i,2]
}

#### NOMBRE DE LOS PAÍSES ####

v =  seq(1, 1395, by=9)

paises = list()

for (i in v) {
  paises = append(países, datos[i,1])
}

#### LISTA DE PAÍSES COMO TIBBLES ####

lista_países = vector("list", 155)

for (i in c(1:155) ) {
  valores <- data[data$País== países[i],]
  nombre <- as.character(paises[i])
  lista_países[[i]] <- assign(nombre, valores)
}

#### PAÍSES COMO TIBBLES SIN SU NOMBRE ####


elim_nombre <- function(var1) {
  var1[,c(2:30)]

}

lista_países = lapply(lista_países, elim_nombre)

#### TRASPUESTAS DE LAS TIBBLES ####


trasponer <- function(var1) {
  var1 %>% pivot_longer(cols= -1) %>% pivot_wider(names_from = "Variables", values_from
     = "value") %>% rename( Year = name)
}

lista_países.tras = lapply(lista_países, trasponer)


#### DATOS COMO MATRICES ####

lista_países.matriz = lapply(lista_países, data.matrix)

lista_países.tras.matriz = lapply(lista_países.tras, data.matrix)
```

Como se ha comentado `rfImpute()` necesita que la base de datos esté dispuesta de manera que las columnas sean las variables y las filas los distintos países. Para ello se procede a eliminar los años ya que no es una variable explicativa y usar la lista de países traspuestos junto un bucle para suprimir los años de cada país. Tras esto se vuelve a aunar los países en una nueva base de datos la cual usaremos para implementar los Bosques Aleatorios.

```
library(randomForest)

#### ELIMINAR LOS AÑOS DE LOS PAÍSES ####

lista_paÍses_noaños = vector("list", 155)

for (i in c(1:155)) {

  v = rep(países[[i]], 28)
  nombre = as_tibble_col(v, column_name = "Name")
  lista_países_noaños[[i]] = cbind(nombre, lista_países.tras[[i]][,-1])
}

#### PREPARAR LA BASE DE DATOS ACOMODADA A RFIMPUTE ####

datos_bosques = data.table::rbindlist(lista_países_noaños)
attach(datos_bosques)

#Para rfImpute se necesita que la variable respuesta sea un factor

datos_bosques$Name = as.factor(Name)

#### IMPUTACIÓN DE LOS DATOS ####

set.seed(17)
datos.imputados <- rfImpute(Name ~ ., datos_bosques)


write_xlsx(datos.imputados, path =  "DatosImputados.xlsx")

detach(datos_bosques)
```

Se obtiene por tanto que la base de datos sobre la que se trabajará es DatosImputados. En las siguientes secciones se verán tanto DTW como MMO aplicados a la base de datos anterior para realizar Análisis de Conglomerados.

6.1.4 Dynamic Time Warping

En esta sección es importante destacar que se busca ejemplificar la teoría hasta ahora vista por lo que no se centrará tanto en el resultado del Análisis de Conglomerados como en la técnica, el algoritmo y los índices usados.

Una vez se ha obtenido la base de datos que se usará en este y el siguiente ejemplo de aplicación de una técnica a un caso real es oportuno dar una descripción más detallada de esta. La base de datos está compuesta por 155 países : Alemania, Algeria, Australia, Albania, Arabia Saudita, Angola, Argentina, Austria, Las Bahamas, Bangladesh, Azerbaijan, Bahrain, Barbados, Bélgica, Belarús, Belice, Benin, Bhutan, Bosnia y Herzegovina, Bolivia, Botswana, Brasil, Brunei Darussalam, Bulgaria, Burundi, Camboya Canada, Chile, China, Chad, Camerún, Cabo Verde, Burkina Faso, Colombia, República Democrática del Congo, Costa Rica, Croatia, República Checa, Dinamarca, Chipre, Cuba, Costa de Marfil, República del Congo, Comoras, República Dominicana, República Árabe de Egipto, Estonia, Eslovaquia, Eslovenia, España, Estados Unidos, Fiji, Finlandia, Reino de Eswatini, Eritrea, Ecuador, El Salvador, Francia, Gabón, Georgia, Ghana, Grecia, Guatemala, Guinea, Gambia, Guinea-Bissau, Guyana, Haití, Honduras, Hungría, Islandia, India, Indonesia, Irán, Iraq, Irlanda, Italia, Jamaica, Japón, Jordania, Kazajistán, Kenia,

Korea, Kuwait, Kirguistán, Laos, Letonia, Líbano, Líbia, Lituania, Luxemburgo, Madagascar, Macedonia del Norte, Malasia, Mali, Malta, Mauritania, Mauricio, México, Moldavia, Mongolia, Montenegro, Marruecos, Mozambique, Namibia, Nepal, Nueva Zelanda, Nicaragua, Nigeria, Noruega, Omán, Países Bajos, Pakistán, Panamá, Paraguay, Perú, Filipinas, Polonia, Portugal, Qatar, Reino Unido, Rumanía, Rusia, Ruanda, Samoa, Senegal, Serbia, Sierra Leona, Singapur, Sudáfrica, Sri Lanka, Sudán, Suecia, Suiza, Siria, Tanzania, Tailandia, Tayikistán, Timor Oriental, Togo, Tonga, Túnez, Turquía, Turkmenistán, Uganda, Ucrania, Emiratos Árabes Unidos, Uruguay, Uzbekistán, Vanuatu, Venezuela, Zambia y Zimbabue. Sobre estos países se miden 9 variables desde 1992 hasta 2019:

1. Tasa de fertilidad adolescente tomado como los nacimientos por cada mil mujeres de entre 15 y 19 años.
2. Agricultura, silvicultura, y pesca tomado como el valor añadido al tanto por ciento del PIB.
3. Emisiones de CO_2 medida en toneladas métrica per cápita.
4. Exportación de bienes y servicios visto como el tanto por ciento del PIB.
5. Crecimiento del PIB medido como el tanto por ciento anual.
6. Importación de bienes y servicios tanto por ciento del PIB.
7. Inflación, deflactor del PIB medido como el tanto por ciento anual.
8. Esperanza de vida al nacer en años.
9. Empleo en agricultura como el tanto por ciento del empleo.

En esta sección, siguiendo el procedimiento descrito en 2.2.2, y ya habiendo seleccionado las variables significativas se usará la librería `IncDTW` [88] para implementar la técnica elegida mediante la función `dtw_dismat()` de la cual se obtendrá la matriz de distancia entre los países a la que se le aplicará un algoritmo de conglomeración jerárquico. La función permite ajustar parámetros como `ws` para ajustar el tamaño de la ventana de Sakoe-Chiba o `step_pattern` para la restricción sobre los pesos asociados a los pasos. En este ejemplo se usará el segundo tipo de patrón simétrico que se vio en 4.1.1 que no propicia los pasos diagonales, denotado `symmetric2` como argumento de `step_pattern`. Además la distancia d_p se tomará con $p = 2$, la euclídea.

Se verá a continuación el proceso desde la lectura de los datos hasta obtener la matriz de distancia entre los países.

```
datos <- read_excel("DatosImputados.xlsx", col_types = c("text", rep("numeric", 9)))
attach(datos)

#### NOMBRE DE LOS PAÍSES ####

v =  seq(1, 4340, by=28)

países = list()

for (i in v) {
  países = append(países, data[i,1])
}

#### LISTA DE PAÍSES COMO TIBBLES ####

lista_países = vector("list", 155)

for (i in c(1:155) ) {
  valores <- data[data$Name == paises[i],]
  nombre <- as.character(paises[i])
  lista_paises[[i]] <- assign(nombre, valores)
```

```
}

#### ACOMODAR LAS TIBBLES PARA SU USO EN DTW_DISMAT  ####


for (i in c(1:155)) {

  v = c(1992:2019)
  años = as_tibble_col(v, column_name = "Year")
  lista_países[[i]] = cbind(años, lista_países[[i]][,-1])
}


lista_países.matriz = lapply(lista_países, as.matrix)

elim_nombre_var <- function(var1) {
  var1[,-1]
}

lista_países.matriz = lapply(lista_países.matriz, elim_nombre_var)


#### OBTENCIÓN DE LA MATRIZ DE DISTANCIA USANDO DTW ####

library(IncDTW)

DTW <-dtw_dismat(lot = lista_paises.matrix2,dist_method = "norm2", step_pattern = "
    symmetric2", return_matrix = FALSE)

matrizDTW_dist = DTW$dismat  #como matriz de distancia en formato dist
```

Ya obtenida la matriz de distancia se usará la función `hclust()` que le aplica un algoritmo de conglomeración jerárquico aglomerativo a una matriz de distancias. El tipo de algoritmo que se aplicará tanto en este caso como para la matriz que produzca los MMOs se conoce como método de unión completa donde se define la distancia entre conglomerados como

$$d(C_i, C_j) = \text{máx}\{d_{DTW}(X, Y) | X \in C_i, Y \in C_j\}.$$

En este caso el algoritmo comienza considerando cada país un conglomerado y va sucesivamente uniendo los objetos o conglomerados más cercanos según la distancia anterior hasta formar uno único. Para visualizar el resultado de aplicar este algoritmo se verá su *dendrograma*, un árbol binario donde cada nodo representa las uniones producidas y la altura a la que se sitúa, la distancia entre los conglomerados.

```
congloDTW <- hclust(matrizDTW_dist, method = "complete")

plot(congloDTW, labels = as.character(c(1:155)),  hang = 0.1, axes = TRUE,
     frame.plot = FALSE, sub = '', main = "Dendrograma", xlab = 'Unión completa',
     ylab = 'Distancia', col = 'blue')
```

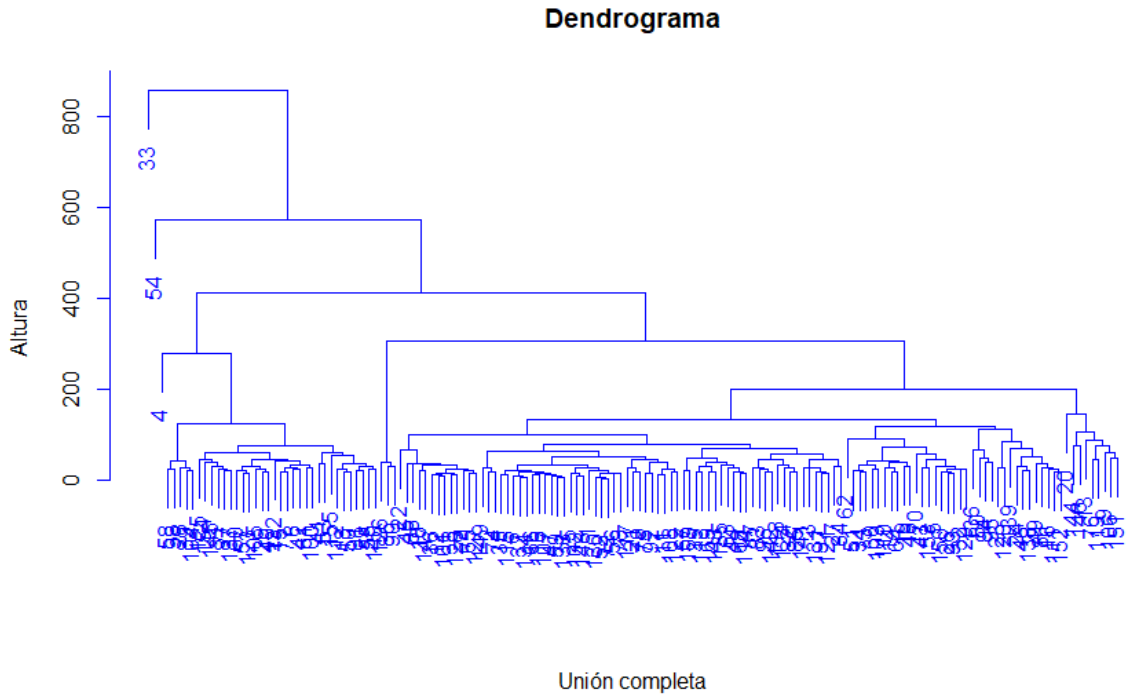


Figura 6.2: Dendrograma mostrando los conglomerados al aplicar DTW y un algoritmo jerárquico aglomerativo de unión completa sobre la base de datos.

En la figura anterior se han sustituido los nombres de los países por su correspondiente orden en la lista de países para permitir una visualización más clara. Aquí los elementos que hay bajo cada nodo representan un conglomerado. De este modo se observa que los países 4, 54 y 33 (Angola, Georgia y La República Democrática del Congo respectivamente) se unen a una mayor altura denotando una distancia mayor con el resto ya que si, por ejemplo, se dividiera en dos conglomerados uno lo formaría 154 países y el otro La República Democrática del Congo. Esto puede ser debido a que estos países sean valores atípicos dentro de la base de datos o bien a una gran falta de valores que no se hayan imputado de la manera adecuada. Al comprobar esto se observa que estos países apenas presentan falta de valores por lo que esto puede ser debido a ser países atípicos. La detección de este tipo de valores en series temporales multivariantes es una tarea extensa que escapa a este documento por lo que no se considerará. Por tanto, se seguirán considerando dentro de los objetos de estudio.

Una vez obtenidos los resultados sobre los distintos conglomerados al aplicar DTW se ha de decidir el número óptimo de grupos en los que dividir los datos. Para ello, como ya se ha visto, se usarán los índices de validación. En este ejemplo y el siguiente se usarán los índices de Dunn, silueta y Davies-Bouldin. Se recuerda que se busca maximizar los 2 primeros y minimizar el último. Usando las respectivas librerías `clValid`, `cluster` y `clv` [89, 90, 91] se calcularán para un total de entre 2 y 10 conglomerados. Se han elegido estos tres índices ya que sólo requieren de la matriz de distancias pues como se verá en el otro ejemplo, se modificará la base de datos.

El procedimiento general seguido en la implementación es realizar un bucle y hacer uso de la función `cutree(tree, k)` que asocia a cada país el conglomerado correspondiente de entre los `k` en los que divide el dendrograma 6.2.

- Para el índice de Dunn basta eso y la matriz de distancias para usar la función `dunn()` ya que como se vio solo hace uso de la distancia entre elementos para su cálculo.

- El índice de silueta se calcula extrayendo el valor `avg.width` de `summary(silhouette)` obteniéndose la media de las siluetas individuales de los distintos países.
- Para el índice DB, en este se hace uso de `clv.Davies.Bouldin()` que toma la medida de dispersión, s, como el diámetro de los conglomerados y la distancia entre conglomerados como la del método de unión completa a través de los argumentos `intracls` e `intercls`, tomando la medida de similitud entre los conglomerados como

$$R_{ij} = \frac{diam(C_i) + diam(C_j)}{d(C_i, C_j)}.$$

  La función `cls.scatt.diss.mx()` se usa para proporcionar la información necesaria sobre los conglomerados y las distancias intra e inter conglomerado para calcular el índice.

```
#### ÍNDICE DE DUNN ####
library(clValid)

#Se necesita la matriz de distancia en formato matrix
matrizDTW_matriz = as.matrix(matrizDTW_dist)


ind_dunn = c()
for (i in c(2:10)) {
  cluster <- cutree(congloDTW, i)
  ind_dunn <- append(ind_dunn, dunn(matrizDTW_matriz, cluster))
}

#### ÍNDICE DE SILUETA ####
library(cluster)

ind_sil = c()
for (i in c(2:10)) {
  cluster <- cutree(congloDTW, i)
  silueta <- summary(silhouette(cluster, matrizDTW_matriz))$avg.width
  ind_sil <- append(ind_sil, silueta)
}

#### ÍNDICE DB ####

library(clv)

intraclust = c("complete")
interclust = c("complete")
ind_db = c()
for (i in c(2:10)) {
  cluster <- cutree(congloDTW, i)
  info <- cls.scatt.diss.mx(matrizDTW_matriz, cluster)
  index <-clv.Davies.Bouldin(info, intracls = intraclust, intercls = interclust)
  ind_db <- append(ind_db, index)
}

#Se redondean los valores a tres cifras decimales
ind_dunn = round(ind_dunn, digits = 3)
ind_sil = round(ind_sil, digits = 3)
ind_db = round(ind_db, digits = 3)

Índices <- matrix(c((c(2:10)), ind_dunn, ind_sil, ind_db), ncol = 4)
colnames(Índices) <- c('Número de conglomerados', 'Índice de Dunn', 'Índice de Silueta'
    , 'Índice DB')
Índices <- as.table(Índices)
```

Se muestra a continuación la tabla anteriormente calculada con los valores obtenidos para los tres índices.

Índices de evaluación de DTW			
Número de conglomerados	Dunn	Silueta	Davies-Bouldin
2	**1,126**	**0,876**	0,669
3	0,889	0,812	**0,640**
4	0,055	0,331	0,933
5	0,061	0,325	0,879
6	0,085	0,309	0,705
7	0,115	0,338	0,832
8	0,126	0,333	0,751
9	0,136	0,262	0,871
10	0,144	0,241	0,880

Tabla 6.1: Tabla que muestra los distintos valores de los índices de Dunn, silueta y DB bajo distintos números de conglomerados, Destacan los valores óptimos

La elección del número óptimo de conglomerados se hace por voto mayoritario siendo por tanto la división resultante de este Análisis de conglomerados en dos grupos uno con La República Democrática del Congo y el otro con el resto.

Se comprueba que si se hace escalamiento multidimensional sobre la matriz de datos para obtener una representación en el espacio de los países Angola, Georgia y La República Democrática del Congo (4, 54 y 33) respectivamente se encuentran muy alejados del total de los países. Se decide entonces eliminar estos países con el objetivo de recalcular los índices y comprobar que partición de los datos se nos presenta más allá de la anterior.

```
library("scatterplot3d")

#Se calculan los puntos en el espacio relacionados a los países

emdDtw <- cmdscale(matrizDTW_dist, k=3)

#Se acomoda a un data.frame para la función scatterplot3d
x <- emdDtw[,1]
y <- emdDtw[,2]
z <- emdDtw[,3]

EMD <- cbind(c(1:155), x, y, z)
colnames(EMD) <- c('países', 'x', 'y', 'z')

# Representación gráfica
grafica <- scatterplot3d(EMD[,-1], angle = 35, box = FALSE, xlim = c(-150, 700), ylim =
    c(-150,350), zlim = c(-185, 250))
text(grafica$xyz.convert(EMD[,-1]), labels = EMD[,1], cex= 0.7, col = 'steelblue') #se
    usa xyz.convert para pasar los puntos del espacio al plano para adjuntar las
    etiquetas
```

Se ha hecho uso del paquete `scatterplot3d` [92] para esta representación.

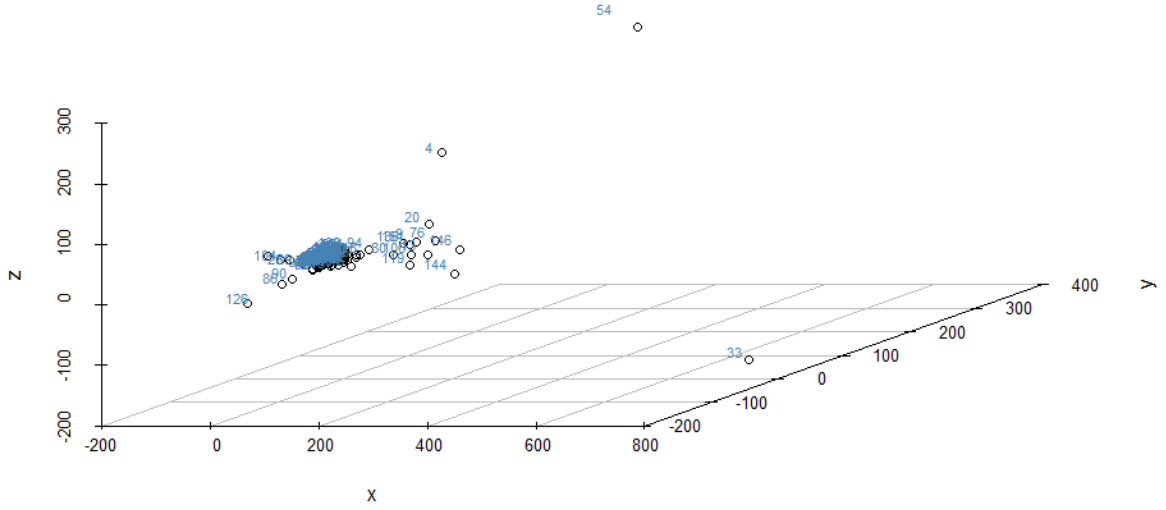


Figura 6.3: Representación en el espacio de las posiciones relativas de los distintos países. Se observan los países 4, 33 y 54 muy alejados del resto.

Una vez vista se procede a eliminar estos datos, volver a aplicar el algoritmo y recalcular los índices.

```
matrizDTW_matriz = matrizDTW_matriz[-c(4,54,33), -c(4,54,33)] #formato de matriz
matrizDTW_dist = as.dist(matrizDTW_matriz) #formato dist

#Se realiza el mismo análisis de conglomerados
congloDTW <- hclust(mat_dist, method = "complete")

#Veamos el dendrograma asociado
plot(congloDTW, labels = as.character(c(1:152)),  hang = 0.1, axes = TRUE, frame.plot =
     FALSE, sub = '', main = "Dendrograma", xlab = 'Unión completa', ylab = "Altura",
    col = 'blue')
```

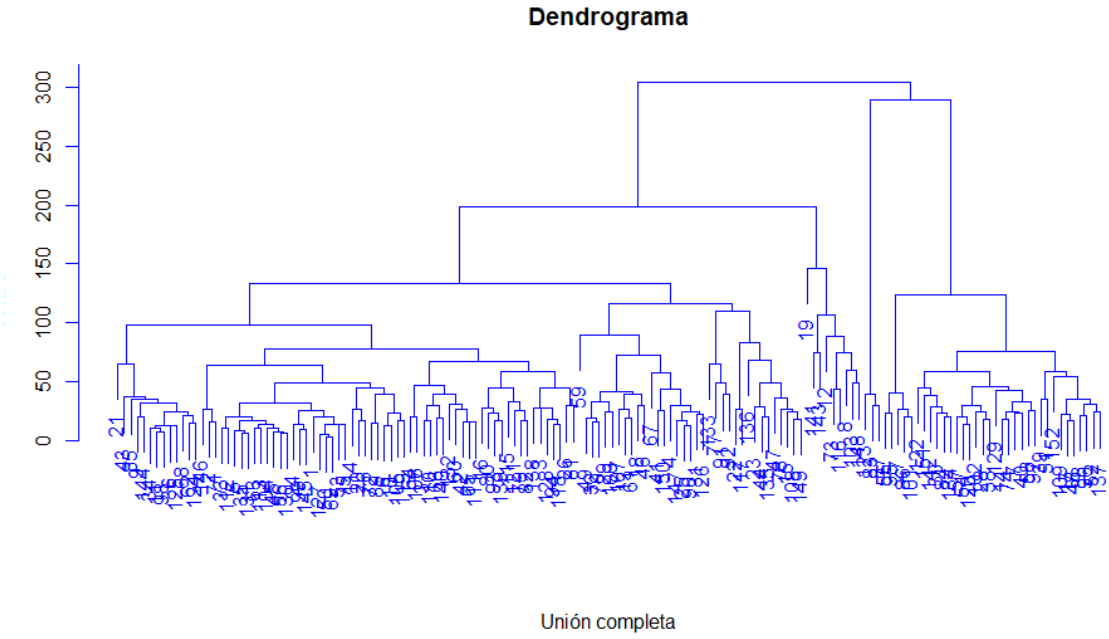


Figura 6.4: Dendrograma obtenido al eliminar Angola, Georgia y La República Democrática del Congo y volviendo a implementar el algoritmo de conglomeración.

Índices de evaluación de DTW			
Número de conglomerados	Dunn	Silueta	Davies-Bouldin
2	0,058	0.318	1.605
3	0,085	0,315	**1,000**
4	0,115	**0,344**	1,195
5	0,126	0,340	1,024
6	0,136	0,267	1,161
7	0,144	0,246	1,145
8	0,153	0,255	1,227
9	0,158	0,253	1,230
10	**0,171**	0,255	1,298

Tabla 6.2: Tabla que muestra los distintos valores de los índices de Dunn, silueta y DB tras eliminar los países 4, 33 y 54. Destacan los valores óptimos.

Es evidente la falta de consenso por parte de los índices sobre el número de conglomerados óptimos aunque se considerará cuatro como óptimo ya que es el que indica el índice de silueta y se encuentra cerca del número indicado por el índice DB. Así se tendría una división en cuatro conglomerados que se podría visualizar haciendo uso de la librería `factoextra` [93].

```
library(factoextra)

fviz_dend(congloDTW, k = 4,  ylab = '', k_colors = c('#34df2e', '#00AFBB', '#1050EF', '
    #F2B99B'), color_labels_by_k = TRUE, ggtheme = theme_classic(), main = "División en
     4 grupos",type = 'rectangle')
```

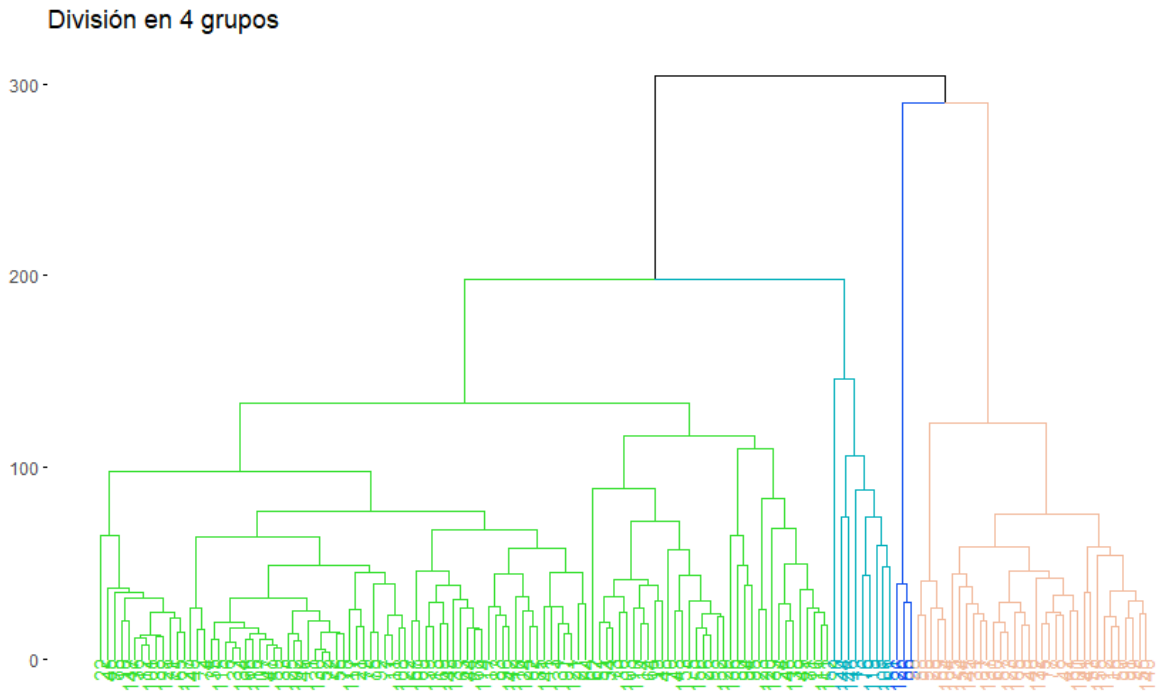


Figura 6.5: Dendrograma mostrando la división en cuatro conglomerados.

Aunque la división es mucho más significativa se comprueba que los índices óptimos son peores que antes de eliminar los países.

Se ha realizado por tanto la aplicación de una técnica de conglomerado, en concreto la conjunción de DTW y algoritmo jerárquico aglomerativo según los pasos indicados. Se han determinado las variables relevantes al estudio, aplicado DTW para obtener la matriz de distancias, sobre esta se ha aplicado un algoritmo de conglomeración y finalmente los datos han sido divididos en 2 y 4 grupos respectivamente de acuerdo a los valores de las medidas de validación usadas.

6.1.5 Modelos Markovianos Ocultos

En esta sección se definirán MMO sobre los distintos países. Es importante notar que ajustar un Modelo Markoviano Oculto a una serie temporal multivariante es una tarea complicada que se ve muy beneficiada de algún conocimiento a priori sobre los datos con los que se trata. En el caso que concierne a este proyecto se tomará como referencia la vía de acción usual para la detección de regímenes de mercado.

Siguiendo con la intención de ejemplificar la teoría expuesta se ha elegido hacer uso de los MMO en esta sección ya que entra dentro del mismo tipo de técnica anterior siendo ambas basadas en la forma de las series temporales. Así, se ha decidido tomar una técnica sencilla, que solo necesita que los datos sean variables continuas pero que no proporciona mucha capacidad de adaptación e interpretación sobre los datos más allá de sujetarlo a ciertas restricciones que aceleran los cálculos como es el DTW y compararla con MMO que proporciona amplias capacidades de adaptación e interpretabilidad. El objetivo de la comparación (en tanto que sobre las medidas de evaluación) que se verá tras dar este ejemplo es el de ejemplificar que una correcta elección de la técnica de conglomerados es esencial y para ello, como se ha

visto, se hace necesario conocer distintas técnicas, sus limitaciones, ventajas y desventajas.

Los MMO que se usarán caen dentro de lo que se ha denotado MMOC ya que las variables son continuas. El primer problema que presenta la implementación de MMO es la elección del número de estados. Esta información puede ser conocida a priori o en su defecto deberá determinarse una cantidad de estados que se suponga adecuada a la naturaleza de los datos, construir los distintos modelos con distinto número de estados y usar o bien CIB, CIA o ambos para determinar el mejor modelo de entre estos y con ello elegir el número óptimo de estados. En el ejemplo que se expone no se posee conocimientos previos sobre los datos por lo que se haría necesario seguir los pasos de la segunda opción. Si bien esto sería lo ideal no se hará en este proyecto ya que se debería hacer 155 veces (una por país) sobre unos datos que requieren de la estimación manual de ciertos parámetros para realizarlo. Por ello se ha decidido tomar como ejemplo la vía de acción de la detección de regímenes de mercado donde, de manera usual, se suponen dos estados (y suelen ser los óptimos cuando se hace el proceso que aquí se omite). Esos dos estados se corresponden con una situación favorecedora y otra más perjudicial para el mercado que se traducen en un estado con media positiva y baja varianza y otro con media algo negativa y mayor varianza respectivamente. Por tanto, dado que las variables de la base de datos que se trabaja se pueden ver como indicadores de mercado donde cada año puede estar en un estado de "mejoría" o de "retroceso" se supondrá que el número de estados es dos.

El segundo problema que se presenta es precisar de la distribución que siguen las variables observables. Esto es necesario para calcular la matriz de emisión. De nuevo, se desconoce la naturaleza de los datos y que distribución pudiera seguir. Determinar esto es una tarea ardua que se escapa a este proyecto por esto se sigue la vía de la detección de regímenes que las supone normales. Ya que se están tratando datos reales esto no tiene porqué darse. Es por esto que se le aplicarán a los datos la transformación de Yeo-Johnson, una ampliación de la Box-Cox a datos con valores negativos, que le aplica una transformación a los datos para que las variables se asemejen más a una Normal. La transformación que se le aplicará a la base de datos puede tener como consecuencia una pérdida de significación sobre los resultados. Es decir, los resultados sobre la base de datos transformada puede diferir de los obtenidos si no se aplicaran. En el ejemplo que se trabaja la intención es mostrar la implementación de los MMO a un caso real y al igual que en el ejemplo de DTW el resultado per se del Análisis de Conglomerados no es de interés sino el proceso.

A continuación se verá la implementación de lo anterior junto a una explicación de la transformación de Yeo-Johnson.

```
#### SE PARTE DE NUEVO DE LA BASE DE DATOS IMPUTADOS ####
datos <- read_excel("DatosImputados.xlsx", col_types = c("text", rep("numeric", 9)))
attach(datos)

#### CAMBIAR EL NOMBRE DE LAS VARIABLES ####

colnames(datos) <- c('Name', 'AdolFerRate', 'AFFinGPD', 'CO2', 'Exports',
                     'GPDgrowth', 'Imports', 'Inflation', 'LifeExpect', '
    EmploymentAgriculture')

#Esto se hace para aligerar notación más adelante.

#### NOMBRE DE LOS PAÍSES ####

v =  seq(1, 4340, by=28)
países = c()
for (i in v) {
  países = append(países, data[i,1])
}

#### LISTA DE PAÍSES COMO TIBBLES ####

lista_países = vector("list", 155)
for (i in c(1:155) ) {
```

```
  valores <- data[data$Name == países[i],]
  nombre <- as.character(países[i])
  lista_países[[i]] <- assign(nombre, valores)
}


#### SE VUELVEN A ASOCIAR LOS AÑOS A LOS PAÍSES Y SE ELIMINAN SUS NOMBRES ####

for (i in c(1:155)) {

  v = c(1992:2019)
  años = as_tibble_col(v, column_name = "Year")
  lista_países[[i]] = cbind(años, lista_países[[i]][,-1])
}

#### APLICACIÓN DE LA TRANSFORMACIÓN DE YEO-JOHNSON ####
library(bestNormalize)

lista_países_normales = vector("list", 155)

for (i in c(1:155) ) {
  país <- lista_países[[i]]

  for (j in c(2:10)) {

    var <- yeojohnson(as.numeric(unlist(país[j])), eps = 0.001)
    pais[j] <- var$x.t
  }

  valores <- país
  nombre <- as.character(países[i])
  lista_países_normales[[i]] <- assign(nombre, valores)
}
```

La transformación de Yeo-Johnson proporcionada por In-kwon Yeo y Richard A. Johnson en [94] consiste en aplicar la siguiente función a los datos

$$\psi(\lambda, x) = \begin{cases} \frac{(x+1)^{\lambda}-1}{\lambda} & \text{si } x \geq 0, \lambda \neq 0 \\ \log(x+1) & \text{si } x \geq 0, \lambda = 0 \\ -\frac{(-x+1)^{2-\lambda}-1}{2-\lambda} & \text{si } x < 0, \lambda \neq 2 \\ -\log(-x+1) & \text{si } x < 0, \lambda = 2 \end{cases}$$

Donde el parámetro λ se calcula por máxima verosimilitud como se muestra en el mismo artículo.

En el ejemplo que se está tratando se hace uso de la librería `bestNormalize` [95] en concreto de la función `yeojohnson()` para transformar cada variable de cada país y de la salida que proporciona se extrae `x.t`, los datos transformados.

Una vez los datos se han acomodado para implementar mejor los MMO se hará uso de la librería `depmixS4` [96] y la función `depmix()`. En el caso de los modelos se ha asumido independencia condicional entre las variables, un total de dos estados y que las variables respuesta son (parecidas a) normales. Tras usar esta función, se estiman los parámetros según el algoritmo EM usando `fit()` Esto se muestra a continuación.

```
#### CONSTRUCCIÓN Y AJUSTADO DE LOS MODELOS ####

library('depmixS4')

modelos = vector('list', 155)
fitmodelos = vector('list', 155)
```

```
set.seed(123)
for (i in c(1:155)) {

  mod <- depmix(list(AdolFerRate~1, AFFinGPD~1, CO2~1,
                     Exports~1, GPDgrowth~1, Imports~1,
                     Inflation~1, LifeExpect~1, EmploymentAgriculture~1),
                nstates=2, data=lista_paises_normales[[i]],
                family=list(gaussian(), gaussian(), gaussian(),
                            gaussian(), gaussian(), gaussian(),
                            gaussian(), gaussian(), gaussian()))
  fmod <- fit(mod)
  modelos[[i]] <- mod
  fitmodelos[[i]] <- fmod
}
```

Se tiene que la lista `fitmodelos` contiene a todos los modelos con los parámetros ajustados. Una vez que se han hecho estos modelos, para calcular la distancia entre países se usa la distancia Kullback-Leibler. Esta distancia presenta un problema y es el hecho que para calcular la distancia entre dos series se requiere del cálculo de $P(Z^i|\lambda_i)$, $P(Z^i|\lambda_j)$ y viceversa. Es decir, la verosimilitud de dadas la observación Z^i (con parámetros λ_i asociados a su MMO) que esta se produzca con los parámetros de ambos modelos (el i-ésimo y el j-ésimo) y análogamente para la observación Z^j. Se ha de hacer esto ya que para obtener una matriz simétrica se usará la distancia Kullback-Leibler simetrizada

$$d_{KLS}(P(Z^i|\lambda_i), P(Z^j|\lambda_j)) = \frac{1}{2}d_{KL}(P(Z^i|\lambda_i), P(Z^i|\lambda_j)) + \frac{1}{2}d_{KL}(P(Z^j|\lambda_j), P(Z^j|\lambda_i)).$$

Para calcular las probabilidades necesarias se usará el algoritmo forward-backward. Este fue propuesto por Baum en 1972 [97]. Se verá para el caso de un MMO discreto por sencillez, el caso continuo se trata en [42]. Este algoritmo se divide en dos partes.

Algoritmos Forward y Backward

Sea un Modelo Markoviano Oculto dado por una cadena de Markov $\{X_s\}_{s=1}^K$, con N estados, las variables observables $\{Y_s\}_{s=1}^K$, y parámetros $\lambda = (P, Q, \pi)$ donde $\pi = (\pi_1, \ldots, \pi_N)$, que se ajustan a una observación $O' = \{O'_s\}_{s=1}^K$ con $E = \{e_s\}_{s=1}^K$ sus estados asociados y una observación O (pudiendo ser $O = O'$). Se pretende calcular $P(O|\lambda)$. Para ello se usan en conjunto dos algoritmos.

Algoritmo Forward

Se define la variable forward $\alpha_t(j)$ como la probabilidad de la observación O hasta el tiempo t dado el estado e_j en el tiempo t y los parámetros λ:

$$\alpha_t(j) = P(Y_1 = O_1, Y_2 = O_2 \ldots Y_t = O_t, X_t = e_j|\lambda).$$

Se considera

$$b_j(O_s) = P(Y_s = O_s|X_s = e_j, \lambda)$$

la probabilidad de que se observe O_s dado el estado e_j y los parámetros λ. Si las observaciones son las asociadas al modelo se obtendrían los valores de la matriz de emisión, Q, siendo $b_j(O_s) = q_{js}$ si $s \neq 1$ y $b_j(O_1) = \pi_j$. Si se asume independencia entre las observaciones

$$P(O|E, \lambda) = \prod_{s=1}^{K} P(Y_s = O_s|X_s = e_s, \lambda) = \prod_{s=1}^{K} b_s(O_s).$$

Ahora la probabilidad conjunta de que ocurra la observación O con los estados E es

$$P(O, E|\lambda) = P(O|E, \lambda)P(E|\lambda).$$

Se tiene por tanto que la probabilidad $P(O|\lambda)$ es la suma en todas las posibles secuencias de estados de la probabilidad conjunta anterior. Así

$$P(O|\lambda) = \sum_{\text{todos los } E} P(O|E, \lambda)P(E|\lambda).$$

Para obtener $\alpha_t(j)$ en estos términos se toma la secuencia ya indicada de la observación hasta el tiempo t. Se tiene que

$$\alpha_{t+1}(j) = P(O_1, O_2 \dots O_{t+1}; X_{t+1} = e_j|\lambda) = \sum_{i=1}^{N} P(O_1, O_2 \dots O_t; X_t = e_i, X_{t+1} = j|\lambda) =$$

$$\sum_{i=1}^{N} P(O_{t+1}|X_{t+1} = e_j, X_t = i; O_1, \dots, O_t)P(X_{t+1} = e_j|O_1, \dots, O_t)P(X_t = i; O_1, \dots, O_t) =$$

$$\sum_{i=1}^{K} p_{ij} b_j(O_{t+1})\alpha_t(i).$$

Se pueden calcular los valores de $\alpha_t(j)$ por inducción como sigue:

1. Inicio:
$$\alpha_1(j) = \pi_j b_j(O_1)\ 1 \leq j \leq N.$$
2. Inducción:
$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) p_{ij} \right) b_j(O_{t+1})\ 1 \leq j \leq N, 1 \leq j \leq K - 1$$
3. Final:
$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_K(i)$$

Algoritmo Backward

En este algoritmo se define la variable backward como la probabilidad de la observación O desde el tiempo t hasta el final dado el estado e_j en el tiempo t y los parámetros λ

$$\beta_t(i) = P(Y_{t+1} = O_{t+1}, \dots, Y_K = O_K|X_t = e_i, \lambda).$$

Su cálculo se hace por inducción de manera parecida al algoritmo forward:

1. Inicio:
$$\beta_T(j) = 1\ 1 \leq j \leq N.$$

2. Inducción:
$$\beta_t(j) = \left(\sum_{i=1}^{N} p_{ij} b_j(O_{t+1})\right) \beta_{t+1}(j) \; 1 \leq j \leq N, 1 \leq j \leq K-1.$$

3. Final:
$$P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i)\pi_i b_i(O_1).$$

Para calcular la distancia de Kullback-Leibler se hará uso del algoritmo Forward. Siguiendo el artículo [42], asumiendo independencia entre las observaciones y por la definición de la variable forward se tiene que

$$\alpha_t(j) = P(O_1, O_2 \ldots, O_t, X_t = e_j|\lambda) \Rightarrow \sum_{j=1}^{N} \alpha_t(j) = P(O_1, O_2 \ldots, O_t|\lambda).$$

Por ser observaciones independientes

$$\sum_{j=1}^{N} \alpha_t(j) = P(O_1, O_2 \ldots, O_t|\lambda) = \prod_{i=1}^{t} P(O_i|\lambda).$$

Se tiene por tanto que se puede calcular $P(O_t|\lambda)$ $\forall t = 1, \ldots, K$ según sigue

$$\begin{aligned} P(O_1|\lambda) &= \sum_{j=1}^{K} \alpha_1(j) \\ P(O_1|\lambda)P(O_2|\lambda) &= \sum_{j=1}^{K} \alpha_2(j) \\ &\vdots \\ \prod_{i=1}^{K} P(O_i|\lambda) &= \sum_{j=1}^{K} \alpha_K(j). \end{aligned}$$

Como los valores $\alpha_1(j)$ son conocidos se tiene que las ecuaciones anteriores tiene solución de manera recursiva

$$\begin{aligned} P(O_1|\lambda) &= \sum_{j=1}^{K} \alpha_1(j) \\ P(O_2|\lambda) &= \frac{\sum_{j=1}^{K} \alpha_2(j)}{P(O_1|\lambda)} \\ &\vdots \\ P(O_K|\lambda) &= \frac{\sum_{j=1}^{K} \alpha_K(j)}{\prod_{i=1}^{K-1} P(O_i|\lambda)}. \end{aligned}$$

Estas probabilidades se normalizarán con

$$\sum_{i=1}^{K} P(O_i|\lambda)$$

para así obtener una distribución empírica. A continuación se verá la implementación de estos cálculos.

```
#### ALGORITMO FORWARD ####
forward <- function(pi,P,Q) {
  alpha <- matrix( 0, nrow = 28, ncol = 2)
  alpha[1,] <- c(pi[1]*B[1,1], pi[2]*B[1,2])

  for (t in 2:28) {
   for (i in 1:2) {
    alpha[t,i] <- (sum(alpha[t-1,]*P[,i]))*Q[t,i]
    }
   }
  return(alpha)

}
```

Donde `pi` es el vector de la distribución inicial, `P` es la matriz de transición y `Q` la de emisión.

Se procede ahora al cálculo de la matriz de distancia. Para ello se hará un bucle donde para hallar la distancia entre los países i y j se calcularán los respectivos algoritmos Forward donde para el caso del primer país se obtendrán los parámetros del elemento i-ésimo de `fitmodelos` y en el caso del segundo país se ajustará el modelo j-ésimo de `modelos` con los parámetros propios del primer país para obtener así también los parámetros necesarios. Tras esto se aplicará la función `calcular()` para obtener las probabilidades anteriores normalizadas de manera que cada una sea un vector de densidad de probabilidad. Con estos vectores y la función `KLD()` de la librería [98] se calculará la divergencia Kullback-Leibler entre los dos países. Al finalizar el bucle se simetriza la matriz obtenida para obtener la buscada. En algunos casos puede ocurrir que la función `fit()` no pueda ajustar el modelo j-ésimo con los parámetros del i-ésimo. Si lo anterior ocurre se sustituye ese valor por un `M` (respectivamente) grande de manera que se entenderá como que la forma de eso dos países es demasiado diferente.

```
#### MATRIZ DE DISTANCIA ####

#Se define una función que calcule los P(O_i|\lambda)

calcular = function(F) {
  v = rep(0,28)
  for (i in 2:28) {
    v[1] = sum(F[1,])
    v[i] = sum(F[i,])/v[i-1]
    }
  return(v/sum(v))
}

#Se procede a calcular la matriz

set.seed(123)
matriz_MMO = matrix(0, nrow=155,ncol=155)
M = 10

for (i in 1:155) {
  pi_1 = fitmodelos[[i]]@init   #distribución inicial
  P_1 = fitmodelos[[i]]@trDens
  P_1 = matrix(c(P_1[1], P_1[3],P_1[2],P_1[4]), 2, 2) #matriz de transición
  Q_1 = fitmodelos[[i]]@posterior[2:3] #matriz de emisión
  F_1 = forward(pi_1, P_1, Q_1)
  H_1 = calcular(F_1) #vector de densidad
```

```
  for (j in c(1:155)[-i]){
    posibleError <- tryCatch({  #detección del error de ajuste
      modelo2 = fit(setpars(modelos[[j]],getpars(fitmodelos[[i]]))),
      error = function(e) {e})
    if (!inherits(posibleError, "error")) {
      pi_2 = modelo2@init
      P_2 = modelo2@trDens
      P_2 = matrix(c(P_2[1], P_2[3],P_2[2],P_2[4]), 2, 2)
      Q_2 = modelo2@posterior[,2:3]
      F_2 = forward(pi_2, P_2, Q_2)
      H_2 = calcular(F_2)
      print(c(i,j))
      matriz_MMO[i,j] <-KLD(H_1, H_2)$sum.KLD.px.py  #distancia entre H_1 y H_2
    }
    else {
      matriz_MMO[i,j] <- M
    }
  }
}

#Se simetriza
matriz_MMO <- 0.5*matriz_MMO+ 0.5*t(matriz_MMO)
```

Ya calculada la matriz de distancia el resto de Análisis de Conglomerados es análogo al ya hecho en el caso de DTW

```
matriz_MMO_dist = as.dist(matriz_MMO)
congloMMO <- hclust(matriz_MMO_dist, method = "complete")
plot(congloMMO, labels = as.character(c(1:155)),  hang = 0.1, axes = TRUE,
     frame.plot = FALSE, sub = '', main = "Dendrograma", xlab = 'Unión completa',
     ylab = 'Distancia', col = 'blue')
```

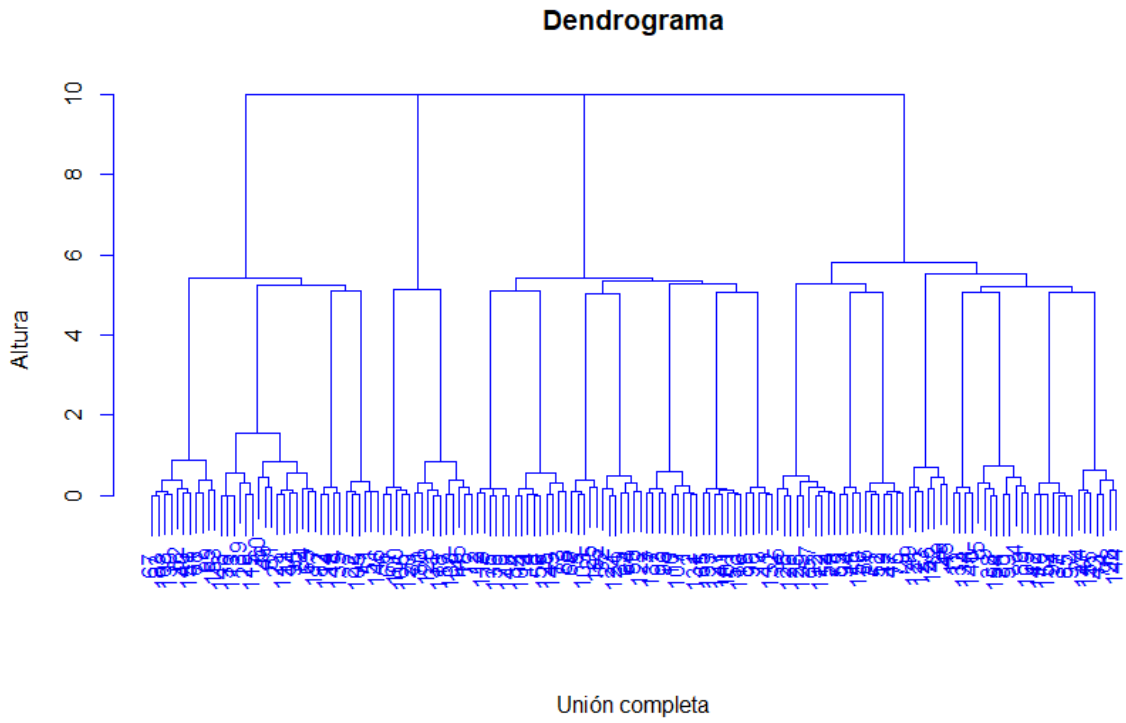


Figura 6.6: Dendrograma mostrando los conglomerados al aplicar MMO y un algoritmo jerárquico aglomerativo de unión completa sobre la base de datos normalizados.

Se procede al cálculo de los índices de validación para determinar el número óptimo de conglomerados

```
#### ÍNDICE DE DUNN ####

ind_dunn = c()
for (i in c(2:10)) {
  cluster <- cutree(congloMMO, i)
  ind_dunn <- append(ind_dunn, dunn(matriz_MMO, cluster))
}

#### ÍNDICE DE SILUETA ####

ind_sil = c()
for (i in c(2:10)) {
  cluster <- cutree(congloMMO, i)
  silueta <- summary(silhouette(cluster, matriz_MMO))$avg.width
  ind_sil <- append(ind_sil, silueta)
}

#### ÍNDICE DB ####

intraclust = c("complete")
interclust = c("complete")
ind_db = c()
for (i in c(2:10)) {
  cluster <- cutree(congloMMO, i)
  info <- cls.scatt.diss.mx(matriz_MMO, cluster)
  index <-clv.Davies.Bouldin(info, intracls = intraclust, intercls = interclust)
  ind_db <- append(ind_db, index)
}
```

```
#Se redondean los valores a tres cifras decimales
ind_sil = round(ind_sil, digits = 3)
ind_db = round(ind_db, digits = 3)

Índices <- matrix(c((c(2:10)), ind_dunn, ind_sil, ind_db), ncol = 4)
colnames(Índices) <- c('Número de conglomerados', 'Índice de Dunn', 'Índice de Silueta'
    , 'Índice DB')
Índices <- as.table(Índices)
```

Índices de evaluación de MMO			
Número de conglomerados	Dunn	Silueta	Davies-Bouldin
2	$7{,}841 \cdot 10^{-7}$	**0,084**	1,541
3	$7{,}841 \cdot 10^{-7}$	-0,023	1,532
4	$1{,}352 \cdot 10^{-6}$	-0,022	**1,115**
5	$1{,}421 \cdot 10^{-6}$	-0,055	1,394
6	$1{,}443 \cdot 10^{-6}$	-0,116	1,319
7	$\mathbf{1{,}45 \cdot 10^{-6}}$	-0,101	1,612
8	$9{,}985 \cdot 10^{-7}$	-0,103	1,580
9	$1{,}015 \cdot 10^{-6}$	-0,108	1,633
10	$1{,}018 \cdot 10^{-6}$	-0,105	1,574

Tabla 6.3: Tabla que muestra los distintos valores de los índices de Dunn, silueta y DB bajo distintos números de conglomerados. Destacan los valores óptimos

Se comprueba en la tabla y el código que no se ha redondeado a tres cifras decimales el índice de Dunn ya que los valores son muy pequeños. Si bien no hay consenso sobre la cantidad óptima de conglomerados se determinará que esta es 4 ya que es la cantidad determinada por el índice DB, la segunda mejor opción según el de silueta y la opción intuitiva si se observa el dendrograma. La división se puede representar como sigue.

```
fviz_dend(congloMMO, k = 4, ylab = '', k_colors = c('#34df2e', '#00AFBB', '#1050EF', '#
    F2B99B'), color_labels_by_k = TRUE, ggtheme = theme_classic(), main = "División en
    4 grupos",type = 'rectangle')
```

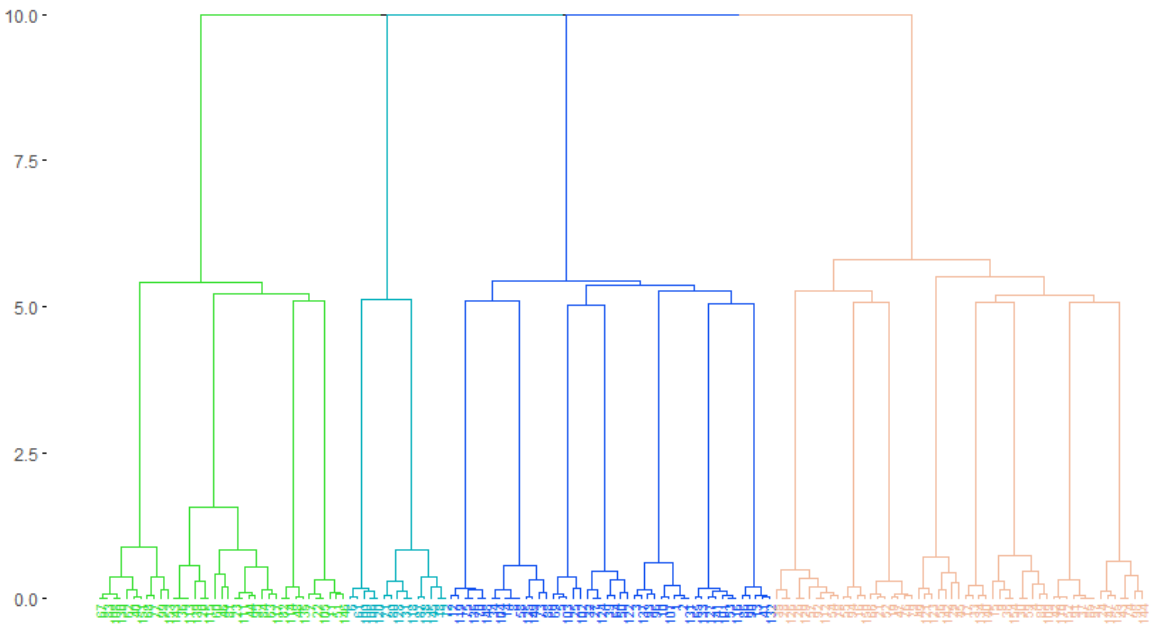


Figura 6.7: Dendrograma mostrando la división en cuatro conglomerados.

Para visualizar mejor los países se puede aplicar de nuevo escalamiento multidimensional a la matriz de distancia. En este caso no se asignarán los respectivos números a los países para permitir una visualización más clara.

```
#Se calculan los puntos en el espacio relacionados a los países

emdMMO <- cmdscale(matriz_MMO_dist, k=3)

#Se acomoda a un data.frame para la función scatterplot3d
x <- emdMMO[,1]
y <- emdMMO[,2]
z <- emdMMO[,3]

EMD <- cbind(c(1:155), x, y, z)
colnames(EMD) <- c('países', 'x', 'y', 'z')

# Representación gráfica
grafica <- scatterplot3d(EMD[,-1], angle = 35, box = FALSE, xlim = c(-3, 3), ylim =c
    (-3,3), zlim = c(-3, 3))
```

En la gráfica se ha hecho un zoom para mostrar los países con más claridad. Se observa que los puntos están muy próximos entre sí salvo contadas excepciones. Esto se debe a asumir una distancia `M` entre los países la cual provoca esa clara división en 4 conglomerados en el dendrograma pero al estar el resto de puntos tan próximos hace que los valores óptimos de los índices sean peores que los obtenidos en el análisis análogo con DTW. Esto se verá en al siguiente sección.

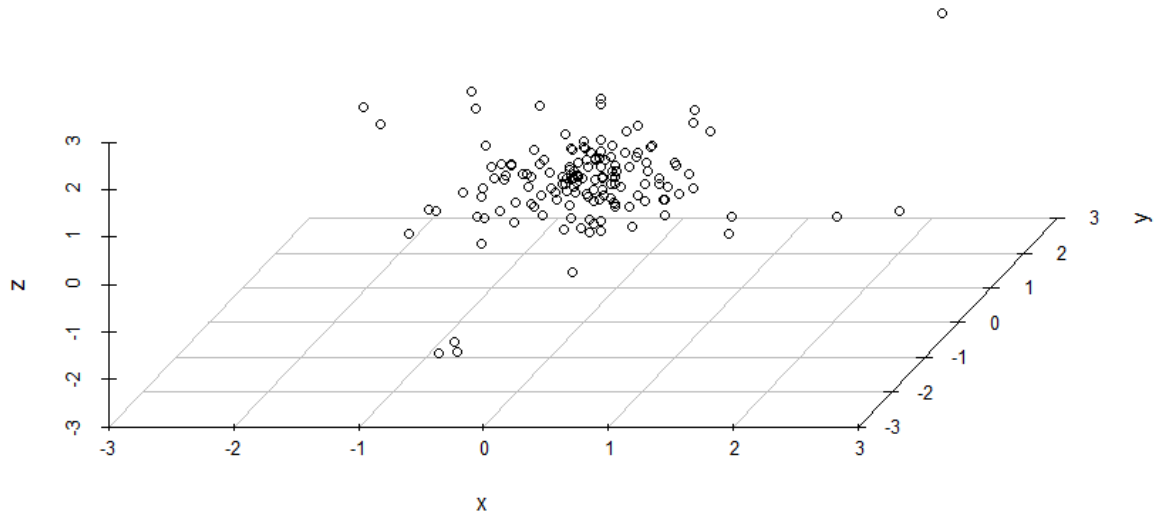


Figura 6.8: Representación en el espacio de las posiciones relativas de los países. Muestra un zoom.

6.2 Conclusiones

Es fácil comprobar que los valores de los índices de evaluación son mejores en el caso de DTW. Además para el caso de MMO se han tenido que hacer distintas consideraciones como asumir el número óptimo de estados, la distribución de los variables observables, transformar los datos o asumir una distancia entre modelos que no se podían ajustar.

Índices de evaluación			
Tipo de técnica	Dunn	Silueta	Davies-Bouldin
MMO	$1{,}352 \cdot 10^{-6}$	-0,022	1,115
DTW	0,115	0,334	1,195

Tabla 6.4: Tabla mostrando los valores de los índices en los dos ejemplos expuestos correspondientes al número considerado óptimo de conglomerados.

En esta tabla se ven reflejadas las numerosas asunciones por parte de la técnica de MMO. Así se comprueba que permitir una gran flexibilidad, adaptabilidad y posibilidades de interpretación no es siempre sinónimo de unos mejores resultados siendo en este caso la técnica más sencilla mucho más eficiente y fiable. Con esto se pone en práctica toda la teoría presentada y se remarca la importancia de conocer distintas técnicas, sus ventajas, desventajas y los datos con los que se trabaja a la hora de elegir una para una Análisis de Conglomerados. Además se comprueba la importancia del cálculo de la matriz de distancia que en este documento se ha tomado como el problema principal.

Se concluye así que dependiendo de los datos y el conocimiento que se tenga sobre ellos unas técnicas se adaptan mejor que otras no siendo siempre la más compleja la mejor opción.

Bibliografía

[1] José-María López-Lozano et al. «Modelling and forecasting antimicrobial resistance and its dynamic relationship to antimicrobial use: a time series analysis». En: *International Journal of Antimicrobial Agents* 14.1 (feb. de 2000), págs. 21-31. ISSN: 0924-8579.

[2] Krishnan Bhaskaran et al. «Time series regression studies in environmental epidemiology». En: *International Journal of Epidemiology* 42.4 (ago. de 2013), págs. 1187-1195. ISSN: 1464-3685, 0300-5771.

[3] N. Radhakrishnan y B.N. Gangadhar. «Estimating regularity in epileptic seizure time-series data». En: *IEEE Engineering in Medicine and Biology Magazine* 17.3 (jun. de 1998), págs. 89-94. ISSN: 07395175.

[4] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi y Teh Ying Wah. «Time-series clustering – A decade review». En: *Information Systems* 53 (oct. de 2015), págs. 16-38. ISSN: 0306-4379. DOI: 10.1016/j.is.2015.04.007.

[5] Charu C. Aggarwal, Alexander Hinneburg y Daniel A. Keim. «On the Surprising Behavior of Distance Metrics in High Dimensional Space». En: *Database Theory — ICDT 2001*. Ed. por Jan Van den Bussche y Victor Vianu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, págs. 420-434. ISBN: 978-3-540-44503-6.

[6] Lluís A Belanche. «Understanding (dis)similarity measures». En: *arXiv preprint arXiv:1212.2791* (2012).

[7] J. C. Gower y P. Legendre. «Metric and Euclidean properties of dissimilarity coefficients». En: *Journal of Classification* 3.1 (mar. de 1986), págs. 5-48. ISSN: 0176-4268, 1432-1343. DOI: 10.1007/BF01896809. URL: http://link.springer.com/10.1007/BF01896809.

[8] Kevin Beyer et al. «When Is "Nearest Neighbor" Meaningful?» En: *Database Theory — ICDT'99*. Ed. por Catriel Beeri y Peter Buneman. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, págs. 217-235. ISBN: 978-3-540-49257-3.

[9] Roger Weber, Hans-Jörg Schek y Stephen Blott. «A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces». En: VLDB '98 (1998), págs. 194-205.

[10] A. K. Jain, M. N. Murty y P. J. Flynn. «Data Clustering: A Review». En: *ACM Comput. Surv.* 31.3 (sep. de 1999), págs. 264-323. ISSN: 0360-0300. DOI: 10.1145/331499.331504. URL: https://doi.org/10.1145/331499.331504.

[11] Jianchang Mao y A.K. Jain. «A self-organizing network for hyperellipsoidal clustering (HEC)». En: *IEEE Transactions on Neural Networks* 7.1 (1996), págs. 16-29. DOI: 10.1109/72.478389.

[12] Adem Karahoca. *Advances in Data Mining Knowledge Discovery and Applications*. Rijeka: InteChOpen, sep. de 2012, págs. 77-81. ISBN: 978-953-51-0748-4. DOI: 10.5772/3349. URL: https://doi.org/10.5772/3349.

[13] Xiaoyue Wang et al. «Experimental comparison of representation methods and distance measures for time series data». En: *Data Mining and Knowledge Discovery* 26.2 (mar. de 2013), págs. 275-309. ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-012-0250-5. URL: http://link.springer.com/10.1007/s10618-012-0250-5.

[14] International Conference FODO y David B. Lomet, eds. *Foundations of data organization and algorithms: 4th International Conference, FODO '93, Chicago, Illinois, USA, October 13-15, 1993: proceedings*. Lecture notes in computer science 730. Berlin ; New York: Springer-Verlag, 1993. ISBN: 978-3-540-57301-2.

[15] Christos Faloutsos, M. Ranganathan y Yannis Manolopoulos. «Fast Subsequence Matching in Time-Series Databases». En: *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*. SIGMOD '94. Minneapolis, Minnesota, USA: Association for Computing Machinery, 1994, págs. 419-429. ISBN: 0897916395. DOI: 10.1145/191839.191925. URL: https://doi.org/10.1145/191839.191925.

[16] Flip Korn, H. V. Jagadish y Christos Faloutsos. «Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences». En: *SIGMOD Rec.* 26.2 (jun. de 1997), págs. 289-300. ISSN: 0163-5808. DOI: 10.1145/253262.253332. URL: https://doi.org/10.1145/253262.253332.

[17] Ahlame Douzal Chouakria y Panduranga Naidu Nagabhushan. «Adaptive dissimilarity index for measuring time series proximity». En: *Advances in Data Analysis and Classification* 1.1 (feb. de 2007), págs. 5-21. ISSN: 1862-5347, 1862-5355. DOI: 10.1007/s11634-006-0004-6. URL: http://link.springer.com/10.1007/s11634-006-0004-6.

[18] Xavier Golay et al. «A new correlation-based fuzzy logic clustering algorithm for FMRI». En: *Magnetic Resonance in Medicine* 40.2 (ago. de 1998), págs. 249-260. ISSN: 07403194, 15222594. DOI: 10.1002/mrm.1910400211. URL: https://onlinelibrary.wiley.com/doi/10.1002/mrm.1910400211.

[19] Thomas Eiter y Heikki Mannila. «Computing discrete Fréchet distance». En: (1994).

[20] Lei Chen y M. Tamer Özsu. «Robust and fast similarity search for moving object trajectories». En: *In SIGMOD*. 2005, págs. 491-502.

[21] Lei Chen y Raymond Ng. «On The Marriage of Lp-norms and Edit Distance.» En: (ene. de 2004), págs. 792-803.

[22] M. Vlachos, G. Kollios y D. Gunopulos. «Discovering similar multidimensional trajectories». En: *Proceedings 18th International Conference on Data Engineering*. San Jose, CA, USA: IEEE Comput. Soc, 2002, págs. 673-684. ISBN: 978-0-7695-1531-1. DOI: 10.1109/ICDE.2002.994784. URL: http://ieeexplore.ieee.org/document/994784/.

[23] Ahmed Shifaz et al. «Elastic Similarity Measures for Multivariate Time Series Classification». En: *arXiv:2102.10231* (feb. de 2021). arXiv: 2102.10231. URL: http://arxiv.org/abs/2102.10231.

[24] Masa Kljun y Matija Tersek. «A review and comparison of time series similarity measures». En: (). URL: https://erk.fe.uni-lj.si/2020/papers/kljun(a_review).pdf.

[25] Kerem Sinan Tuncel y Mustafa Gokce Baydogan. «Autoregressive forests for multivariate time series modeling». En: *Pattern Recognition* 73 (2018), págs. 202-215. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2017.08.016. URL: https://www.sciencedirect.com/science/article/pii/S0031320317303291.

[26] J. C. Gower. «A General Coefficient of Similarity and Some of Its Properties». En: *Biometrics* 27.4 (1971). Publisher: [Wiley, International Biometric Society], págs. 857-871. ISSN: 0006-341X. DOI: 10.2307/2528823. URL: https://www.jstor.org/stable/2528823.

[27] Christian H. Weiß y Rainer Göb. «Measuring serial dependence in categorical time series». En: *AStA Advances in Statistical Analysis* 92.1 (feb. de 2008), págs. 71-89. ISSN: 1863-8171, 1863-818X. DOI: 10.1007/s10182-008-0055-4. URL: http://link.springer.com/10.1007/s10182-008-0055-4.

[28] Shima Ghassempour, Federico Girosi y Anthony Maeder. «Clustering Multivariate Time Series Using Hidden Markov Models». En: *International Journal of Environmental Research and Public Health* 11.3 (mar. de 2014), págs. 2741-2763. ISSN: 1661-7827. DOI: 10.3390/ijerph110302741. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3968966/.

[29] Ville Hautamaki, Pekka Nykanen y Pasi Franti. «Time-series clustering by approximate prototypes». En: *2008 19th International Conference on Pattern Recognition*. Tampa, FL, USA: IEEE, dic. de 2008, págs. 1-4. ISBN: 978-1-4244-2174-9. DOI: 10.1109/ICPR.2008.4761105. URL: http://ieeexplore.ieee.org/document/4761105/.

[30] T. Warren Liao. «Clustering of time series data—a survey». En: *Pattern Recognition* 38.11 (nov. de 2005), págs. 1857-1874. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2005.01.025. URL: https://www.sciencedirect.com/science/article/pii/S0031320305001305.

[31] Nazanin Asadi, Abdolreza Mirzaei y Ehsan Haghshenas. «Creating Discriminative Models for Time Series Classification and Clustering by HMM Ensembles». En: *IEEE Transactions on Cybernetics* 46.12 (dic. de 2016), págs. 2899-2910. ISSN: 2168-2275. DOI: 10.1109/TCYB.2015.2492920.

[32] H. Sakoe y S. Chiba. «Dynamic programming algorithm optimization for spoken word recognition». En: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (feb. de 1978), págs. 43-49. ISSN: 0096-3518. DOI: 10.1109/TASSP.1978.1163055.

[33] Eamonn Keogh, Stefano Lonardi y Chotirat Ann Ratanamahatana. «Towards parameter-free data mining». En: *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*. Seattle, WA, USA: ACM Press, 2004, págs. 206-215. DOI: 10.1145/1014052.1014077. URL: http://portal.acm.org/citation.cfm?doid=1014052.1014077.

[34] Toni Giorgino. «Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package». En: *Journal of Statistical Software* 31.7 (2009), págs. 1-24. DOI: 10.18637/jss.v031.i07. URL: https://www.jstatsoft.org/index.php/jss/article/view/v031i07.

[35] F. Itakura. «Minimum prediction residual principle applied to speech recognition». En: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.1 (feb. de 1975), págs. 67-72. ISSN: 0096-3518. DOI: 10.1109/TASSP.1975.1162641.

[36] Alexis Sardá-Espinosa. «Comparing time-series clustering algorithms in r using the dtwclust package». En: *R package vignette* 12 (2017).

[37] Daniel Lemire. «Faster retrieval with a two-pass dynamic-time-warping lower bound». En: *Pattern Recognition* 42.9 (sep. de 2009), págs. 2169-2180. ISSN: 00313203. DOI: 10.1016/j.patcog.2008.11.030. URL: https://linkinghub.elsevier.com/retrieve/pii/S0031320308004925.

[38] Wai-Ki Ching et al. *Markov Chains*. Vol. 189. International Series in Operations Research & Management Science. Boston, MA: Springer US, 2013, págs. 19, 20, 35, 36. ISBN: 978-1-4614-6311-5. DOI: 10.1007/978-1-4614-6312-2. URL: http://link.springer.com/10.1007/978-1-4614-6312-2.

[39] L. Rabiner y B. Juang. «An introduction to hidden Markov models». En: *IEEE ASSP Magazine* 3.1 (ene. de 1986), págs. 4-16. ISSN: 1558-1284. DOI: 10.1109/MASSP.1986.1165342.

[40] William J Anderson. *Continuous-time Markov chains: An applications-oriented approach*. Springer Science & Business Media, 2012, pág. 1.

[41] Lawrence R. Rabiner. «Mathematical Foundations of Hidden Markov Models». En: *Recent Advances in Speech Understanding and Dialog Systems*. Ed. por H. Niemann, M. Lang y G. Sagerer. NATO ASI Series. Berlin, Heidelberg: Springer, 1988, págs. 183-205. ISBN: 978-3-642-83476-9. DOI: 10.1007/978-3-642-83476-9_19.

[42] L.R. Rabiner. «A tutorial on hidden Markov models and selected applications in speech recognition». En: *Proceedings of the IEEE* 77.2 (feb. de 1989), págs. 257-286. ISSN: 1558-2256. DOI: 10.1109/5.18626.

[43] Konrad Banachewicz, André Lucas y Aad van der Vaart. «Modelling portfolio defaults using hidden Markov models with covariates». En: *The Econometrics Journal* 11.1 (2008). Publisher: [Royal Economic Society, Wiley], págs. 155-171. ISSN: 1368-4221. URL: https://www.jstor.org/stable/23116066.

[44] S. Kullback y R. A. Leibler. «On Information and Sufficiency». En: *The Annals of Mathematical Statistics* 22.1 (1951). Publisher: Institute of Mathematical Statistics, págs. 79-86. ISSN: 0003-4851. URL: https://www.jstor.org/stable/2236703.

[45] Frank Nielsen. «On the Kullback-Leibler divergence between discrete normal distributions». En: *Journal of the Indian Institute of Science* (ene. de 2022). arXiv:2109.14920 [cs, math]. ISSN: 0970-4140, 0019-4964. DOI: 10.1007/s41745-021-00279-5. URL: http://arxiv.org/abs/2109.14920.

[46] Teemu Räsänen y Mikko Kolehmainen. «Feature-Based Clustering for Electricity Use Time Series Data». En: *Adaptive and Natural Computing Algorithms*. Ed. por Mikko Kolehmainen, Pekka Toivanen y Bartlomiej Beliczynski. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, págs. 401-412. ISBN: 978-3-642-04921-7. DOI: 10.1007/978-3-642-04921-7_41.

[47] Julien C. Sprott. *Chaos and time-series analysis*. Oxford ; New York: Oxford University Press, 2003. ISBN: 978-0-19-850840-3.

[48] Holger Kantz. «A robust method to estimate the maximal Lyapunov exponent of a time series». En: *Physics Letters A* 185.1 (ene. de 1994), págs. 77-87. ISSN: 0375-9601. DOI: 10.1016/0375-9601(94)90991-1. URL: https://www.sciencedirect.com/science/article/pii/0375960194909911.

[49] Lambert H. Koopmans. *The spectral analysis of time series*. Probability and mathematical statistics, 22. New York: Academic Press, 1974. ISBN: 978-0-12-419250-8.

[50] William N. Venables y Brian D. Ripley. *Modern Applied Statistics with S*. 4ª ed. Statistics and Computing. New York, NY: Springer New York, 2010. ISBN: 978-1-4419-3008-8.

[51] Xiaozhe Wang, Kate Smith-Miles y Rob Hyndman. «Characteristic-Based Clustering for Time Series Data». En: 13 (sep. de 2006), págs. 335-364. DOI: 10.1007/s10618-005-0039-x.

[52] Whitney Griggs. «Penalized spline regression and its applications». En: *Whitman College Report. Available online at: https://www. whitman. edu/Documents/Academics/Mathematics/Griggs. pdf* (2013).

[53] Spyros Makridakis, S. Wheelwright y Rob Hyndman. «Forecasting: Methods and Applications». En: *The Journal of the Operational Research Society*. Vol. 35. Journal Abbreviation: The Journal of the Operational Research Society. Ene. de 1984. DOI: 10.2307/2581936.

[54] Xiaozhe Wang, A. Wirth y Liang Wang. «Structure-Based Statistical Features and Multivariate Time Series Clustering». En: (2007). DOI: 10.1109/ICDM.2007.103.

[55] J. Timmer et al. «Characteristics of hand tremor time series». En: *Biological Cybernetics* 70.1 (1993), págs. 75-80. ISSN: 0340-1200. DOI: 10.1007/BF00202568.

[56] J. Jeong, J. C. Gore y B. S. Peterson. «Mutual information analysis of the EEG in patients with Alzheimer's disease». En: *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology* 112.5 (mayo de 2001), págs. 827-835. ISSN: 1388-2457. DOI: 10.1016/s1388-2457(01)00513-2.

[57] Jeffrey D. Banfield y Adrian E. Raftery. «Model-Based Gaussian and Non-Gaussian Clustering». En: *Biometrics* 49.3 (1993). Publisher: Wiley, International Biometric Society, págs. 803-821. ISSN: 0006-341X. DOI: 10.2307/2532201. URL: https://www.jstor.org/stable/2532201.

[58] Chris Fraley y Adrian Rafter. «MCLUST Version 3 for R: Normal mixture modeling and model-based clustering». En: *Technical Report, Department of Statistics, University of Washington* 504 (mayo de 2012), pág. 7.

[59] A. P. Dempster, N. M. Laird y D. B. Rubin. «Maximum Likelihood from Incomplete Data Via the EM Algorithm». En: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), págs. 1-22. ISSN: 2517-6161. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

[60] C. F. Jeff Wu. «On the Convergence Properties of the EM Algorithm». En: *The Annals of Statistics* 11.1 (1983). Publisher: Institute of Mathematical Statistics, págs. 95-103. ISSN: 0090-5364. URL: https://www.jstor.org/stable/2240463.

[61] Yimin Xiong y Dit-Yan Yeung. «Time series clustering with ARMA mixtures». En: *Pattern Recognition* 37.8 (ago. de 2004), págs. 1675-1689. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2003.12.018. URL: https://www.sciencedirect.com/science/article/pii/S0031320304000585.

[62] George E. P. Box, Gwilym M. Jenkins y Gregory C. Reinsel. *Time series analysis: forecasting and control*. 4ª ed. Wiley series in probability and statistics. OCLC: ocn176895531. Hoboken, N.J: John Wiley, 2008, pág. 11. ISBN: 978-0-470-27284-8.

[63] Greg Ridgeway. *Finite discrete Markov process clustering*. Inf. téc. Technical Report TR 97-24, Microsoft Research, Redmond, WA, 1997.

[64] Hans H. Bock. «Probabilistic models in cluster analysis». En: *Computational Statistics & Data Analysis* 23.1 (nov. de 1996), págs. 5-28. ISSN: 01679473. DOI: 10.1016/0167-9473(96)88919-5. URL: https://linkinghub.elsevier.com/retrieve/pii/0167947396889195.

[65] Amit Saxena et al. «A review of clustering techniques and developments». En: *Neurocomputing* 267 (dic. de 2017), págs. 664-681. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.06.053. URL: https://www.sciencedirect.com/science/article/pii/S0925231217311815.

[66] Maria Halkidi, Yannis Batistakis y Michalis Vazirgiannis. «On clustering validation techniques». En: *Journal of intelligent information systems* 17.2 (2001), págs. 107-145.

[67] Eréndira Rendón et al. «A comparison of internal and external cluster validation indexes». En: *Proceedings of the 2011 American Conference, San Francisco, CA, USA*. Vol. 29. 2011, págs. 1-10.

[68] J. C. Dunn. «Well-Separated Clusters and Optimal Fuzzy Partitions». En: *Journal of Cybernetics* 4.1 (ene. de 1974), págs. 95-104. ISSN: 0022-0280. DOI: 10.1080/01969727408546059. URL: https://www.tandfonline.com/doi/full/10.1080/01969727408546059.

[69] N. R. Pal y J. Biswas. «Cluster validation using graph theoretic concepts». En: *Pattern Recognition* 30.6 (jun. de 1997), págs. 847-857. ISSN: 0031-3203. DOI: 10.1016/S0031-3203(96)00127-6. URL: https://www.sciencedirect.com/science/article/pii/S0031320396001276.

[70] Godfried T. Toussaint. «The relative neighbourhood graph of a finite planar set». En: *Pattern Recognition* 12.4 (ene. de 1980), págs. 261-268. ISSN: 00313203. DOI: 10.1016/0031-3203(80)90066-7. URL: https://linkinghub.elsevier.com/retrieve/pii/0031320380900667.

[71] K. Ruben Gabriel y Robert R. Sokal. «A New Statistical Approach to Geographic Variation Analysis». En: *Systematic Zoology* 18.3 (sep. de 1969), págs. 259-278. ISSN: 00397989. DOI: 10.2307/2412323. URL: https://academic.oup.com/sysbio/article-lookup/doi/10.2307/2412323.

[72] David Davies y Don Bouldin. «A Cluster Separation Measure». En: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-1 (mayo de 1979), págs. 224-227. DOI: 10.1109/TPAMI.1979.4766909.

[73] T. Caliński y J Harabasz. «A dendrite method for cluster analysis». En: *Communications in Statistics* 3.1 (1974), págs. 1-27. DOI: 10.1080/03610927408827101.

[74] Leonard Kaufman y Peter J. Rousseeuw, eds. *Finding Groups in Data*. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., mar. de 1990. ISBN: 978-0-470-31680-1. DOI: 10.1002/9780470316801. URL: http://doi.wiley.com/10.1002/9780470316801.

[75] Adrian E. Raftery. «A Note on Bayes Factors for Log-Linear Contingency Table Models with Vague Prior Information». En: *Journal of the royal statistical society series b-methodological* 48 (1986), págs. 249-250.

[76] Asa Ben-Hur, Andre Elisseeff e Isabelle Guyon. «A stability based method for discovering structure in clustered data». En: *Biocomputing 2002*. Kauai, Hawaii, USA: WORLD SCIENTIFIC, dic. de 2001, págs. 6-17. ISBN: 978-981-02-4777-5. DOI: 10.1142/9789812799623_0002. URL: http://www.worldscientific.com/doi/abs/10.1142/9789812799623_0002.

[77] Asa Ben-Hur e Isabelle Guyon. «Detecting Stable Clusters Using Principal Component Analysis». En: *Methods in molecular biology (Clifton, N.J.)* 224 (feb. de 2003), págs. 159-82. DOI: 10.1385/1-59259-364-X:159.

[78] *World Development Indicators | DataBank*. URL: https://databank.worldbank.org/source/world-development-indicators.

[79] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc. Boston, MA, 2019. URL: http://www.rstudio.com/.

[80] Hadley Wickham y Jennifer Bryan. *readxl: Read Excel Files*. R package version 1.4.0. 2022. URL: https://CRAN.R-project.org/package=readxl.

[81] Hadley Wickham et al. «Welcome to the tidyverse». En: *Journal of Open Source Software* 4.43 (2019), pág. 1686. DOI: 10.21105/joss.01686.

[82] Steffen Moritz y Thomas Bartz-Beielstein. «imputeTS: Time Series Missing Value Imputation in R». En: *The R Journal* 9.1 (2017), págs. 207-218. DOI: 10.32614/RJ-2017-009.

[83] Jeroen Ooms. *writexl: Export Data Frames to Excel 'xlsx' Format*. R package version 1.4.0. 2021. URL: https://CRAN.R-project.org/package=writexl.

[84] Roderick J. A. Little y Donald B. Rubin. *Statistical analysis with missing data*. 3ª ed. Wiley series in probability and statistics. Hoboken, NJ: Wiley, 2020, págs. 13-14. ISBN: 978-0-470-52679-8.

[85] Wen Yu, Edgar N. Sanchez y Janusz Kacprzyk, eds. *Advances in Computational Intelligence*. Vol. 116. Advances in Intelligent and Soft Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN: 978-3-642-03155-7. DOI: 10.1007/978-3-642-03156-4. URL: http://link.springer.com/10.1007/978-3-642-03156-4.

[86] Yanjun Qi, Judith Klein-Seetharaman y Ziv Bar-Joseph. «Random Forest Similarity for Protein-Protein Interaction Prediction From Multiple Sources». En: *Biocomputing 2005*. Hawaii, USA: World Scientific, dic. de 2004, págs. 531-542. ISBN: 978-981-256-046-9. DOI: 10.1142/9789812702456_0050. URL: http://www.worldscientific.com/doi/abs/10.1142/9789812702456_0050.

[87] Andy Liaw y Matthew Wiener. «Classification and Regression by randomForest». En: *R News* 2.3 (2002), págs. 18-22. URL: https://CRAN.R-project.org/doc/Rnews/.

[88] Maximilian Leodolter, Claudia Plant y Norbert Brändle. «IncDTW: An R Package for Incremental Calculation of Dynamic Time Warping». En: *Journal of Statistical Software* 99.9 (2021), págs. 1-23. DOI: 10.18637/jss.v099.i09.

[89] Guy Brock et al. «clValid: An R Package for Cluster Validation». En: *Journal of Statistical Software* 25.4 (2008), págs. 1-22. URL: https://www.jstatsoft.org/v25/i04/.

[90] Martin Maechler et al. *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.2. 2021. URL: https://CRAN.R-project.org/package=cluster.

[91] Lukasz Nieweglowski. *clv: Cluster Validation Techniques*. R package version 0.3-2.2. 2020. URL: https://CRAN.R-project.org/package=clv.

[92] Uwe Ligges y Martin Mächler. «Scatterplot3d - an R Package for Visualizing Multivariate Data». En: *Journal of Statistical Software* 8.11 (2003), págs. 1-20. URL: http://www.jstatsoft.org.

[93] Alboukadel Kassambara y Fabian Mundt. *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.7. 2020. URL: https://CRAN.R-project.org/package=factoextra.

[94] In-Kwon Yeo y Richard A. Johnson. «A new family of power transformations to improve normality or symmetry». En: *Biometrika* 87.4 (dic. de 2000), págs. 954-959. ISSN: 0006-3444. DOI: 10.1093/biomet/87.4.954. eprint: https://academic.oup.com/biomet/article-pdf/87/4/954/633221/870954.pdf. URL: https://doi.org/10.1093/biomet/87.4.954.

[95] Ryan A. Peterson y Joseph E. Cavanaugh. «Ordered quantile normalization: a semiparametric transformation built for the cross-validation era». En: *Journal of Applied Statistics* 47.13-15 (2020), págs. 2312-2327. DOI: 10.1080/02664763.2019.1630372.

[96] Ingmar Visser y Maarten Speekenbrink. «depmixS4: An R Package for Hidden Markov Models». En: *Journal of Statistical Software* 36.7 (2010), págs. 1-21. URL: https://www.jstatsoft.org/v36/i07/.

[97] Leonard E Baum et al. «An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes». En: *Inequalities* 3.1 (1972), págs. 1-8.

[98] Statisticat y LLC. *LaplacesDemon Tutorial*. R package version 16.1.6. Bayesian-Inference.com, 2021. URL: https://web.archive.org/web/20150206004624/http://www.bayesian-inference.com/software.