Trabajo Fin de Máster

Un estudio computacional de diferentes técnicas de clustering

Fausto Varela Durán

Máster en Técnicas Estadísticas

2017-2018

Propuesta de Trabajo Fin de Máster

Título en galego: Un estudo computacional de diferentes técnicas de clustering
Título en español: Un estudio computacional de diferentes técnicas de clustering
English title: A computational study of some clustering techniques
Modalidad: B Modalidad B
Autor: Fausto Varela Durán, Universidad de Santiago de Compostela
Directores: Balbina Casas Méndez, Universidad de Santiago de Compostela; Juan Carlos Vidal Aguiar, Universidad de Santiago de Compostela
Tutor: Marcos Matabuena Rodríguez, Citius
Breve resumen del trabajo: El análisis cluster es una técnica de análisis multivariante que permite agrupar un conjunto de datos, encontrándose en un grupo los datos más similares entre sí; y siendo los grupos formados lo más diferentes entre sí. Se revisan primero algunas técnicas de clustering, sin considerar aquellas para datos de naturaleza específica. Se comparan los métodos estudiados a partir de la simulación estadística de datos provenientes de distribuciones paramétricas conocidas y con características claramente diferenciadas. Finalmente se realiza un estudio computacional con datos simulados para establecer, en un entorno controlado, las diferencias reales de los algoritmos considerados. Para ello se emplean contrastes de hipótesis, a los que se aplican procedimientos de corrección por comparaciones múltiples.
Recomendaciones: Haber cursado Análisis Multivariante y Simulación Estadística.
Otras observaciones:

Doña Balbina Casas Méndez, Doctora de la Universidad de Santiago de Compostela, don Juan Carlos Vidal Aguiar, Doctor de la Universidad de Santiago de Compostela, don Marcos Matabuena Rodríguez, Becario investigador de Citius, informan que el Trabajo Fin de Máster titulado

Un estudio computacional de diferentes técnicas de clustering

fue realizado bajo su dirección por don Fausto Varela Durán para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En Santiago, a 5 de Septiembre de 2018.

La directora:

Doña Balbina Casas Méndez

El director/a:

Don Juan Carlos Vidal Aguiar

El tutor:

Don Marcos Matabuena Rodríguez

El autor:

Don Fausto Varela Durán

Agradecimientos

A los directores Balbina Casas y Juan Carlos Vidal por haberme dado la oportunidad de realizar el TFM en la modalidad de prácticas en el Citius, y al investigador Marcos Matabuena, por haberme tutorizado durante dicho período.

Índice general

Resumen

Resumen en español

El análisis clúster es una técnica de análisis multivariante que permite agrupar un conjunto de datos, de tal forma que en un mismo grupo se encuentren los datos más similares entre sí, siendo además, los diferentes grupos formados lo más dispares entre sí. En este trabajo, se pretende realizar en primer lugar una recopilación bibliográfica de algunas de las más relevantes técnicas de clustering dentro de un contexto general de un análisis estadístico de datos, es decir, sin considerar técnicas para datos de naturaleza específica, como pueden ser datos con componente temporal, espacial o funcional.

En una segunda fase, se compararán algunos de los distintos métodos estudiados a partir de la simulación estadística de datos provenientes de distribuciones paramétricas conocidas y con características claramente diferenciadas. El objeto fundamental de este trabajo es la realización de un intenso estudio computacional con datos simulados, para establecer en un entorno controlado las diferencias reales de algunos de los algoritmos más populares de clustering. La significancia real de las diferencias entre algoritmos será establecida mediante el uso de contrastes de hipótesis y se aplicarán procedimientos de corrección por comparaciones múltiples.

English abstract

Cluster analysis is a multivariate technique which allows to group a dataset, in a manner that finds the most simmilar data points among them, and at the same time form the most dissimilar groups among them. In this work, the aim is to do at first a literature review of the most relevant clustering techniques in an statistical data analysis overall context, without considering specific nature data such as time, space or functional. At a second stage, a comparison of the different methods studied will be held, by simulating statistical data from well-known parametric distributions, all with clear different features. The aim of this work is the implementation of an intense computational study using simulated data to determine, in a well contolled environment, the real differences among some of the most popular clustering algorithms. The real significance of the differences among those algorithms will be established by the use of hypothesis testing and different procedures of correction for multiple comparisons will be made.

Capítulo 1

Tópicos básicos del análisis cluster

1.1. Introducción

El objetivo de esta técnica del análisis multivariante es partiendo de un conjunto de datos X, formar k grupos, de tal forma que se optimice algún criterio de decisión establecido previamente.

En un problema de clasificación cada dato tiene asociado una variable que indica a qué clase pertenece. Contrariamente, en el análisis de grupos esta información no se encuentra disponible. Por ello, en todo algoritmo de agrupamiento, a diferencia de cualquier modelo de regresión o clasificación, la función a optimizar únicamente tiene en cuenta la estructura interna de los datos (según el criterio especificado mediante la función objetivo), y no el grado de precisión del resultado obtenido con el modelo frente a una muestra $\{Y_i\}_{i=1}^n$ de una variable aleatoria respuesta Y.

Matemáticamente, un problema de agrupamiento se formaliza como sigue: Dados n datos de la muestra $\{X_i\}_{i=1}^n$ y un número positivo k (con $k \leq n$), se construyen $S_1, S_2, \ldots, S_k$ subconjuntos de tal forma que:

$$S_1 \cup S_2 \cup \cdots \cup S_k = \{X_i\}_{i=1}^n \tag{1.1}$$

.

La intersección de dichos subconjuntos, es dos a dos, vacía, o lo que es lo mismo, que cada dato pertenece a uno y solo uno de los k subconjuntos creados:

$\forall i \in \{1, \ldots, n\}$, $\exists j \in \{1, \ldots, k\}$ tal que $X_i \in S_j$ y $X_i \notin S_s$ si $s \neq j$ con $s \in \{1, \ldots, k\}$.

1.2. Agrupamiento paramétrico Vs no paramétrico

El problema del agrupamiento puede ser visto tanto desde la perspectiva de la estadística paramétrica, como de la no paramétrica. En la literatura es frecuente encontrar ambos enfoques (Roberts, 1997). Sin embargo, estos dos paradigmas de la estadística tienen características bien diferenciadas:

- En la Estadística paramétrica, dada una variable aleatoria X, se asume que esta se encuentra caracterizada por una distribución que depende de un número finito de parámetros:

  $$X \approx F_{\theta \in \Theta} \tag{1.2}$$

  siendo Θ un espacio de parámetros de dimensión finita.

  Este paradigma tiene como ventajas que si el modelo está correctamente especificado, las tasas de convergencia de los estimadores son más rápidas, y por tanto se necesita una cantidad menor de datos en las tareas de inferencia. Además en muchos casos pueden lograrse resultados respecto a la optimalidad del procedimiento estadístico empleado como es el caso en problemas de bondad

de ajuste con el lema de Neyman Pearson. Sin embargo, los modelos no son lo suficiente flexibles para modelizar algunas relaciones complejas entre los datos.

- La estadística no paramétrica por el contrario, no asume restricciones fuertes acerca del mecanismo generador de los datos (generalmente se restringen a la continuidad o al carácter Lipchitziano de la variable aleatoria (Tsybakov, 2009). Este paradigma posibilita ajustar modelos altamente flexibles. A pesar de esta ventaja, las tasas de convergencia son muy lentas, siendo en muchos casos necesaria una gran cantidad de datos para mantener la significación estadística, como ocurre de manera especialmente notable en los espacios de alta dimensión, donde dichas técnicas no son adecuadas. En esta situación, para su utilización, se deben combinar con técnicas de reducción de la dimensión o selección de variables.

1.3. Diferentes tipos de grupos

Sobre la definición de grupo no existe un consenso pleno entre la comunidad científica, y como consecuencia de ello, es necesario saber si los grupos obtenidos son realmente satisfactorios o no. De hecho, para algunos, el análisis de grupos es en realidad más un arte que una ciencia (Von Luxburg, 2012). Algunas de las definiciones encontradas en la literatura son las siguientes:

- Basados en la separación entre grupos: Los miembros de cada grupo son más cercanos o similares a cualquier punto del grupo, que a cualquier punto perteneciente a otro grupo.

- Basados en centros: Un punto de un grupo es más cercano o similar al centro de su grupo que al centro de cualquier otro grupo.

- Basado en contigüidad: Un punto del grupo es más cercano o similar a uno, o a más puntos de ese grupo que a cualquier otro punto.

- Basado en densidad: Un grupo es una de región densa donde concentra a diversos datos. Los puntos agrupados en ese grupo comparten una propiedad común y que generalmente es especificada por una distancia.

1.4. Tipos de algoritmos

En diversos trabajos como los de (Xu, 2005) y (Firdaus, 2015) se han intentado agrupar todos los algoritmos de agrupamiento. (Friedman, 2001) distingue tres categorías:

- Algoritmos combinatorios: Consideran los datos observados sin reparar explícitamente en la distribución de probabilidad subyacente.

- Modelos de mixtura: Consideran que la muestra, está formada por datos independientes e idénticamente distribuidos de una población descrita por una función de densidad de probabilidad. La función está caracterizada por un modelo parámetrico compuesto por una mixtura de funciones de densidad, representando cada función de densidad, el mecanismo generador de cada grupo.

- Buscadores de modas: Desde una perspectiva de la estadística no paramétrica, se intentan identificar las distintas modas que se encuentran en el conjunto de datos. De esta forma, se establece una regla que asigna a que grupo pertenece cada dato en base a la cercanía a las distintas modas.

En este trabajo de fin de máster se han considerado 15 de los algoritmos más populares de agrupamiento. En la tabla 1 se muestran los distintos algoritmos contemplados.

1.5. Métodos de reducción de la dimensionalidad

Fue (Bellman,1961) quien acuñó el término "maldición de la dimensionalidad". Con éste se establece que, al incrementar la dimensionalidad el número de datos necesarios, para mantener la significancia estadística, crece de manera exponencial con la dimensión. Debido a ello, en la práctica, bajo estas condiciones, en todo análisis estadístico han de aplicarse técnicas de reducción de la dimensionalidad o de selección de variables. En el análisis de grupos, las técnicas espectrales que traen incorporado procedimientos de reducción de la dimensionalidad, u otras basadas en redes neuronales como es el caso del algoritmo SOM son muy populares.

1.6. Aplicaciones del análisis de grupos

Se enumeran a continuación, algunas de las aplicaciones de diversas técnicas del análisis de grupos:

- El aprendizaje estadístico semisupervisado es aquel que compagina los problemas de clasificación estadística junto con el análisis de grupos. Algunas referencias básicas pueden ser (Cholaquidis, 2018), o (Zhu, 2006).
- Filtrado colaborativo: El agrupamiento proporciona un resumen de usuarios con una mentalidad parecida. Las calificaciones dadas por cada individuo para los demás son usadas para realizar dicho proceso, obteniéndose así sistemas de recomendación como los empleados por (Herlocker, 2004) o (Ekstrand, 2011).
- Segmentación de clientes para agrupamiento de atributos arbitrarios: Se establece como objetivo para adecuar las características de los productos, a las necesidades de los clientes, utilizándose en el Marketing, la segmentación de mercados, el diseño de producto y la toma de decisiones sobre la tecnología de fabricación. (Punj, 1983) realiza un importante estudio de la aplicabilidad de esta técnica al Marketing, mientras que (Wedel, 2012) elabora un estudio general de los diversos métodos.
- Minería de datos: Extrae conocimiento de grandes bases de datos ya sean relacionales o no estructuradas, como la información de las páginas web. Puede detectar tendencias y patrones, así como la detección de datos atípicos (Ro, 2015).
- Resumen de datos y reducción de la dimensionalidad: El primero permite crear representaciones de datos compactos que se procesan e interpretan más fácilmente. El segundo elimina datos irrelevantes-ruido- y atributos redundantes. La extracción de características proyecta los datos en un espacio de dimensionalidad más reducida, como ejemplos están el Análisis de Componentes Principales (Jolliffe, 2011), la Descomposición del Valor Singular como en (Wall, 2003), y el Análisis Lineal Discriminante como en (Ye, 2005). La selección de características selecciona un subconjunto reducido de aquellas, que minimiza la redundancia y maximiza la relevancia del atributo. Destacan entre otros la divergencia de Kullback-Leibler (Hershey, 2007) y el Lasso empleado por (Yamada, 2016).
- Detección de tendencias dinámicas para datos disponibles en streaming: Con la creciente utilización de toma de datos en tiempo real a través de sensores, cámaras y datos tomados de internet, pueden detectarse tendencias que se produzcan en el tráfico o climatología, por ejemplo. El algoritmo ha de ser adaptable y de fácil escaneo. Tanto (Guha, 2016), como (Luhr, 2009) emplean en sus trabajos esta técnica.
- Análisis de datos multimedia: Las imágenes de resonancia magnética permiten conocer la estructura interna de objetos y organismos biológicos. (Swanson, 1998) determina la existencia de segmentos similares en las imágenes y realiza la segmentación de las mismas. Esto puede hacerse, por ejemplo mediante un agrupamiento jerárquico. También pueden emplearse las k medias, o también algoritmos borrosos-fuzzy-, como en el caso de (Clark, 1994).

- Agrupamiento de documentos, basado en su contenido: Se utiliza en el almacenamiento y búsqueda automática de documentos. Separa los documentos en grupos que reflejan el contenido de cada documento. (Berry, 2004) indica los algoritmos más empleados, como el de términos frecuentes. Estos algoritmos, han de ser computacionalmente eficientes, dado el elevado volumen de datos manejado. En (Blei, 2003) por ejemplo, desarrollan una técnica de gran utilidad para el agrupamiento y análisis de texto.

- Análisis de datos biológicos: Se ha aplicado en la secuenciación del genoma humano. (Libbrecht, 2015) recoge la tecnología desarrollada para los datos de expresión genética extraídos del ADN. También se han agrupado secuencias lineales de genes y proteínas con niveles similares de expresiones, estableciendo taxonomías biológicas. En (Madeira, 2004) por ejemplo, se aplican técnicas de biclustering a datos genéticos.

- Análisis de las redes sociales: Se utiliza la estructura de la red social para determinar las comunidades importantes de la red subyacente. Esto permite una buena comprensión de la estructura de la comunidad en la red. El agrupamiento también permite resumir las características de la red social. Algunas referencias clásicas de este tipo de análisis, pueden ser (Hoff, 2002), (Paltoglou, 2012), o (Becker, 2010).

Las aplicaciones mencionadas, entre otras, nos dan idea de la diversidad de disciplinas que plantean problemas resolubles mediante algoritmos de agrupamiento.

Tabla de los métodos de clustering tratados en el TFM con sus características generales

Familia	Algoritmo	Complejidad	K	Cran.r	referencia
Jerárquica					
hclust	Fast HAC	$O(n^2)$	no	fastcluster	(Mullner, 2013)
hclust.vector	Fast HAC (datos vectoriales)	$O(n \times d)$	no	fastcluster	(Mullner, 2013)
energy.hclust	Ward con energy	100 veces + veloz que hclust	no	energy	(Szekely, 2017)
No paramétrico					
Mclust	Algoritmo EM	$n \times d \times i$	no	mclust	(Fraley, 2006)
pdfCluster	Clustering vía ENPD	$O(n \times k)$según estructura	no	pdfCluster	(Azzalini, 2007)
Particional					
kmeans	K-medias	$O(n(^{(dk+1)} \times logn))$	si	stats	(Hartigan, 1979)
kGmedian	Fast K-medianas	$O(n^2 \times k \times i)$	si	Gmedian	(Dohan, 2015)
kcca	K-medoides	$O(k \times (n-k)^2)$	si	flexclust	(Park, 2009)
kproto	K-prototipos	$O(i \times k \times n \times d)$	si	clusMixType	(Huang, 1998)
kgroups	K-grupos	$O(n^2)$	si	energy	(Li, 2015)
Espectral					
speccalt	Espectral	$O(n^3)$	no	speccalt	(Von Luxburg, 2007)
Fuzzy					
FKM	Fuzzy k-medias	$O(n \times d \times k^2 \times i)$	si	fclust	(Bezdek, 1981)
Basado en Densidad					
dbscan	DBSCAN	$O(n^2)$, espacial $O(n \times logn)$	no	fpc	(Ester, 1996)
Clustering de Subespacio					
clique	Algoritmo CLIQUE	$O(3^{3/n})$	no	subspace	(Agrawal, 1998)
ewkm	Algoritmo EWKM	$O(n)$	si	wskm	(Cui, 2012)

Observación 1: En la tabla, la complejidad computacional se calcula considerando las siguientes variables:

- n: número de datos.
- d: número de atributos.
- k: número de clusters.
- i: número de iteraciones.

- La complejidad del algoritmo CLIQUE está calculada para un grafo de n vértices, en su caso menos favorable.

Cuadro 1.1: Algoritmos de agrupamiento seleccionados, implementados en R

Librería Cran.r	Código en R
fastcluster	hclust(dist(USArrests), .ªve")
	hclust.vector(USArrests, çen")
energy	energy.hclust(dist(USArrests))
	kgroups(as.matrix(iris[,1:4]), k = 3, iter.max = 5, nstart = 2)
mclust	Mclust(iris[,1:4])
pdfcluster	pdfCluster(wine[, c(2,5,8)])
stats	kmeans(iris[,1:4], 3))
Gmedian	kGmedian(x)
flexclust	kcca(Nclus, k=4)
clusMixType	kproto(x, 4)
speccalt	speccalt(local.rbfdot(synth2))
fclust	FKM(Mc[,1:(ncol(Mc)-1)],k=6,m=1.5,stand=1)
fpc	dbscan(iris, eps = .5, minPts = 5)
wskm	ewkm(iris[1:4], 3, lambda=0.5, maxiter=100)

Capítulo 2

REVISIÓN DE ALGUNOS ALGORITMOS IMPLEMENTADOS EN R.

En este capítulo se va a realizar una revisión bibliográfica de algunos de los algoritmos más populares de agrupamiento pertenecientes a algunas de las familias principales. En primer lugar se definirá formalmente lo que se entiende por distancia y medida de similitud para pasar a continuación a explicar los distintos algoritmos tratados.

2.1. Las distancias

2.1.1. Introducción

Dados n datos $X_1, X_2 \ldots, X_n$, pertenecientes a un conjunto X, las técnicas de agrupamiento, como ya se ha indicado, consisten en agrupar los n datos en k grupos diferenciados, sin disponer de información previa acerca de la composición de los grupos.

Un aspecto central para realizar lo anterior, es definir una distancia o una medida de similitud entre los objetos, para poder establecer el grado de igualdad existente entre ellos.

Desde un punto de vista formal, para un conjunto X, se define una distancia para cada par de elementos en X, como una función continua $d : X \times X \to \mathbb{R}^+$ que verifique las siguientes condiciones:

- $d(a, b) \geq 0 \ \forall a, b \in X$ y $\forall a \in X : d(a, a) = 0$ (no negatividad).
- $d(a, b) = d(b, a) \ \forall a, b \in X$ (simetría).
- $d(a, b) \leq d(a, c) + d(c, b) \ \forall \ a, b, c \in X$ (desigualdad triangular).

Una medida de similitud se define también como una función continua $s : X \times X \to \mathbb{R}$, pero que cumple las siguientes propiedades:

- $s(a, a) = s_0 \ \forall a \in X$.
- $s(a, b) \leq s_0 \ \forall a, b \in X$.
- $s(a, b) = s(b, a)$ (simetría).

Donde s_0 es un número real finito.

A continuación vamos a explicar algunas de las principales distancias y medidas de similitud usadas en la literatura, y que fueron extraídas de (Xu, 2005).

Distancia de Minkowski

La distancia de Minkowski es una familia de distancias que dependen de un parámetro $r \in \mathbb{R}^+$ que extienden a la distancia euclidiana (caso $r = 2$). Dados cualesquiera, x_i y x_j de $\mathbb{R}^p$, se define la distancia de Minkowski como sigue:

$$\parallel x_i - x_j \parallel_r = (\sum_{k=1}^{n} |x_{ik} - x_{jk}|^r)^{1/r}. \tag{2.1}$$

La similitud de coseno

Uno de los inconvenientes de la distancia de Minkowski es su dependencia de la escala de cada variable, o que el rango de valores que puede tomar no es fijo, lo que impide muchas veces comparar los elementos de X de manera objetiva. A su vez, en muchas situaciones reales como es el caso del análisis de sentimientos, es necesario establecer si la relación entre dos frases es positiva o negativa, y por tanto tiene más sentido usar una medida de similaridad como puede ser la del coseno.

La medida de similitud del coseno toma valores entre $[-1, 1]$, y simplemente consistente en calcular el producto escalar de dos vectores, y normalizar su valor, usando el producto del módulo de cada uno de ellos. Formalmente, se define $\forall x_i, x_j \in R^p$ mediante:

$$s(x_i, x_j) = \frac{x_i^T . x_j}{\|x_i\| . \|x_j\|}. \tag{2.2}$$

La medida de similitud del coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson, es la medida usada tradicionalmente en estadística para calcular el grado de dependencia entre dos variables. Su principal limitación consiste en que únicamente es capaz de capturar relaciones lineales entre variables. Dados dos vectores x_i y x_j de $\mathbb{R}^p$ se define la medida de similitud por:

$$d(x_i, x_j) = \frac{(x_i - \overline{x}_i)^T . (x_j - \overline{x}_j)}{\parallel x_i - \overline{x}_i \parallel . \parallel x_j - \overline{x}_j \parallel} \tag{2.3}$$

Donde $\overline{x}_i = \frac{1}{p} \sum_{i=1}^{p} x_i$ denota el valor medio del vector x_i. Utilizando la desigualdad de Cauchy-Bunyakovsky-Schwarz, se puede comprobar trivialmente que la expresión anterior siempre toma valores entre $[-1, 1]$.

Otras medidas: Similitud de Jaccard extendida y de Dice

En la literatura han aparecido muchas otras medidas de similitud, como es el caso de la distancia de Jaccard o Dice. $\forall x_i, x_j \in R^p$ la primera se define como sigue:

$$s(x_i, x_j) = \frac{x_i^T x_j}{||x_i|| + ||x_j|| - x_i^T x_j}, \tag{2.4}$$

mientras que la segunda:

$$s(x_i, x_j) = \frac{2 . x_i^T . x_j}{\parallel x_i \parallel^2 + \parallel x_j \parallel^2}. \tag{2.5}$$

Elección de la medida de similaridad adecuada

En base a los objetivos que se busque lograr mediante el análisis de grupos, así como las características de los datos analizados se usará una medida u otra. De hecho, no existen metodologías específicas para establecer la distancia a utilizar, y en la práctica únicamente se podrán aplicar reglas generales en la elección de la distancia.

La distancia euclídea es la más usada en la práctica. Sin embargo, es muy sensible al ruido, y por ello, su uso no es muy recomendable en contextos de alta dimensionalidad.

Las medidas de similaridad, tienen la ventaja de que permiten introducir cierta interpretabilidad a un problema concreto, como ya se ha mencionado para la distancia de coseno en el caso del análisis de sentimientos.

2.2. Agrupamiento jerárquico

En los métodos de agrupamiento jerárquicos, los grupos, son formados iterativamente de una manera aglomerativa o divisiva. Esto quiere decir que en el primer caso, a partir de los n datos de la muestra $\{X_i\}_{i=1}^n$ se forman inicialmente n grupos individuales, y en cada iteración posterior, el algoritmo crea nuevos grupos, fusionando aquellos que presenten la menor distancia intergrupos, bajo la restricción de que todos los elementos pertenecientes a un grupo en el paso n del algoritmo, se encuentran juntos en todos los nuevo grupos creado posteriormente. En el segundo caso, por el contrario, los n datos de la muestra $\{X_i\}_{i=1}^n$ se encuentran inicialmente todos juntos en un único grupo, y el algoritmo en cada nueva iteración, crea los nuevos grupos manteniendo una estructura de jerarquía de la siguiente manera: Se crean los nuevos grupos de tal forma que se minimice o maximice (dependiendo del algoritmo) una distancia entre los grupos ya formados.

Los resultados obtenidos tras aplicar un algoritmo de agrupamiento jerárquico suelen representarse a través de un dendograma.

2.2.1. Agrupamiento de enlace simple

Este tipo de agrupamiento es llamado habitualmente como el método del vecino más próximo. La unión de grupos se realiza en base a la distancia de dos elementos pertenecientes a dos grupos diferentes, y que al mismo tiempo, sean los más cercanos entre si. Este algoritmo crea los nuevos grupos, en cada iteración, utilizando la siguiente distancia:

$$D(A,B) = \text{mín}\{d(a,b) : a \in A, b \in B\} \tag{2.6}$$

donde A y B son dos grupos diferentes y d es una distancia fijada de antemano.

2.2.2. Agrupamiento de enlace completo

Es también denominado como el método del vecino más lejano. La unión de dos grupos se realiza usando la mayor distancia existente entre dos elementos pertenecientes a dos grupos diferentes. La distancia se define como sigue:

$$D(A,B) = max\{d(a,b) : a \in A, b \in B\} \tag{2.7}$$

2.2.3. Agrupamiento de enlace promedio

Es conocido también como el método de varianza mínima. Para formar los nuevos grupos, este método utiliza en la formación de los dos nuevos grupos la distancia media entre todos los elementos pertenecientes a ambos grupos. Dicha distancia se expresa como sigue:

$$\frac{1}{\|A\|\|B\|} \sum_{a \in A} \sum_{b \in B} d(a,b) \tag{2.8}$$

2.2.4. Algoritmo de agrupamiento jerárquico basado en la distancia energía

Un enfoque novedoso para realizar un análisis de agrupamiento jerárquico es usar como función de distancia, la distancia energía. A continuación se define la distancia de energía.

2.2.5. Distancia de energía

Sean $A = \{a_1, ..., a_{n_1}\}$ y $B = \{b_1, ..., b_{n_2}\}$ dos conjuntos de vectores no vacíos de $\mathbb{R}^d$ de tamaño n_1 y n_2 respectivamente. Se define la distancia de energía entre A y B mediante:

$$e(A,B) = \frac{n_1 n_2}{n_1 + n_2} \left(\frac{2}{n_1 n_2} \Sigma_{i=1}^{n_1} \Sigma_{j=1}^{n_2} \|a_i - b_j\| - \frac{1}{n_1^2} \Sigma_{i=1}^{n_1} \Sigma_{j=1}^{n_1} \|a_i - a_j\| - \frac{1}{n_2^2} \Sigma_{i=1}^{n_2} \Sigma_{j=1}^{n_2} \|b_i - b_j\| \right) \tag{2.9}$$

donde $||\cdot||$ denota la distancia euclideana, o cualquier potencia suya con $\alpha \in (0, 2]$. Si la potencia es $\alpha = 2$ el algoritmo de agrupamiento explicado a continuación coincidirá con el método jerárquico de clustering de Ward.

Propiedades de la distancia de energía

Teorema 1. *Supongamos $X, X' \in \mathbb{R}^d$ son vectores aleatorios independientes e idénticamente distribuidas según una distribución F e $Y, Y' \in \mathbb{R}^d$ bajo una distribución G. Asumamos además que los momentos de orden 1 son finitos, $E\|X\| < \infty$ y $E\|Y\| < \infty$.*

Entonces:

$$2E\|X - Y\| - E\|X - X'\| - E\|Y - Y'\| \geq 0 \tag{2.10}$$

y se cumple la igualdad si y solo si X e Y siguen la misma distribución.

Se muestra a continuación una demostración elemental para el caso particular de $d = 1$. Esta demostración no es válida para un contexto general con $d > 1$, siendo esta, de carácter más técnico.

Demostración.

Si $d = 1$ se verifica que:

$$\begin{aligned} & 2\int_{\mathbb{R}} (F(t) - G(t))^2 dt \\ &= 2\int_{\mathbb{R}} [F(t)(1 - G(t)) + (1 - F(t)G(t)) - F(t)(1 - F(t)) - G(t)(1 - G(t))]dt \\ &= 2\int_{R} [P(X \leq t < Y) + P(Y \leq t < X) - P(X \leq t < X') - P(Y \leq t < Y')]dt \\ &= 2E|X - Y| - E|X - X'| - E|Y - Y'|. \end{aligned}$$

Para completar la demostración para el caso $d > 1$, se puede hacer uso del teorema 2 que se enunciará más adelante, tomando $\alpha = 1$.

El agrupamiento que se presenta en esta sección busca minimizar la distancia entre las observaciones en cada nuevo grupo formado. El siguiente corolario justifica dicho procedimiento.

Corolario 1. *Con la notación anterior, $A, B \subset \mathbb{R}^d, e(A, B) \geq 0$ y la igualdad se cumple si y solo si $A = B$.*

Demostración:

Se supone que $A = \{a_1, ..., a_{n_1}\}$ y $B = \{b_1, ..., b_{n_2}\}$ son subconjuntos no vacíos y finitos de $\mathbb{R}^d$. Sean X, X' y Y, Y' vectores independientes e idénticamente distribuidos entre si. Cada elemento del conjunto A sigue la distribución del vector aleatorio X con todas los elementos independientes entre si, mientras que las componentes del vector B están distribuidas según el vector Y, y siendo además, indendientes entre si.

Teniendo en cuenta la afirmación anterior, se verifica:

$E\|X - Y\| = \Sigma_{i=1}^{n_1}\Sigma_{j=1}^{n_2}\|a_i - b_j\|/(n_1 n_2), E\|X - X\prime\| = \Sigma_{i=1}^{n_1}\Sigma_{j=1}^{n_1}\|a_i - b_j\|/n_1^2$, y $E\|Y - Y'\| = \Sigma_{i=1}^{n_2}\|b_i - b_j\|/n_2^2$. Por lo tanto,

$$\frac{n_1 n_2}{n_1 + n_2}[2E\|X - Y\| - E\|X - X'\| - E\|Y - Y'\|] = e(A, B). \tag{2.11}$$

Por el Teorema 1, $e(A, B) \geq 0$. Además. $e(A, B) = 0$, si y solamente si $X \overset{d}{=} Y$, y esto ocurre, si y solo si $A = B$.

El algoritmo de agrupamiento jerárquico aplicado a n objetos tiene inicialmente n grupos individuales. Al corresponderse los valores pequeños de la distancia energía a grupos homogéneos, en cada paso se calcula dicha distancia entre los pares de grupos y se selecciona el par que tenga una distancia energía mínima como el par óptimo para fusionar. Tras fusionar el par óptimo de los grupos, las distancias energía entre grupos se actualizan. La altura $h(k)$ del correspondiente nodo en el dendograma es la distancia energía entre los dos grupos fusionados en el paso k. Esta implementación sigue, en esencia, el algoritmo jerárquico general, indicado por (Anderberg, 1973) y (Hartigan, 1975).

Para vectores aleatorios $X \in R^d$ y $Y \in R^d$ y α constante, de modo que $E\|X\|^\alpha < \infty$ y $E\|Y\|^\alpha < \infty$, define la función de valores reales

$$\varepsilon^\alpha(X, Y) = 2E\|X - Y\|^\alpha - E\|X - X\prime\|^\alpha - E\|Y - Y\prime\|^\alpha \tag{2.12}$$

donde $X, X\prime$ e $Y, Y\prime$ son i.i.d. En caso de que ε aparezca sin α superíndice, el exponente será $\alpha = 1$. Así, el teorema 1 establece que $\varepsilon(X, Y)$ es siempre no negativo y que $\varepsilon(X, Y) = 0$ si y solo si $X \overset{D}{=} Y$.

Teorema 2. *Sean X, X' vectores aleatorios independientes e idénticamente distribuidos en $\mathbb{R}^d$, y sean Y, Y' vectores aleatorios en $\mathbb{R}^d$ independientes en Y. Si α es una constante tal que $E\|X\|^\alpha < \infty$, y $E\|Y\|^\alpha < \infty$, entonces se verifican las siguientes afirmaciones:*

Si $0 < \alpha \leq 2$, entonces $e^{(\alpha)}(X, Y) \geq 0$

Si $0 < \alpha < 2$, entonces $e^{(\alpha)}(X, Y) = 0$ si y solo si $X \overset{D}{=} Y$

si $\alpha = 2$, entonces $e^{\alpha}(X, Y) = 0$ si y solo si $E[X] = E[Y]$.

Observación 2: Para $\alpha > 2$, la desigualdad $\varepsilon^{(\alpha)}(X, Y) \geq 0$ no se cumple siempre. Sea por ejemplo $X = 0$ con probabilidad 1 e $Y = \pm 1$ con probabilidad $1/2$. Entonces $\varepsilon^{(\alpha)}(X, Y) = 2 - 2^{(\alpha)}(Y, Y) = 2 - 2^{\alpha}/2$, que es negativa si $\alpha > 2$. Para $\alpha < 0$ si X toma un valor cualquiera, con probabilidad positiva, entonces $E\|X - X\prime\|^\alpha = \infty$. Se ve así la motivación para que $0 < \alpha < 2$. En este intervalo $\alpha = 1$ es el único número, este es el caso más sencillo, que hemos aplicado al agrupamiento, y que corresponde al teorema 1.

Dicho teorema, puede consultarse en los trabajos de (Szekely, 2005)

Consistencia estadística

Sea $A = \{X_1, ..., X_{n_1}\}$ y $B = \{Y_1, ..., Y_{n_2}\}$ muestras aleatorias, independientes de $\mathbb{R}^d$ formadas por vectores aleatorios X, Y respectivamente. Supongamos que $E\|X\| < \infty, E\|Y\| < \infty$. Se definen las distancias $\mu_{AB} = E\|X - Y\|$,$\mu_A = E\|X_1 - X_2\|, \mu_B = E\|Y_1 - Y_2\|$, y defínase $e(A, B)$. Ahora $e(A, B)$ es un vector aleatorio con valor esperado

$$E[e(A,B)] = \frac{n_1 n_2}{n_1+n_2}(2\mu_{AB} - \frac{n_1-1}{n_1}\mu_A - \frac{n_2-1}{n_2}\mu_B)) = \frac{n_1 n_2}{n_1+n_2}(2\mu_{AB} - \mu_A - \mu_B) + \frac{n_2\mu_A}{n_1+n_2} + \frac{n_1\mu_B}{n_1+n_2} \tag{2.13}$$

Si X e Y están idénticamente distribuídas $\mu_{AB} = \mu_A = \mu_B$, que implica que $2\mu_{AB} - \mu_A - \mu_B = 0$ y

$$E[e(A, B)] = \frac{n_2\mu_A + n_1\mu_B}{n_1 + n_2}$$

para todos los valores positivos $n_1 y n_2$ Si X e Y no están idénticamente distribuidas, $2\mu_{AB} + \mu_A - \mu_B$ equivale a una constante positiva, por el teorema 1. Por tanto, si $X \overset{D}{\neq} Y$, y $n = n_1 + n_2$, la distancia esperada entre A y B tiende a ser una constante positiva, lo que expresado de otro modo dice que $E[e(A, B)]$ tiende a infinito.

2.3. Algoritmos particionales

2.3.1. Algoritmo k medias

El algoritmo k medias trata de minimizar el criterio clásico de distancia de cada una de las observaciones de la muestra a la media de su grupo al cuadrado. Dicho algoritmo presenta una serie de extensiones y generalizaciones entre las que destacan el caso borroso- o fuzzy-, el de máxima verosimilitud, y aquellos criterios basados en convexidad, entre otros.

Con la publicación de (Sokal, 1963), el análisis cluster surge como una importante área de estudio. Ello supuso el inicio de la investigación, a nivel internacional, sobre las diferentes técnicas disponibles. Con (Lerman, 1970) se inicia la publicación de un compendio de libros y artículos que sentaron las bases de esta área de estudio, destacan cronológicamente (Jardine, 1971), que profundiza en la naturaleza matemática del problema del agrupamiento, (Anderberg, 1973) con un enfoque más orientado hacia las aplicaciones, (Bijnen, 1973) que realiza un primera recopilación de los métodos disponibles. (Bock, 1974) elabora un estudio de los métodos enfocado al análisis de datos y las posibilidades que se abren mediante el conocimiento de la estructura de los datos. (Sodeur, 2013) trata acerca los métodos empíricos necesarios para la realizar tareas de clasificación, mientras que (Vogel, 1975) está más orientado a la resolución de problemas numéricos de clasificación. La publicación de (Hartigan, 1975) tiene una gran repercusión en el mundo académico por su estudio de aspectos computacionales de los algoritmos, al igual que (Spath, 1980). Como resultado de ello, los problemas y métodos básicos del agrupamiento se hicieron ampliamente conocidos por la comunidad científica, en la Estadística y el Análisis de datos, así como, en particular, en sus aplicaciones a otros campos de la ciencia.

Uno de los principales enfoques de las técnicas de agrupamiento se basa en el criterio de la suma de los cuadrados de la varianza, y en el algoritmo que, a día de hoy se denomina k-medias.

Al realizar una revisión de dicho algoritmo desde sus orígenes, se observa que fue propuesto por diversos investigadores, bajo diversas formas e hipótesis. Posteriormente muchos otros autores analizaron sus aspectos

teóricos, algorítmicos, así como modificaciones del método. Ejemplos de ello se encuentran al considerar la sucesión de analogías al criterio de la suma de cuadrados planteado por diversos científicos, con las aportaciones de (Cox, 1957), (Fisher, 1958), (Engelman, 1969); investigando el comportamiento asintótico bajo estrategias de muestreo aleatorio están las aportaciones de (Hartigan, 1975) (Pollard, 1982) y (Bock, 1985) mediante la extensión de su dominio de aplicación a diferentes tipos de datos y modelos probabilísticos. Posteriormente, la monografía (Diday, 1979) escrita por dicho autor junto a otros 22 coautores, marcó un considerable nivel de generalización de la idea fundamental y estableció su uso para los modelos de agrupamiento, basados en distribución.

El criterio de la suma de cuadrados para el agrupamiento de datos

Sean n datos $X_1, ..., X_n$ en $\mathbb{R}^p$, y una partición de tamaño k, $C = (C_1, ..., C_k)$ del conjunto O de todas las particiones de tamaño k posibles con el conjunto $\{X_1, ..., X_n\}$.

El criterio de suma de cuadrados para el caso de objetos subyacentes; también denominado criterio de la varianza, inercia, o criterio de la traza, viene dado mediante:

$$g_n(C) := \sum_{i=1}^{k} \sum_{\ell \in C_i} \|x_\ell - \overline{x}_{c_i}\|^2 \longrightarrow \min_{C} \tag{2.14}$$

donde $\overline{x}_{c_i}$ representa al centroide de los datos puntuales pertenecientes a la clase C_i, con $\ell \in C_i$. Se busca una partición C de O, con valor mínimo del criterio $g_n(C)$.

El problema de optimización con un parámetro-ver ecuación anterior- se relaciona e incluso es equivalente al problema de optimización de 2 parámetros

$$g_n(C, Z) := \sum_{i=1}^{k} \sum_{\ell \in C_i} \|x_\ell - z_i\|^2 \longrightarrow \min_{C,Z} \tag{2.15}$$

donde la minimización es también respecto al conjunto de centroides de cada grupo $Z = (z_1, ..., z_k)$.

Para la minimización de la función objetivo subyacente se han desarrollado una amplia variedad de métodos por diversos autores en los que se han usado desde técnicas de optimización combinatoria, métodos de optimización exactos o algoritmos heurísticos.

El algoritmo k medias busca aproximar una partición k óptima, mediante la iteración de los pasos de minimización parcial (i) e (ii) que se definen a continuación. Su algoritmo es el siguiente:

- Paso $t = 0$ Comienza con un sistema de centroides arbitrarios para cada grupo $Z^0 = (z_1^{(0)}, ..., z_k^{(0)})$
- Paso $t \Rightarrow t + 1$:

  [(i)] Minimiza el criterio $g_n(C, Z^{(t)})$ con respecto a la partición C que,-por ejemplo- determina una partición de distancia mínima $C^{(t+1)} := C(Z^t)$

  [(ii)] Minimiza el criterio $g_n(C^{(t+1)}, Z)$ con respecto al sistema de prototipos Z, que calcula el sistema de centroides de cada clase $Z^{(t+1)} := Z(C^{(t+1)})$.

El algoritmo se detendrá tras ejecutar los pasos descritos cuando se alcance la estacionalidad.

Por su construcción, este algoritmo genera una secuencia $Z^0, C^1, Z^1, C^2, ...$ de centroides y particiones con valores decrecientes en cada iteración, que convergen a un mínimo que es típicamente local.

Criterio de suma de cuadrados continua para realizar disección del espacio

Dada una variable aleatoria X con medida de probabilidad P y que toma valores en $\mathbb{R}^p$. Desde el punto de vista poblacional se puede expresar el problema de clustering definido anteriormente a nivel muestral como sigue. Se busca una partición $B = (B_1, ..., B_k)$ de R^p que hace mínimo la siguiente expresión.

$$g(B) := \sum_{i=1}^{k} \int_{B_i} \|x - E[X|X \in B_i]\|^2 dP(x) \longrightarrow \min_{B} \tag{2.16}$$

Al igual que antes, el problema de optimización subyacente puede expresarse como uno de dos parámetros.

$$g(B, Z) := \sum_{i=1}^{k} \int_{B_i} \|x - z_i\|^2 dP(x) \longrightarrow \min_{B,Z} \tag{2.17}$$

Teorema 3. *(i) Para cualquier partición B en R^p de k elementos, el criterio $g(B,Z)$ es mínimo en Z, por el sistema de prototipos $Z^* = (z_1^*, ..., z_k^*) = Z(B)$ dado por las esperanzas condicionadas. $z_i^* := E[X|X \in B_i]$ de B_i. Esto es:*

$$g(B,Z) \geq g(B,Z^*) = \sum_{i=1}^{k} \int_{B_i} \|x - E[X|X \in B_i]\|^2 dP(x) = g(B) \ \ \forall Z. \tag{2.18}$$

(ii) Para cualquier sistema de prototipos fijados Z, el criterio g(B,Z) es parcialmente minimizado con respecto a B por cualquier partición de distancia mínima $B_i^* := \{x \in \mathbb{R}^p | d(x,Z_i) \leq d(X,Z_j) \ \ con \ \ i \neq j\}$, esto es:

$$g(B,Z) \geq g(B^*,Z) = \int_{\chi} \min_{j=1,...,k} \{\|x - z_j\|^2\} dP(x) =: g(Z) \ \ \forall B \tag{2.19}$$

El anterior teorema es una extensión del clásico resultado que afirma que la media hace mínimo las desviaciones de una variable aleatoria con la distancia euclídea al cuadrado, pero en este caso para k grupos,y tomando la esperanza condicional a cada grupo.

2.3.2. El algoritmo k grupos

Se trata de un algoritmo de agrupamiento que intenta agrupar los datos en base a sus distribuciones de probabilidad. Utiliza la distancia energía para medir las diferencias entre muestras, siendo este algoritmo, una extensión del algoritmo k medias para distribuciones.

La implementación del algoritmo k grupos se basa parcialmente en el algoritmo k medias de (Hartigan, 1979). El algoritmo se generaliza tomando inicialmente 1 punto, para finalizar tomando m puntos. Para datos univariantes se demuestra que el algoritmo k-medias, según el método mencionado, es un caso especial de k grupos en primera variación. Cuando los grupos están bien separados y normalmente distribuidos, el comportamiento del algoritmo k grupos es similar al algoritmo k medias, para el caso considerado. En caso de que los datos no tengan su primer momento finito, o que los datos tengan un elevado apuntamiento o curtosis los dos métodos k grupos tienen mejor rendimiento que el k medias. Para grupos no esféricos ambos algoritmos de k grupos se comportan mejor que el k medias en alta dimensión, además el k grupos es consistente a medida que la dimensión crece.

El agrupamiento permite agrupar objetos similares, sin el uso de información externa como las etiquetas de clase. Este método permite cumplir el doble objetivo que se plantea el agrupamiento:

Por un lado, hallar grupos de objetos que comparten características similares. Por otra parte resumir los objetos característicos de aquellos objetos pertenecientes a los mismos grupos. Se alcanzará así un resultado comprensible y útil.

El método k grupos generaliza y extiende el método de las k medias. La distancia energía se basa en una caracterización de la igualdad entre distribuciones y se emplea en dimensión arbitraria. Es una generalización del k medias, este último separa los grupos por diferencias en las medias. La distancia entre grupos del k grupos se basa en la distancia energía que separa los grupos, según las diferencias en las distribuciones.

La distancia energía como medida de distancia entre distribuciones

Esta distancia, fue propuesta para un contexto más general para medir el grado de dependencia estadística entre dos variables aleatorias X e Y (Szekely, 2007).

Definición: La distancia energía entre variables aleatorias X e Y independientes y de dimensión d; se define como:

$$\varepsilon(X,Y) = 2E|X - Y|_d - E|X - X'|_d - E|Y - Y\prime|_d \tag{2.20}$$

donde $E|X|_d < \infty$ y $E|Y|_d < \infty$, $X\prime$ es una copia de X iid, e $Y\prime$ es una copia de Y iid.

En lo sucesivo cuando la dimensión ya esté especificada se omitirá su notación. Asimismo se usará la notación $|.|_d$ para denotar la distancia euclídea en $\mathbb{R}^d$.

Sean $F(X)$ y $G(X)$ dos funciones de distribución acumulativas y sean $\hat{f}$ $\hat{g}$ las funciones características de las variables independientes X e Y.

Definición: Sean X e Y variables aleatorias d-dimensionales e independientes. con funciones características $\hat{f}$ y $\hat{g}$ respectivamente y $E|X|^\alpha < \infty$, $E|Y|^\alpha < \infty$ para algún $0 < \alpha < 2$. La distancia de energía entre X e Y se define mediante

$$\varepsilon^\alpha(X,Y) = 2E|X - Y|_d^\alpha - E|X - X'|_d^\alpha - E|Y - Y\prime|_d^\alpha \tag{2.21}$$

$$= \frac{1}{C(d,\alpha)} \int_{R_d} \frac{|\hat{f}(t) - \hat{g}(t)|^2}{|t|_d^{d+\alpha}} dt \tag{2.22}$$

donde $0 < \alpha < 2$, $|.|$ es la norma compleja, y

$$C(d,\alpha) = 2\pi^{\frac{d}{2}} \frac{\Gamma(1-\frac{\alpha}{2})}{\alpha 2^{\alpha} \Gamma(\frac{d+\alpha}{2})} \tag{2.23}$$

El teorema siguiente, establece que la distancia energía entre variables aleatorias caracteriza la igualdad en distribución.

Teorema 4. $\forall \alpha \in (0,2)$
$\varepsilon^{\alpha}(X,Y) = 2E|X-Y|_d^{\alpha} - E|X-X'| - E|Y-Y'| \geq 0$,
Con igualdad a cero $\Leftrightarrow$ *X e Y están idénticamente distribuidos.*
Nótese que cuando $\alpha = 2$, *tenemos que*
$2E|X-Y|^2 - E|X-X'|^2 - E|Y-Y'|^2 = 2|E(X) - E(Y)|^2$
que mide la distancia al cuadrado entre las medias. Por lo tanto, la caracterización de arriba no se cumple para $\alpha = 2$ *desde que se obtiene la igualdad a 0 aun cuando* $E(X) = E(Y)$

El estadístico energía para dos muestras, correspondiente a la distancia energía $\varepsilon^{\alpha}(X,Y)$, para muestras aleatorias independientes $X = \{X_1, X_2, ..., X_{n1}\}$ e $Y = \{Y_1, Y_2, ..., Y_{n2}\}$ es

$$\varepsilon^{\alpha}_{n_1,n_2}(X,Y) = \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{m=1}^{n_2} |X_i - Y_m|^{\alpha} - \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} |X_i - X_j|^{\alpha} - \frac{1}{n_2^2} \sum_{l=1}^{n_2} \sum_{m=1}^{n_2} |Y_l - Y_m|^{\alpha} \tag{2.24}$$

donde $\alpha \in (0,2)$. El estadístico ponderado de dos muestras

$$T_{X,Y} = (\frac{n_1 n_2}{n_1 + n_2}) \varepsilon_{n_1,n_2}(X,Y) \tag{2.25}$$

establece un test consistente para la igualdad de distribuciones de X e Y. El test de energía de muchas muestras, para la igualdad de k distribuciones, $k \geq 2$ es una extensión no paramétrica del análisis de la varianza.

Explicación del algoritmo

El k medias utiliza habitualmente la distancia al cuadrado para calcular la disimilaridad entre el dato y el prototipo preespecificado, minimizando la varianza interna de los grupos. Puede emplearse, como se ha visto, $T_{X,Y}$ que es el estadístico energía ponderado de dos muestras. Esta es la función estadística que mide la disimilaridad entre los grupos, y modifica el algoritmo de k medias considerado. El método presentado pertenece, generalmente a la clase de algoritmos basados en distribución.

Este tipo de algoritmo toma un grupo como una región densa de datos que está rodeada de regiones de bajas densidades. Se emplean a menudo cuando los grupos son irregulares o están entremezclados, o cuando hay presencia de ruido y de atípicos. Como la distancia energía mide la similaridad entre 2 conjuntos, en lugar de la similaridad entre objeto y prototipo, llamamos a este método k grupos.

Definimos la dispersión entre dos conjuntos A y B como,

$$G^{\alpha}(A,B) = \frac{1}{n_1 n_2} \Sigma_{i=1}^{n_1} \Sigma_{m=1}^{n_2} |a_i - b_m|^{\alpha} \tag{2.26}$$

donde $0 \leq \alpha \leq 2$, y n_1, n_2 son los tamaños muestrales para los conjuntos A, B.

Sea $P = \{\pi_1, \pi_2, ..., \pi_k\}$ una partición de las observaciones, donde k es el número de grupos preestablecido. Se define la dispersión total entre todos los grupos como

$$T^{\alpha}(\pi_1, ..., \pi_k) = \frac{N}{2} G^{\alpha}(\cup_{i=1}^{k} \pi_i, \cup_{i=1}^{k} \pi_i) \tag{2.27}$$

Donde N es el número total de observaciones. La dispersión dentro de los grupos está definido por

$$W_{\alpha}(\pi_1, ..., \pi_k) = \sum_{j=1}^{k} \frac{n_j}{2} G^{\alpha}(\pi_j, \pi_j) \tag{2.28}$$

Donde n_j es el tamaño de la muestra para los grupos π_j. La dispersión entre las muestras es:

$$B^{\alpha}(\pi_1, ..., \pi_k) = \sum_{1 \leq i \leq j \leq k} \{\frac{n_i n_j}{2N} (2G^{\alpha}(\pi_i, \pi_j) - G^{\alpha}(\pi_i, \pi_i) - G^{\alpha}(\pi_j, \pi_j))\} \tag{2.29}$$

Entonces $0 < \alpha \leq 2$ tenemos la descomposición:

$$T^{\alpha}(\pi_1, ..., \pi_k) = W^{\alpha}(\pi_1, ..., \pi_k) + B^{\alpha}(\pi_1, ..., \pi_k) \tag{2.30}$$

donde tanto $W^{\alpha}(\pi_1, ..., \pi_k)$, como $B^{\alpha}(\pi_1, ..., \pi_k)$ son no negativas, aplicados a esta descomposición de T^{α} en componentes de la distancia disco para obtener un test no paramétrico consistente de igualdad de k distribuciones. Para maximizar la dispersión entre muestras $B^{\alpha}(\pi_1, ..., \pi_k)$, con $T^{\alpha}(\pi_1, ..., \pi_k)$ constante , es equivalente a minimizar $W^{\alpha}(\pi_1, ..., \pi_k)$. Por lo tanto nuestro propósito es encontrar las mejores particiones que minimicen la dispersión dentro del grupo W^{α}. Es decir, la función objetivo de k grupos es :

$$\min_{\pi_1,...,\pi_k} \sum_{j=1}^{k} \frac{n_j}{2} G^{\alpha}(\pi_j, \pi_j) = \min_{\pi_1,...,\pi_k} W^{\alpha}(\pi_1, ..., \pi_k) \tag{2.31}$$

2.3.3. K prototipos

El algoritmo k prototipos integra a los algoritmos k medias y k modas para el tratamiento de datos de tipo mixto. En la práctica es un algoritmo de mayor utilidad puesto que los datos recogidos en el mundo real son observaciones de tipo mixto. Asumiendo un conjunto de n objetos, $X = \{X_1, X_2, ..., X_n\}$. $X_i = \{X_{i1}, X_{i2}, ..., X_{im}\}$ está formado por m atributos, donde m_r son atributos numéricos, y m_c son atributos categóricos, con $m = m_r + m_c$.

Objetivo

El objetivo del agrupamiento es dividir n objetos en k grupos distintos $C = \{C_1, C_2, ..., C_k\}$, donde C_i es un centro del grupo i-ésimo. La distancia $d(X_i, C_j)$ entre X_i y C_j se calcula como

$$d(X_i, C_j) = d_r(X_i, C_j) + \gamma d_c(X_i, C_j) \tag{2.32}$$

donde $d_r(X_i, C_j)$ es la distancia entre los atributos numéricos, y $d_c(X_i, C_j)$ es la distancia entre atributos categóricos, y γ es el peso de los atributos categóricos.

$$d_r(X_i, C_j) = \Sigma_{i=1}^{p} |x_{il} - c_{jl}|^2, \tag{2.33}$$

$$d_c(X_i, C_j) = \Sigma_{i=1}^{p} \delta(x_{il}, c_{jl}), \tag{2.34}$$

$$\delta(x_{il}, c_{jl}) = \begin{cases} 0, & \text{cuando } x_{il} = c_{jl} \\ 1, & \text{cuando } x_{il} \neq c_{jl} \end{cases}$$

2.3.4. Fuzzy C medias

Introducción

El algoritmo k medias es un algoritmo de agrupamiento rápido, robusto y sencillo. Presenta buenos resultados con clusters bien separados. Es además relativamente eficiente respecto a la complejidad de tiempo computacional. No obstante, los problemas que el k medias suele tener con determinadas bases de datos, son el de no encontrar grupos superpuestos, ser sensible al ruido y no poder agrupar correctamente grupos no separables linealmente.

En este contexto, (Bezdek, 1981) presenta el algoritmo Fuzzy C means(FCM), basándose en el estudio previo de (Dunn, 1973), como una extensión del k medias. (Imdad, 2008) y (Suganya, 2012), identifican una docena de algoritmos desarrollados para mejorar la eficiencia y precisión del FCM. no obstante su versión básica ha sido ampliamente utilizada, tanto en Ingeniería como en Economía. Se trata de un algoritmo suave que agrupa datos borrosos, de modo que un dato no es miembro de forma estricta de un grupo concreto, sino que es miembro de muchos grupos diferentes en un grado de pertenencia diferente para cada uno de ellos. Los objetos situados en los límites de los grupos no están obligados a pertenecer exclusivamente a un grupo determinado, sino que pertenecen de manera parcial a distintos grupos, en un grado de pertenencia que oscila entre 0 y 1. A pesar de su mayor coste computacional de $O(t \times c^2 \times n \times p)$, respecto al k medias de $O(t \times c \times n \times p)$, es muy usado, y en diferentes estudios como el de (Bora, 2014) ha analizado su comportamiento con datos de características heterogéneos.

Algoritmo Fuzzy C medias

En la explicación del algoritmo vamos asumir que disponemos de una muestra de tamaño n $\{X_1, \ldots, X_n\}$ y que disponemos de k clusters.

Este algoritmo minimiza la función objetivo:

$$J_{FCM}(X; U, V) = \Sigma_{i=1}^{k} \Sigma_{j=1}^{n} u_{ij}^{m} D_{ij}^{m} D_{ij\mathbf{A}}^{2}$$

Respecto a la del k medias, introduce ponderaciones dando lugar a los cuadrados de los errores ponderados. U es una matriz de partición, calculada a partir de los datos de X. U es una matriz de partición obtenida a partir de los datos. Representa el agrupamiento borroso. Indica grados de pertenencia, entre 0 y 1, del dato i-ésimo al cluster k-ésimo.. V es un vector de centroides. $D_{ij}^{m} D_{ij\mathbf{A}}^{2}$ son las distancias entre el vector de características i y el centroide del cluster j. Finalmente m es el parámetro de borrosidad. La función de pertenencia es suave puesto que cada dato no pertenece exclusivamente a un grupo. Si el valor es mayor que cero y es cercano a 1, hay un mayor grado de pertenencia al grupo considerado del dato, mientras que para valores mayores que cero pero alejados de 1, el grado de pertenencia al grupo considerado es muy bajo, en detrimento de otros grupos.

$D_{ij\mathbf{A}}^{2}$ se define como sigue:

$$D_{ij\mathbf{A}}^{2} = \|x_j - v_i\|_{\mathbf{A}}^{2} = (x_j - v_i)^T \mathbf{A} (x_j - v_i)$$

,

donde A es la norma simétrica y positiva de una matriz. El producto interior de $\mathbf{A}$ es una medida de distancias entre las observaciones y prototipos de los grupos. Cuando A es igual a $\mathbf{I}$, $D_{ij\mathbf{A}}^{2}$ es obtenida con la norma euclidiana al cuadrado.

Volviendo a la función objetivo anterior, a m se le llamará el parámetro de borrosidad, o exponente de ponderación, cuyo valor se elige como un número real, mayor que 1 ($m \in [1, \infty)$. A medida que m se aproxima a 1, el agrupamiento tiende a establecer grupos claramente diferenciados, pero al irse a infinito dicho agrupamiento se vuelve difuso. El valor de borrosidad se elige habitualmente como 2 en la mayoría de las aplicaciones. La función objetivo se minimiza con las siguientes restricciones

$$u_{ij} \in [0, 1]; 1 \leq i \leq c, 1 \leq j \leq n.$$

$$\Sigma_{i=1}^{c} u_{ij} = 1; 1 \leq j \leq n.$$

$$0 < \Sigma_{j=1}^{n} u_{ij} < n; 1 \leq i \leq k$$

FCM es un proceso iterativo y se detiene cuando el número de iteraciones alcanza su máximo, cuando la diferencia entre dos valores consecutivos de la función objetivo es menor que un valor de convergencia predefinido (ε). Los pasos que implica el FCM son los siguientes:

- Inicializa aleatoriamente la matriz de pertenencia $\mathbf{U}^{(0)}$
- Calcula los vectores prototipo: $\mathbf{v}_j = \frac{\Sigma_{i=1}^{n} u_{ij}^{m} \mathbf{x}_j}{\Sigma_{i=1}^{n} u_{ij}^{m}}; 1 \leq j \leq k$
- Calcula los valores de pertenencia con $u_{ij} = \frac{1}{\Sigma_{k=1}^{c} (D_{ij\mathbf{A}} / D_{ik\mathbf{A}})^{2/(m-1)}}; 1 \leq i \leq n, 1 \leq j \leq k$
- Compara $\mathbf{U}^{(t+1)}$ con $\mathbf{U}^{(t)}$, donde t es el número de la iteración.
- Si $\|\mathbf{U}^{(t+1)} - \mathbf{U}^{(t)}\| < \varepsilon$ entonces se detiene o vuelve al segundo paso del algoritmo.

2.4. Técnicas posteriores al planteamiento clásico

Analizamos en esta parte del capítulo las restantes técnicas de agrupamiento, consideradas en este trabajo.

2.4.1. El Algoritmo EM. La desigualdad de Jensen

El algoritmo EM, o de esperanza y maximización es un algoritmo en dos etapas que permite encontrar estimadores máximo-verosímiles de parámetros de modelos de probabilidad, que dependen de variables latentes-ocultas-. Presenta un elevado coste computacional. Para demostrar la convergencia del algoritmo EM la desigualdad de Jensen es crucial.

Sea I un intervalo de la recta real, decimos que $f : I \subset \mathbb{R} \to \mathbb{R}$

Si $f''(x) > 0$ para todo x, entonces f es estrictamente convexa.

Esto, para el caso multivariante equivale a que H es definida positiva $H > 0$.

Teorema 5. *Teorema: Si f es función convexa, para toda variable aleatoria X se verifica:*

$$\mathbb{E}[f(X)] \geq f(EX)$$

Como una función cóncava (caso real $f''(x) < 0$, caso multivariante $H <= 0$) es $g = -f$ de una función convexa. Triviamente del teorema anterior se deduce:

$$\mathbb{E}[g(X)] \leq g(EX)$$

El algoritmo EM

En muchas ocasiones el método de máxima verosimilitud tiene una convergencia lenta, en otras se queda atrapado fácilmente en mínimos locales, y ya en situaciones extremas cuando el tamaño muestral es muy bajo, la función de verosimilitud puede tomar valores muy elevados, lo que indudablemente conduce a que la estimación realiza sea completamente anómala (Mackay, 2003). Ante estos problemas surgió el algoritmo EM, propuesto por (Dempster, 1977). Este es una extensión del método de máxima verosimitud, que consiste principalmente en añadir una serie de variables latentes a la función de verosimilitud de forma adecuada, consiguiendo superar en muchos casos las limitaciones anteriores, aunque el coste computacionalmente se eleva considerablemente.

A continuación, supondremos que tenemos una muestra $x_{(1)}, ..., x_{(m)}$ de m datos independientes e idénticamente distribuidos, según una distribución f_θ. La condición de log-verosimilitud, se puede escribir

$$l(\theta) = \sum_{i=1}^{m} log p(x; \theta)$$

Introduciendo una variable latente z, lo anterior se puede escribir

$$= \sum_{i=1}^{m} log \sum_{z} p(x, z; \theta)$$

Para cada i, sea Q_i una distribución sobre los z $\sum_s Q_i(z) = 1, Q_i(z) \geq 0$. Se considera lo siguiente:

$$\sum_i \log p(x^{(i)}; \theta) = \sum_i \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta) \tag{2.35}$$

A continuación, asumimos la existencia de una variable aleatoria Q para la variable latente-oculta o desconocida-z.

$$= \sum_i \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \tag{2.36}$$

$$\geq \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \tag{2.37}$$

El último paso se deduce de forma inmediata de la desigualdad de Jensen, al ser f una función cóncava.

$$\sum_{z^{(i)}} Q_i(z^{(i)}) \left[\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right]$$

El sumatorio es simplemente una esperanza de la $p(x^{(i)}, z^{(i)}; \theta / Q_i(z^{(i)}))$

con relación a los $z^{(i)}$ tomados de acuerdo a la distribución dada por Q_i. Entonces, por la desigualdad de Jensen, tenemos

$$f\left(E_{z^{(i)} \sim Q_i} \left[\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right] \right) \geq E_{z^{(i)} \sim Q_i} \left[f\left(\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i z^{(i)}} \right) \right]$$

Los subíndices $z^{(i)} \sim Q_i$ indican que las esperanzas relativas $z^{(i)}$ se obtienen a partir de distribuciones Q_i, con lo que se pasa de la ecuación 2.35 a la 2.36. Ahora, entre todos los conjunto de distribuciones Q_i, se establece cual es el mejor a elegir. En la práctica para simplificar el problema se pide:

$$\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} = c$$

Para una constante c que no dependa de z^i. Esto se logra eligiendo

$$Q_i(z^{(i)}) \propto p(x^{(i)}, z^{(i)}; \theta)$$

De hecho, se sabe que $\sum_z Q_i(z^{(i)}) = 1$. Esto es así a porque se trata de una distribución. Adicionalmente, esto indica que

$$\begin{aligned} Q_i(z^{(i)}) &= \frac{p(x^{(i)}, z^{(i)}; \theta)}{\sum_z p(x^{(i)}, z; \theta)} \\ &= \frac{p(x^{(i)}, z^{(i)}; \theta)}{p(x^{(i)}; \theta)} \\ &= p(z^{(i)}|x^{(i)}; \theta) \end{aligned}$$

(Paso E) Para cada i,se establece

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)}; \theta)$$

(Paso M)

$$\theta := \underset{\theta}{argmax} \sum_i \sum_{z(i)} Q_i(z^{(i)}) log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

Para saber si el algoritmo converge, se supone que $\theta(t)$ y $\theta(t+1)$, son los parámetros para dos iteraciones sucesivas del algoritmo. Probamos a continuación que $l\theta^{(t)} \leq l\theta^{(t+1)}$, lo que muestra que EM siempre mejora la log-verosimilitud de forma positiva. La clave para mostrar este resultado reside en la elección de los Q_i. Específicamente, en la iteración del algoritmo en la que los parámetros se hayan iniciado como $\theta^{(t)}$, se elegiría $Q_i^{(t)}(z^{(i)}) := p(z^{(i)}|x^{(i)}; \theta^{(t)})$.

Dicha elección, asegura que la desigualdad de Jensen, cumple con la igualdad, y por lo tanto

$$l(\theta^{(t)}) = \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})}$$

Los parámetros $\theta^{(t+1)}$ se obtienen maximizando el segundo término de la ecuación anterior. Por lo tanto,

$$l(\theta^{(t+1)}) \geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})} \tag{2.38}$$

$$\geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})} \tag{2.39}$$

$$l(\theta^{(t)}) \tag{2.40}$$

la desigualdad de la ecuación 2.38 viene del hecho que

$$l(\theta) \geq \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i z^{(i)}}$$

se cumple para cualquier valor de Q_i y de θ, y en particular se cumple para $Q_i = Q_i^{(t)}, \theta = \theta^{(t+1)}$ En la obtención de la ecuación 2.39, se usa el hecho de que $\theta^{(t+1)}$ se elige de forma explícita para ser

$$\arg\underset{\theta}{\max} \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i z^{(i)}}$$

y por lo tanto, esta fórmula evaluada en $\theta^{(t+1)}$ ha de ser igual o mayor que la fórmula previamente evaluada en $\theta^{(t)}$. Finalmente, el paso realizado para obtener la ecuación 2.40, y se obtiene de que $Q_i^{(t)}$, sea eligida para hacer que la desigualdad de Jensen cumpla la igualdad para $\theta^{(t)}$.

Por tanto, el algoritmo EM hace que la verosimilitud converga monotónicamente. Como se ha detallado previamente en la descripción del algoritmo, este se ejecuta, hasta alcanzar la convergencia. Una vez obtenido el resultado mostrado, un test de convergencia adecuado sería comprobar si el incremento en $l(\theta)$ entre iteraciones sucesivas es menor que un parámetro de tolerancia determinado, y poder así confirmar la convergencia en caso de que el algoritmo EM mejore $l(\theta)$ muy lentamente.

Aclaración: Si se define

$$J(Q,\theta) = \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

Sabemos entonces,por la derivación previa que $l(\theta) \geq J(Q,\theta)$. El algoritmo EM puede también ser visto como un método de descenso de coordenadas respecto a Q, y el paso M hará lo propio, respecto a θ.

Revisión de la mixturas gausianas

El algoritmo EM es de gran utilidad para realizar el agrupamiento-clustering- de una mixtura de gausianas. Una vez concluida la definición del algoritmo EM, con las herramientas ya disponibles, se revisa el caso del ajuste de los parámetros ϕ, μ y Σ en una mixtura de gausianas. Por brevedad, se realizan las derivaciones para las actualizaciones del paso M únicamente para ϕ, μ. El paso E es sencillo. Siguiendo la derivación del algoritmo ya realizada, se calcula

$$\{\omega\}_j^{(i)} = Q_i(z^{(i)} = j) = P(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$$

El término $Q_i(z^{(i)} = j)$ indica la probabilidad de $z^{(i)}$ tomando el valor de j bajo la distribución Q_i. A continuación, en el paso M, es necesario maximizar, con respecto a los parámetros ϕ, μ, Σ, la cantidad

$$\sum_{i=1}^{m} \sum_{z^(i)} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \phi, \mu, \Sigma)}{Q_i(z^{(i)})}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{k} Q_i(z^{(i)} = j) \log \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{Q_i(z^{(i)} = j)}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{k} \omega_j^{(i)} \log \frac{\frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_j|^{1/2}} \exp(\frac{-1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j)).\phi_j}{\omega_j^{(i)}}$$

Se maximiza esto con respecto a μ_ι. Si se considera la derivada respecto a μ_ι, encontramos que

$$\nabla \mu_\iota \sum_{i=1}^{m} \sum_{j=1}^{k} \omega_j^{(i)} \log \frac{\frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_j|^{1/2}} \exp(\frac{-1}{2}(x^{(i)} - \mu_j)^T \Sigma_j{}^{-1} 1 (x^{(i)} - \mu_j)).\phi_j}{\omega_j^{(i)}}$$

$$= - \nabla_{\mu_\iota} \sum_{i=1}^{m} \sum_{j=1}^{k} \omega_j^{(i)} \frac{1}{2} (x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j)$$

$$= \frac{1}{2} \sum_{i=1}^{m} w_\iota^{(i)} \nabla_{\mu_\iota} 2\mu_\iota^T \Sigma_\iota^{-1} x^{(i)} - \mu_\iota^T \Sigma_\iota^{-1} \mu_\iota$$

$$= \sum_{i=1}^{m} \omega_\iota^{(i)} (\Sigma_\iota^{-1} x^{(i)} - \Sigma_\iota^{-1} \mu_\iota)$$

Si se fija esto a cero y despejamos μ_ι se genera la regla de actualización

$$\mu_\iota := \frac{\sum_{i=1}^{m} \omega_\iota^{(i)} x^{(i)}}{\sum_{i=1}^{m} \omega_\iota^{(i)}}$$

que es lo que teníamos inicialmente

Para realizar otra aplicación, se deriva ahora la actualización del paso M para los parámetros ϕ_j Agrupando los términos que dependen de ϕ_j, se observa que es necesario maximizar

$$\sum_{i=1}^{m} \sum_{j=1}^{k} \omega_j^{(i)} \log \phi_j$$

Hay, sin embargo una restricción adicional de que la suma de los ϕ_j es 1, al representar estos las probabilidades $\phi_j = p(z_{(i)} = j; \phi)$. Para cumplir la restricción de que $\Sigma_{j=1}^{i}\phi_j = 1$, se construye el Lagrangiano

$$\ell(\phi) = \sum_{i=1}^{m}\sum_{j=1}^{k}\omega_j^i \log\phi_j + \beta(\sum_{j=1}^{k}\phi_j - 1),$$

donde β es el multiplicador de Lagrange. Derivando tenemos que:

$$\frac{\partial}{\partial\phi_j}\ell(\phi) = \sum_{i=1}^{m}\frac{\omega_j^{(i)}}{\phi_{(j)}} + \beta$$

Igualando esta expresión a cero y despejando $\phi_{(j)}$,tenemos que

$$\phi_j = \frac{\sum_{i=1}^{m}\omega_j^{(i)}}{-\beta}$$

Consideremos que $\phi_j \propto \Sigma_{i=1}^{m}\omega_j^{(i)}$. Empleando la restricción de que $\Sigma_{i=j}\phi^j = 1$ se obtiene que $-\beta = \Sigma_{i=1}^{m}\Sigma_{j=1}^{k}\omega_j^{(i)} = \Sigma_{i=1}^{m}1 = m$. Para ello se utiliza el hecho de que $\omega_j^{(i)} = Q_i(z^{(i)} = j)$, y como las probabilidades suman 1, $\Sigma_j\omega_j^{(i)} = 1$. Se obtienen así actualizaciones del paso M para el parámetro ϕ_j :

$$\phi_j := \frac{1}{m}\Sigma_{i=1}^{m}\omega_j^{(i)}.$$

La derivación para las actualizaciones de Σ_j es ánaloga

2.4.2. El agrupamiento espectral

Las técnicas de agrupamiento espectral han sido desarrolladas inicialmente por (Donath, 2003), así como (Shi, 2000), y desde entonces han tenido una gran popularidad en la literatura, al ser muy útil su aplicación con un número elevado de variables.

Siendo más precisos, podemos decir que el agrupamiento espectral es un conjunto de técnicas basadas en descomposición espectral de una matriz de similaridad calculada a partir de la muestra. Una matriz de similaridad $S = (s_{ij})\ I$, con $j = 1, ..., n$. Dicha matriz está formada por puntuaciones de 0 a 1, que indican el grado de similaridad entre los pares de puntos de una muestra.

Construcción de un grafo de similaridad

Las relaciones entre pares de variables se suelen interpretar a partir de un grafo $G = (V, E)$ de vértices V y aristas E, y el valor de la distancia de similitud entre cada par indica el peso en el grafo. Los vértices del grafo v_i representa a los puntos x_i.

Considerando G un grafo no dirigido con un conjunto de vértices $V = v_1, \cdot, v_n$ y ponderaciones $w_{ij} \geq 0$ para cada arista formada por v_i y v_j. La matriz de adyacencia ponderada del grafo es una matriz $W = (w_{ij})_{i,j=1,...,n}$, que refleja relaciones binarias. Al tratarse de un grafo no dirigido la matriz será simétrica. Finalmente se define la matriz de grados D mediante $dii = \sum_{j=1}^{n} w_{ji}$ y $d_{ij} = 0$ (si $i \neq j$).

Algoritmo de agrupamiento espectral no normalizado

Sea $X_1, X_2, ..., X_n$ una muestra aleatoria compuesta por n datos. La similitud entre todos los pares de datos, se recoge con la siguiente matriz de similitud $S_{ij} = S(x_i, x_j)\ \forall i, j \in \{1, \ldots, n\}$, donde s denota una medida de similitud.

El algoritmo del agrupamiento espectral no normalizado se especifica a continuación. En él se asume que s es la medida de similitud y el objetivo es establecer k grupos-clusters-.

1. Calcular la matriz de similaridad $S \in \mathbb{R}^{n\times n}$, asumiendo n puntos y k el número de grupos a construir.
2. Construir un grafo de similaridad. Sea W su matriz de adyacencia ponderada.
3. Calcular la matriz Laplaciana no normalizada L como $L = D - A$, siendo D la matriz de grados de A.
4. Calcular los primeros k autovectores $u_1, ..., u_k$ de L.

5. Sea $U \in \mathbb{R}^{n \times k}$ la matriz que contiene los vectores $u_1, ..., u_k$ en forma de columnas.
6. Para $i = 1, ..., n$, sea $y_i \in \mathbb{R}^k$ el vector correspondiente a la fila i-ésima de U.
7. Agrupar los puntos (y_i) $i = 1, ..., n$ en $\mathbb{R}^k$, mediante el algoritmo k medias en los grupos $C_1, ..., C_k$.
8. Se obtiene como resultado los grupos $A_1, ..., A_k$ con $A_i = \{X_j | y_j \in C_i\}$.

2.4.3. Algoritmo DBSCAN

Introducción

El algoritmo DBSCAN, es una técnica de agrupamiento que se basa en buscar regiones de datos de alta densidad.

Puede hallar grupos no lineales, o cualquier forma arbitrarias, y suele usarse, cuando el k medias o el algoritmo EM no funcionan correctamente. Fue desarrollado por (Ester, 1996).

La idea principal detrás del algoritmo DBSCAN es que para cada punto de un grupo o cluster, en el entorno de un radio determinado ha de contener, al menos un número mínimo de puntos. Así, la densidad del vecindario tendrá que exceder un umbral determinado. La forma de un vecindario se determina por la elección de una función $d(p, q)$. Aunque en la práctica, puede usarse cualquier función para definir la función anterior, su elección dependerá de la aplicación a utilizar.

El algoritmo requiere dos parámetros:

- minPts: es el mínimo tamaño deseado, o el número mínimo de puntos que ha de existir en el entorno para dar lugar a un grupo.
- Eps: es el radio máximo del vecindario de un punto.

Asímismo, se denotara por k el tamaño o dimensión del vecindario.

Clasificación de los puntos realizada por el algoritmo

Cuando el algoritmo DBSCAN considera un conjunto de puntos pertenecientes a un espacio determinado, con el fin de agruparlos, clasifica dichos puntos en:

Puntos núcleo Son aquellos conjuntos de puntos, que tienen un número superior a MinPts en el interior del entorno con radio Eps. Un punto núcleo está en el interior del grupo.

Puntos frontera Son aquellos conjuntos de puntos con un número inferior a MinPts en el interior de Eps, pero se encuentran en el entorno de un conjunto puntos núcleo.

Outliers Los outliers del conjunto de datos, son aquellos conjuntos de puntos que no son puntos núcleo o que no son puntos frontera.

Relaciones de densidad entre los puntos El algoritmo distingue tres tipos:

Puntos directamente alcanzables en densidad son aquellos puntos p frontera que desde un punto q núcleo, con respecto a Eps y MinPts cumplen las siguientes 2 condiciones:

- $p \in N_{Eps}(q)$
- $|N_{Eps}(q)| \geq MinPts$

donde $N_{Eps}(q)$ denota un entorno con centro q y radio EPS.

Puntos alcanzables en densidad Un punto q es alcanzable en densidad desde p, respecto a Eps y $MinPts$ si hay una cadena de puntos o trayectoria $p_1, ..., p_n$ con $p_1 = p$ y $p_n = q$, donde cada p_{i+1} es directamente alcanzable desde p_i (todos los puntos de la trayectoria han de ser puntos núcleo, con la posible excepción de q).

Puntos conectados en densidad Son puntos p, conectados a puntos q con respecto a Eps y MinPts si hay un punto o tal que los puntos, p y q, son alcanzables en densidad desde o.

Definición formal de grupo y sus propiedades para este algoritmo

Aquellas regiones con una alta densidad indican la existencia de grupos. Un grupo o cluster para el algoritmo DBSCAN, es un conjunto de puntos conectados en densidad que es máximo con respecto a la alcanzabilidad en densidad.

A continuación se define formalmente el concepto de cluster como sigue. Sea D el conjunto de datos. Un grupo C respecto a Eps y MinPts es un subconjunto no vacío de D, que satisfaga las siguientes condiciones:

- (1)$\forall\ \ p, q$: si $p \in C$ y q es alcanzable en densidad desde p con respecto a Eps y MinPts, entonces $q \in C$. Se da así la maximalidad. Todos los puntos dentro del grupo están conectados en densidad mutuamente.
- (2)$\forall p, q \in C$: p si p está conectado en densidad a q con respecto a EPS y MinPts. Si un punto es alcanzable en densidad desde cualquier punto del grupo, es una parte del grupo también.

Ventajas e inconvenientes del DBSCAN

Entre las principales ventajas del DBSCAN destaca que no requiere el número de grupos inicial. Es robusto a los atípicos. Puede encontrar un grupo rodeado por otro pero no conectado.

Como principales inconvenientes destaca el hecho que no ser adecuado para separar grupos cercanos. No puede tampoco agrupar conjuntos de datos que presenten grandes diferencias en las densidades. Ello es debido a la imposibilidad de elegir una combinación adecuada de min-Pts para todos los grupos. Es necesario además una comprensión de los datos y la escala, pues es difícil elegir un umbral de distancia.

Finalmente, si Eps tiene valores muy bajos, algunos puntos serán considerados ruido en vez de ser agregados al grupo. Cuando se observan puntos que deberían incorporarse a un grupo y que se clasifican como ruido, el valor de Eps debería ser más elevado, aunque ello puede generar otros problemas. Finalmente, cuando existen grupos con diferentes densidades, esto hace que algunos puntos establezcan conexiones entre dos conjuntos diferentes, haciendo que formen un único grupo.

2.4.4. Agrupamiento basado en subespacio

Esta familia de algoritmos estudian el problema de modelizar un conjunto de datos, formado por la unión de subespacios. Sea $\{X_j \in \mathbb{R}^D\}_{j=1}^N$ un conjunto dado de puntos sacados de una unión no conocida de $n \geq 1$ subespacios lineales o afines $\{S_i\}_{i=1}^n$ de dimensiones desconocidas $d_i = dim(S_i), 0 < d_i < D, i = 1, ..., n$. Los subespacios pueden describirse como

$$S_i = \{x \in \mathbb{R}^D : x = \mu_i + U_i\}, i = 1, ..., n, \tag{2.41}$$

Donde $\mu_i \in \mathbb{R}^D$ es un punto arbitrario en el subespacio S_i $\mu_i = 0$ para espacios lineales, $U_i \in \mathbb{R}^{d^i}$ es una representación en baja dimensión, para el punto x. El objetivo del agrupamiento de subespacio es encontrar:

- El número de subespacios n.
- Sus dimensiones $\{d_{i=1}^n\}$.
- Las bases de subespacio $\{U_i\}_{i=1}^n$.
- Los puntos $\{\mu_i\}_{i=1}^n$, en el caso de espacios afines.
- La forma de segmentación de los puntos, en función de los subespacios.

Cuando el número de subespacios es igual a 1, el problema queda reducido a hallar:

- Un vector $\mu \in \mathbb{R}^D$.
- Una base $U \in \mathbb{R}^{D \times d}$.
- Una representación de baja dimensión $Y = [y_1, ..., y_N] \in \mathbb{R}^{d \times N}$.
- La dimensión d.

Este problema es conocido como Análisis de Componentes Principales y se puede resolver de una forma muy sencilla: $\mu = \frac{1}{N}\Sigma_{j=1}^N x_j$ es la media de los puntos (U, Y) que se pueden obtener a partir del rango-d de la descomposición del valor singular de la matriz de datos X, tomados de la media. $X = [x_1 - \mu, x_2 - \mu, ..., x_N - \mu] \in \mathbb{R}^{DxN}$ como

$$U = \upsilon Y = \Sigma\nu^T, dondeX = \upsilon\Sigma\nu^T \tag{2.42}$$

además, d puede obtenerse como $d = rango(X)$ con datos libres de ruido, o usando técnicas de selección de modelo, en caso de que los datos contienen ruido. Cuando n es mayor que 1, el problema del agrupamiento de subespacio aumenta significativamente su dificultad, dado el número de retos:

1. Hay un fuerte acoplamiento entre la segmentación de datos y la estimación del modelo. En particular, si la segmentación de los datos fuera conocida, se podría ajustar cada subespacio individual a cada grupo de puntos usando el Análisis de Componentes Principales estándar. Por otra parte, si los parámetros de subespacio se conocieran, se podría fácilmente encontrar los puntos que mejor ajustan cada subespacio. En la práctica, ni la segmentación, ni los parámetros del subespacio son conocidos y se necesita resolver ambos problemas simultáneamente.
2. La distribución de los datos dentro de los subespacios generalmente se desconoce. Si los datos dentro de cada grupo se distribuyen en torno al centro de grupo, y los centros de los grupos, para los diferentes subespacios están alejados, entonces el problema del agrupamiento de subespacio se reduce al más sencillo y mejor estudiado problema del agrupamiento central, en donde los grupos se distribuyen en torno a múltiples centros de grupo. Por otra parte, si la distribución de los datos puntuales en los subespacios es arbitrario y hay muchos puntos cerca de la intersección de los subespacios, entonces el problema no podrá resolverse con técnicas de centrado de agrupamiento.
3. La posición relativa de los subespacios puede ser arbitraria. Cuando 2 subespacios se intersecan o están muy cercanos, el problema del agrupamiento de subespacio se convierte en un problema muy difícil. Sin embargo, cuando los subespacios son disjuntos o son independientes, el problema es menos complejo.
4. Los datos pueden corromperse por el ruido, los datos o entradas faltantes, así como por datos atípicos-outliers-. Dichas complicaciones pueden causar que los subespacios estimados sean totalmente erróneos. Mientras que las técnicas de estimación robusta han sido desarrolladas para el caso de un subespacio individual, el caso de los subespacios múltiples no son comprendidos en igual forma.
5. En el Análisis de Componentes Principales clásico, el único parámetro es la dimensión del subespacio, el cual puede ser hallado mediante la búsqueda de la menor dimensión que ajusta los datos con una precisión dada. En el caso de los subespacios múltiples se puede ajustar los datos con n subespacios diferentes de dimensión 1, denominado 1 subespacio por punto, o mediante un subespacio sencillo de dimensión D. Obviamente, ninguna solución es satisfactoria. El reto consiste en encontrar un criterio de selección de modelo que favorezca la obtención de un pequeño número de subespacios de pequeña dimensión.

Capítulo 3

BONDAD DE AJUSTE Y COMPARACIONES MÚLTIPLES

3.1. Bondad de ajuste

Consideramos dos métodos de contraste:

3.1.1. Test de Wilcoxon-Mann-Whitney

Es un test de comparación de dos muestras respecto a su mediana. La idea del test es muy simple se ordenan las observaciones, y se comparan las ordenaciones promedio entre dos muestras. Aunque dicha idea ha aparecido varias veces en diversas disciplinas, fue propuesta por (Wilcoxon, 1945), a quien siguieron con rapidez posteriores desarrollos, siendo el primero de ellos el trabajo de (Mann, 1947).

Representaciones del test

Sea una muestra aleatoria simple $X_1, X_2, ..., X_m$ con función de distribución desconocida, F.
Se define

$$F_m(t) = \sum_{i=1}^{m} \frac{1}{m} I(X_i \leq t) \tag{3.1}$$

De manera análoga, para otra muestra aleatoria simple $Y_1, Y_2, \ldots, Y_n$, siendo

$$G_n(t) = \sum_{j=1}^{m} \frac{1}{m} I(Y_j < t) \tag{3.2}$$

Se asume además, que $N = m + n$, si $r_i = r(x_i)$ es el rango de x_i en la muestra combinada. Sea $R(x) = \Sigma_{i=1}^{m} r(x_i)$. Además si $s_i = s(y_i)$ es el rango de y_j en la muestra combinada, $R(y) = \Sigma_{j=1}^{n} s(y_j)$.
Se observa que $R(x) + R(y) = N(N+1)/2$ en donde cada término representa la suma de las cantidades $1, 2, \ldots, N$
Se define ahora

$$I(x < y) = \sum_{i=1}^{m} \sum_{j=1}^{n}, I(x_i < y_j) \tag{3.3}$$

Mann-Whitney muestra que

$$R(y) = n(n+1)/2 + I(x < y) \tag{3.4}$$

Debido a que una transformación continua monótona como $x^{\frac{1}{2}}$ o el $logx$ no cambia las relaciones de orden, tanto $I(x < y)$ y $R(y)$, no se ven afectadas.

Teoría de la distribución cuando $F = G$ (se cumple H_0)

La distribución exacta bajo la nula de $I(x < y)$ se obtiene a partir de la muestra ordenada conjunta.
Bajo H_0 se puede ver fácilmente que la media y la varianza de $I(x < y)$ son respectivamente:

$$E_0[I(x < y)] = \frac{mn}{2} \tag{3.5}$$

$$var_0[I(x < y)] = \frac{mn(N+1)}{12} \tag{3.6}$$

La normalidad asintótica, es una manera de aproximar la distribución cuando el tamaño muestral es razonable.

3.1.2. Estadístico Kolmogorov-Smirnov

Fundamentos teóricos

Sea $X_1, X_2, \ldots, X_n$ una muestra de una función de distribución F. Se plantea el problema fundamental de cómo es F o qué forma tiene. Se plantea entonces obtener un estimador estadístico de F desconocida. Se considera la función de distribución empírica

$$\hat{F}_n(x) := \frac{1}{n}\Sigma_{i=1}^{n} I\{X_i \leq x\} \tag{3.7}$$

Veamos algunas propiedades de la empírica:

Consistencia e insesgadez en un punto

Fíjese x *in* R entonces

$$n\hat{F}_n(x) \sim binomial(n, F(x))$$

. Consecuentemente

$$E[\hat{F}_n(x)] = \frac{1}{n}E[n\hat{F}_n(x)] = F(x) \tag{3.8}$$

por lo cual $(\hat{F}_n(x))$ es un estimador insesgado de $F(x)$ para cada x fijado. Además,

$$Var(\hat{F}_n(x)) = \frac{1}{n^2}Var(n\hat{F}_n(x)) = \frac{F(x)[1-F(x)]}{n} \tag{3.9}$$

y por la desigualdad de Chebyshev resulta que, $\hat{F}_n(x) \xrightarrow{P} F(x)$. Por lo tanto, $\hat{F}_n(x)$ es un estimador insesgado y consistente de $F(x)$ para cada x fijo $\in \mathbb{R}$

Partiendo de esto pueden construirse intervalos de confianza para $F(x)$, para un x fijo. Se plantea ahora que sucede si lo que se busca es tener un conjunto de confianza para $(F(x_1), F(x_2)$, o tener conocimiento de la función F en su totalidad.

El estadístico Kolmogorov-Smirnov

$$D_n := \max_{-\infty<x<\infty} |\hat{F}_n(x) - F(x)| \tag{3.10}$$

Teorema 6. *Teorema Glivenco-Cantelly Se plantea lo siguiente:*

$$n \to \infty, D_n \xrightarrow{C.S.P} 0$$

En particular, podemos fijar un $\epsilon > 0$ pequeño y deducir que si n es grande, entonces con alta probabilidad $D_n \leq \epsilon$

$$\zeta_n(\epsilon) := \{(x, y) : |\hat{F}_n(x) - y| \leq \epsilon\} \tag{3.11}$$

Esto nos permite tener una idea aproximada de la forma de F, debido a que $|D_n| \leq \epsilon$.

Teorema 7. *Teorema*

D_n presenta la propiedad de la distribución libre. La distribución de D_n es la misma para todas las familias de funciones de distribución F continuas.

Para demostrarlo primero se prueba que F es estrictamente creciente. En dicho caso, su inversa F^{-1}, existe y es también estrictamente creciente. Por tanto.

$$D_n = \underset{-\infty<x<\infty}{\text{máx}} |\hat{F}_n(x) - F(x)| \tag{3.12}$$
$$= \underset{0\le y\le 1}{\text{máx}} |\hat{F}_n(F^{-1}(y)) - F(F^{-1}(y))|$$
$$= \underset{0\le y\le 1}{\text{máx}} |\hat{F}_n(F^{-1}(y)) - y|$$

Ahora

$$\hat{F}_n(F^{-1}(y)) = \frac{1}{n}\sum_{i=1}^{n} I\{X_i \le F^{-1}(y)\} = \frac{1}{n}\sum_{i=1}^{n} I\{F(X_i) \le y\} \tag{3.13}$$

y esta es la función de distribución empírica para la muestra aleatoria i.i.d $F(X_1), \ldots, F(X_n)$. Esta es una muestra de la distribución uniforme (0,1), y por tanto encontramos que la distribución de D_n es la misma que para el estadístico Kolmogorov-Smirnov, para una muestra uniforme(0,1). Esto prueba el resultado, en el caso de que F^{-1} exista. En el caso general, F^{-1} no existe necesariamente. Sin embargo consideremos $F^{+}(x) := miny : F(x) > x$.Entonces, esto tiene la propiedad de que $F^{+}(x) \le z, \to x \le F(z)$.Se sustituye $F^{+}(x)$ por todos los $F^{-}(x)$ en la demostración previa.

Intervalos de confianza y test puntuales

Se busca describir intervalos de confianza a un nivel $(1-\alpha)$ asintótico, para $F(X)$, para un $x \in R$. Téngase en cuenta que $n\hat{F}_n(x) \sim B(n, F(x))$. Por tanto, un intervalo de confianza, con un nivel de confianza asintótico $(1-\alpha)$, para un $x \in R$. Se recuerda que $n\hat{F}_n(x) \sim B(n, F(x))$. Por tanto, por el teorema central del límite

$$\frac{n[F_n(x) - F(x)]}{(nF(x)[1-F(x)])^{1/2}} \xrightarrow{d} N(0,1) \tag{3.14}$$

también, $\hat{F}_n(x) \xrightarrow{P} F(x)$. Por tanto el teorema de Slutsky implica que como $n \longrightarrow \infty$

$$n^{1/2}\frac{\hat{F}_n(x) - F(x)}{(\hat{F}_n(x)[1-\hat{F}_n(x)])^{1/2}} \xrightarrow{d} N(0,1) \tag{3.15}$$

Por tanto, un intervalo de confianza con un nivel $(1-\alpha)$ asintótico para $F(x)$ es

$$\zeta_n(\alpha) := \left[\hat{F}_n(x) - z_\alpha/2\sqrt{\frac{\hat{F}_n(x)[\hat{F}_n(x)]}{n}}, \hat{F}_n(x) + z\alpha/2\sqrt{\frac{\hat{F}_n(x)[1-\hat{F}_n(x)]}{n}}\right] \tag{3.16}$$

suponiéndose que se realiza un contraste de hipótesis $H_0 : F(x) = F_0(x)$ frente a $H_1.F(x) \ne F_0(x)$

siendo F_0 una función de distribución fijo y conocido, entonces un nivel $(1-\alpha)$ asintótico puede basarse en:

$$rechazarH_0 \Leftrightarrow \frac{|\hat{F}_n(x) - F_0(x)|}{\sqrt{F_0(x)[1-F_0(x)]}} > \frac{z\alpha/2}{\sqrt{n}}$$

Teoría de procesos empíricos

Se considera ahora un problema más realista para encontrar intervalos de confianza simultáneo para $F(x)$, simultáneamente sobre todo $x \in R$. O también supóngase que el objetivo es realizar el contraste $H_0 : F = F_0$ frente a $H_1 : F \ne F_0$, donde F_0 es conocida. Básicamente los dos son el mismo problema. Supóngase que la distribución exacta de $D_n(F_0) := max_x|\hat{F}_n(x) - F_0(x)$ es conocida. Se puede, entonces hallar $\delta_\alpha(n)$ tal que $P_F\{D_n(F) \le \delta_\alpha(n)\} \ge 1-\alpha$,para todo F. Considérese ahora el intervalo de confianza $C_n(\alpha) := \{F : D_n(F) \le \delta_\alpha(n)$ Su nivel de confianza es $(1-\alpha)$. Nótese que $\delta_\alpha(n)$ no depende de F, debido a que D_n! es una distribución libre. De manera adicional $\delta_\alpha(n)$ puede simularse: Sin pérdida de la generalidad, supóngase que $F \sim U(0,1)$. En dicho caso, $D_n = max_{1\le j\le n}|X_{j:n} - (j/n)|$, cuya distribución puede ser simulada mediante Monte-Carlo. Sin embargo, por motivos teóricos, puede ser útil construir un intervalo de confianza asintótico a un nivel de

$(1-\alpha)$ Esto también es útil cuando n es grande. Para ello se necesita saber la rapidez con la que D_n converge a 0. Para comprenderlo, asumimos que $X_1, ..., X_n$ se distribuye como una Uniforme(0,1), así que:

$$F(x) = x \forall 0 \leq x \leq 1$$

Consideramos primero el vector aleatorio

$$\sqrt{n}\begin{bmatrix} \hat{F}_n(x_1) - F(x_1) \\ . \\ . \\ . \\ \hat{F}_n(x_k) - F(x_k) \end{bmatrix} = \frac{1}{\sqrt{n}}\Sigma_{i=1}^{n}\begin{bmatrix} \{I\}X_1 \leq x_1\} - F(x_1) \\ . \\ . \\ . \\ \{I\}X_k \leq x_1\} - F(x_k) \end{bmatrix} := \frac{1}{\sqrt{n}}\Sigma_{i=1}^{n} Z_i$$

donde $x_1, ..., x_k$ están fijados.

$Z_1, Z_2...$ son vectores aleatorios k-dimensionales i.i.d con $EZ_1 = 0$ y $Cov(Z_1) = Q$, donde $Q_{i,i} = F(X_i)\{1 - F(x_i)\}, 1 \leq i \leq k$ y $Q_{i,j} = F(min(x_i, x_j)) - F(x_i)F(x_j) = F(min(x_i, x_j))\{1 - F(max(x_i, x_j))\}, 1 \leq i \neq j \leq k$

Por el teorema central del límite multidimensional $n^{-1/2}\Sigma_{i=1}^{n} Z_i$ converge en distribución a $N_k(0, Q)$. En particular, bajo F,

$$\sqrt{n} max_{1 \leq i \leq k} |\hat{F}_n(x_i) - F(x_i)| \xrightarrow{d} \max_{1 \leq i \leq k} |W_i|$$

Donde

$$W = (W_1, \ldots, W_k)\prime \sim N_k(0, Q)$$

Se elige ahora una pequeña partición de $[0, 1]$ $x_1, \ldots, x_k$, para ver lo cercano que está por la izquierda a $\sqrt{n}D_n$. Por lo tanto, puede realizarse la conjetura de que $\sqrt{n}D_n$ converge en distribución. Para averiguar el límite asintótico, necesitamos comprender mejor cómo es por la derecha, lo que conduce a los procesos gausianos, en concreto al movimiento browniano o proceso de Wiener.

Una revisión de los procesos empíricos

Nuevamente, sea $X_1, X_2, \ldots, X_n$ variables aleatorias $U(0,1)$ i.i.d. Consideremos la función aleatoria

$$E_n(x) := \sqrt{n}[\hat{F}_n(x) - x], 0 \leq x \leq 1$$

Nótese que se emplea el $max_{0 \leq x \leq 1}|$, el estadístico K-S. Reconsideramos también el resultado del teorema central del límite multidimensional, bajo F. Encontramos de este modo la demostración a lo siguiente.

$$\sqrt{n}\begin{bmatrix} \{E\}_n(x_1) \\ . \\ . \\ . \\ \{E\}_n(x_k) \end{bmatrix} \xrightarrow{d} \begin{bmatrix} \{B\}^o(x_1) \\ . \\ . \\ . \\ \{B\}^o(x_k) \end{bmatrix}$$

En particular,

$$\sqrt{n} \max_{1 \leq j \leq k} | E_n(x_j)| \xrightarrow{d} \max_{1 \leq j \leq k} |B^o(x)|$$

$$\sqrt{n}D_n := \max_{0 \leq x \leq 1} | E_n(x_j)| \xrightarrow{d} \max_{0 \leq x \leq 1} |B^o(x_j)|$$

La ventaja aquí reside en que la distribución de la variable aleatoria por la derecha es conocida. El movimiento browniano es bien conocido. B^0 además está relacionado con los procesos gausianos. Consideramos que

$$B^0(t) = B(t) - tB(1)$$

Tomamos como ejemplo

$$P\left\{ \underset{0\leq t\leq 1}{max} |B^o(t)| > x \right\}$$

Por lo tanto, Si n es "grande"tenemos que

$$P\left\{ D_n > \frac{x}{\sqrt{n}} \right\} \approx 2 \sum_{k=1}^{\infty} (-1)^{k+1} e^{-2k^2x^2 - 2e^{-8x^2}} \pm ...$$

3.2. Ajuste para comparaciones múltiples

3.2.1. Introducción

En las ciencias experimentales, la inferencia estadística hace posible establecer si hay evidencias de que las hipótesis científicas planteadas no son ciertas. Para ello, a partir de la hipótesis nula H_0 fijada de antemano, y la muestra $X_1, \cdot X_n$ de datos, se calcula la probabilidad de que dicha hipótesis sea cierta. A dicha valor se le conoce como p valor

No obstante, en la práctica se contrastan múltitud de hipótesis simultáneamente, lo que se llama el problema de comparación de contrastes múltiples. En los ensayos clínicos frecuentente se comparan simultáneamente los efectos que los niveles de una o más dosis de un nuevo medicamento, tienen en comparación al tratamiento habitual. También los avances en secuenciación genética, hacen habitual que puedan determinarse simultáneamente expresiones diferenciales, entre decenas de miles de genes, siendo necesario establecer si cada gen es realmente significante con la patología.

3.2.2. Planteamiento matemático

Si contrastamos m hipótesis de manera simultanea, los posibles resultados son

Hipótesis nula (H_0)	Significativo (H_0 rechazado)	No significativo (H_0 no rechazado)
m Total	R	m-R
m_0 Verdadero	U	$m_0 - U$
$m - m_0$ Falso	$R - U$	$m - R - (m_0 - U)$

3.2.3. Conceptos a usar

Errores de tipo I y tipo II

Los errores estadísticos que se cometen en un problema de bondad de ajuste de comparaciones múltiples son los siguientes:

- El error de tipo I consiste en rechazar incorrectamente una hipótesis nula que es verdadera.
- El error de tipo II o también conocido como falso negativo. Consiste en aceptar una hipótesis nula cuando en realidad es falsa.

Dado que las cantidades exactas, tanto de errores tipo I, como de tipo II no son observables, se debería tratar de controlar la probabilidad de cometer dichos errores dentro de unos niveles tolerables. Por lo general, las probabilidades controladas de cometer errores de tipo I y de tipo II tienen correlación negativa. Por ello se ha de establecer un compromiso adecuado según diferentes características del experimento y los objetivos del estudio. Si una conclusión significativa tiene consecuencias importantes a nivel práctico, como por ejemplo

un nuevo tratamiento, se controlará más estrictamente el error de tipo I. Por otra parte, si se busca obtener los candidatos principales para seguir investigando, como por ejemplo en genómica, se ha de evitar cometer demasiados errores de tipo II. En el caso de las comparaciones múltiples el error de tipo I, se ve incrementado considerablemente, y por lo tanto han de hacerse esfuerzos para controlarlo.

P valor ajustado o nivel de significación

En inferencia estadística, un valor de probabilidad (denominado p valor) se calcula directa o indirectamente para cada hipótesis y entonces se compara con el nivel de significación preespecificado α, para determinar si H_0 se rechaza o no. Por lo tanto, hay dos formas de ajustar la inferencia estadística de comparaciones múltiples. La primera consiste en ajustar el p valor observado para cada hipótesis, y mantener sin cambios el nivel de significación α preestablecido. A este procedimiento se le denomina p valor ajustado. Segundo,se podría determinar también a nivel computacional un corte ajustado para el α preestablecido y compararlo con el p valor observado para realizar inferencia. Por lo general, el p valor ajustado se adecua mejor, porque utiliza un nivel de significación perceptible. No obstante, en determinadas situaciones no es posible computar con exactitud el p valor ajustado.

Medidas del error tipo I

Una vez vistos los posibles resultados de las comparaciones múltiples, se han de centrar los esfuerzos en controlar la variable U. Para ello se han propuesto dos medidas estadísticas. Cada una de ellas tiene aplicaciones diferentes, y también distintas fortalezas y debilidades.

FWER En las aplicaciones prácticas, es mejor considerar conjuntamente todas las hipótesis como una familia, para controlar el error tipo I, y por tanto el criterio más exigente consiste en garantizar que ningun H_0 sea rechazado de manera incorrecta. En base a esto se define la medida tasa de error por familia (FWER) como la probabilidad de rechazar incorrectamente al menos un H_0.

$$FWER = P(U > 0)$$

FDR La Tasa de Falso Descubrimientos (FDR) es otra medida para controlar el error de tipo I en problemas de comparación múltiples. Se define como la proporción esperada de los H_0 erróneamente rechazados, de entre todos los rechazados:

$$FDR = \begin{cases} E(\frac{U}{R})\ siR > 0 \\ 0, \quad \text{si R=0} \end{cases} \tag{3.17}$$

Por tanto, la Tasa de Falso Descubrimiento, o False Discovery Rate permite la ocurrencia de errores de tipo I, en una proporción razonable, tomando en consideración el número total de rechazos. Una ventaja evidente de controlar la FDR es el importante aumento de la potencia de la inferencia estadística, que será útil cuando se contrasten simultáneamente un gran número de hipótesis.

3.2.4. Métodos comunes para el ajuste

Supóngamos que nuestro objetivo es contrastar m hipótesis $H_1, ..., H_m$ simultáneamente, y que se corresponden con los p valores inicialmente calculados de $p_1, ..., p_m$. Asimismo, los p valores ajustados de las comparaciones múltiples se denominan $p\prime_1, ..., p\prime_m$. Los niveles de significación preestablecidos serán α y $\alpha\prime$, respectivamente. Se supone además que todas las hipótesis se ordenan como $H_{(1)}, ..., H_{(m)}$ según sus p valores observados de $P(1) \leq ... \leq P(m)$; y que los p valores asociados y sus niveles de significación son $P_{(i)}, P\prime_{(i)}$ y $\alpha\prime_{(i)}$ para la i-ésima hipótesis ordenada de $H_{(i)}$.

El ajuste de Bonferroni

Es uno de los más usados para realizar comparaciones múltiples. Este ajuste intenta controlar el FWER según un criterio muy exigente y calcula los p valores ajustados, multiplicando directa el número de hipótesis contrastadas simultáneamente m.

$$p\prime_i = \{p_i \times m, 1\}(1 \leq i \leq m) \tag{3.18}$$

De forma análoga, se puede dejar los p valores observados sin modificaciones, y ajustar directamente el nivel de significación como $\alpha\prime = \alpha/m$.

Se comparan los p valores ajustados con los niveles de significación preestablecidos α. La conclusión estadística se ve alterada, de forma obvia, antes y después del ajuste. El ajuste de Bonferroni es reconocido como un método muy conservador, en especial cuando hay un elevado número de hipótesis que están siendo contrastadas simultáneamente y/o con unas hipótesis que tienen elevada correlación.

Método de Benjamini-Hochberg(BH)

De manera opuesta al estricto control de FWER, el método Benjamini y Hochberg controla FDR, que es denominado como ajuste BH.

Sea q el límite superior pre-especificado de FDR. El primer paso para calcular el índice k es:

$$k = \text{máx}\left\{i : p_{(i)} \leq \frac{i}{m}q\right\}$$

Si k no existe, no se rechaza ninguna hipótesis, en otros casos se rechazan las hipótesis de H_i para $(i = 1, ..., k)$. Este método se inicia comparando $H_{(i)}$, del mayor al menor p valor con $(i = m, , ..., 1)$. El control basado en FDR es menos estricto, teniendo un mayor aumento de la potencia, y ha sido usado en casos donde un elevado número de hipótesis se prueban simultáneamente.

Capítulo 4

COMPARACIÓN DE RESULTADOS

4.1. Introducción

El objetivo de este capitulo es doble: i) Explicar las métricas externas que se van a utilizar para evaluar los distintos algoritmos estudiados. ii) Explicar el estudio de simulación que se ha diseñado para comparar las distintas técnicas estudiadas que se comparan con los contrastes de hipótesis y procedimientos de corrección por contrastes múltiples del capítulo anterior.

4.2. Métricas externas de agrupamiento

En el análisis de grupos no se dispone de información previa del grupo de pertenencia de cada dato. De hecho, tal como se ha indicado en el capítulo 1 de este trabajo, no existe ni siquiera un consenso entre la comunidad científica acerca de cual ha de ser la definición precisa de grupo y, en consecuencia, se hace difícil establecer si aquellos que se obtienen son suficientemente satisfactorios. En muchos ocasiones, ello dependerá del problema concreto que esté planteado y de cuales sean los objetivos reales. Por ello, muchos piensan que en realidad, se trata más de un arte que una ciencia (Von Luxburg, 2012).

Ante esta problemática, se ha optado con seguir una metodología similar a la encontrada en (Li, 2015), usando algunas de las métricas contempladas en este trabajo, junto con otras de (Chang, 2010). Todas ellas son de naturaleza externa. Esto significa que para su utilización, es necesario disponer para cada dato de la información relativa a su grupo de pertenencia.

Las cinco métricas externas que pueden utilizarse son: la tasa de clasificación correcta, el índice de Rand (Rand, 1971), índice Rand Ajustado Hubbert Arabie (Hubert, 1985), el índice de Rand ajustado Morey y Agresti (Morey, 1984), el índice de Folks and Mallows (Dubes, 1988) y el índice de Jaccard. A continuación, introduciremos algunos elementos para su explicación.

Sea S un conjunto de n elementos, y sean $X = \{X_1, X_2, \ldots, X_r\}$ e $Y = \{Y_1, Y_2, \ldots, Y_s\}$ dos particiones del conjunto S anterior. El grado de coincidencia entre X y Y puede ser resumido en una tabla de contingencia (n_{ij}) en donde cada entrada n_{ij} denota el número de elementos en común entre X_i y $Y_j : n_{ij} = |X_i \cap Y_j|$.

$_X\backslash Y$	Y_1	Y_2	$\dots$	Y_s	Suna
X_1	n_{11}	n_{12}	$\dots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\dots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	a_r
Suma	b_1	b_2	$\dots$	b_s	

Utilizando la tabla anterior, en la literatura relacionada se han definido diversos estadísticos para evaluar el grado de dependencia de dos variables aleatorias de estudio X e Y categóricas, Algunos ejemplos clásicos de contrastes de hipótesis asociados a tales estadísticos pueden ser el test-chi cuadrado (Cochran, 1952) o el test-exacto de Fisher (Fisher, 1922). Para realizar un análisis de datos estadísticos categóricos, el trabajo (Agresti, 2003), constituye una referencia básica.

Aunque a través del valor de los estadísticos anteriores o incluso el p valor obtenido tras aplicar un test, se pueda evaluar el rendimiento de cada algoritmo para cada base de datos, se utilizan aquí únicamente las tablas de contingencia para representar la información de interés y después calcular las métricas citadas anteriormente.

En lo sucesivo, se asume que tanto X como Y son siempre particiones generadas bajo la condición de que, en cada familia de subconjuntos construido se encuentran siempre todos los elementos de una misma clase; y que no contienen elementos de otra clase. X se elabora a partir de un conjunto de n elementos S_{real}, en el que el elemento i-ésimo de dicho conjunto se encuentra el grupo del elemento i-ésimo del conjunto de datos, mientras en que Y, se usa el conjunto del mismo tamaño $S_{algoritmoXXX}$ que contiene en el elemento i-ésimo, la asignación que realiza el $algoritmoXXX$ para el dato i-ésimo.

S_{real} y $S_{algoritmoXXX}$ tienen los mismos elementos, no obstante el número de elementos diferentes que contiene puede ser distinto e.g (simplemente especificando en el algoritmo un número diferente de grupos al número de clases que vienen reflejadas en el conjunto de datos). En cualquier caso, en el presente trabajo, el número de grupos especificado en cada algoritmo, es igual al número de clases del conjunto de datos, y por consiguiente, $r = s$; así la tabla de contingencia tiene siempre la siguiente estructura cuadrada:

$_X\backslash Y$	Y_1	Y_2	$\dots$	Y_s	Suma
X_1	n_{11}	n_{12}	$\dots$	n_{1r}	a_1
X_2	n_{21}	n_{22}	$\dots$	n_{2r}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	a_r
Suma	b_1	b_2	$\dots$	b_r	

Los elementos n_{ij}, a_i, b_j (con $i, j \in \{1, \dots, r\}$) de la tabla de contingencia anterior, junto con n ($n = \sum_{i=1}^{r}\sum_{j=1}^{r} n_{ij}$) determinan de manera completamente unívoca que son consideradas este trabajo y que se especifican a continuación:

Índice de Rand (*Rand*):

$$Rand = \frac{n_{11} + n_{22} + n_{33} + \cdots + n_{rr}}{\binom{n}{2}}.$$

.

Índice de Rand Ajustado (Hubber) (*cRand*):

$$\overbrace{cRand}^{\text{Índice ajustado}} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Índice}} - \overbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}^{\text{Índice Esperado}}}{\underbrace{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]}_{\text{Índice Máximo}} - \underbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}_{\text{Índice Esperado}}}.$$

Índice de Fowlkes Mallows ($cFowMall$):
$cFowMall = \frac{T}{\sqrt{PQ}}$, donde:
$T = \sum_{i=1}^{r}\sum_{j=1}^{r} a_{i,j}^2 - n.$
$P = \sum_{i=1}^{r}(\sum_{j=1}^{r} a_{i,j})^2 - n.$
$Q = \sum_{j=1}^{r}(\sum_{i=1}^{r} a_{i,j})^2 - n.$
Índice de Rand Ajustado (Morey-Agresti) (cMA):
$P = \frac{TP}{TP+FP}$ $R = \frac{TP}{TP+FN}$ $F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R}$
Índice de Jaccard ($cJacc$): Para dos particiones A y B, esta métrica se obtiene como el cociente de la intersección, y la unión de ambas. $J(A,B) = \frac{TP}{TP+FP+FN}$

4.3. Procedimientos de Bondad de Ajuste en los resultados de los algoritmos de agrupamiento

Una vez establecidas las 5 métricas, se puede cuantificar el grado de similitud entre las asignaciones realizadas por cada algoritmo de agrupamiento, sobre una base de datos particular frente a la clasificación a priori que tienen asociada.

El resultado obtenido en cualquier caso para cada métrica es aleatorio ya que el resultado de ésta, se estima a partir de una muestra aleatoria $X_1, \ldots, X_n$ (no necesariamente independiente ni idénticamente distribuida). Nuestro objetivo final es comparar el rendimiento de los distintos algoritmos en un mismo escenario simulado, en donde se repetirá el mismo experimento un número B veces ($B = 1000$), generando varias distribuciones paramétricas definidas de antemano. En cada una de estas dos situaciones, es conveniente la utilización de contrastes de hipótesis para establecer si los resultados obtenidos entre los distintos algoritmos presentan diferencias estadísticamente significativas.

El problema anterior, se puede formalizar como sigue:

Sea $Rijk$ el resultado obtenido con la métrica k tras utilizar el algoritmo j en el conjunto de datos i. En donde, $i \in \{basededatos_1, \ldots, basededatos_m\}$,
$j \in \{algoritmo_1, \ldots, algoritmo_r\}$ y finalmente $k \in \{Rand, cRand, cFowMall, cMA, cJacc\}$.

4.4. Estudio de simulación

4.4.1. Descripción

En este capítulo, se resume el procedimiento llevado a cabo para comparar los 6 algoritmos seleccionados en la tabla 1 del capítulo 1. Para ello se ha realizado un estudio de simulación en el que se generan datos procedentes de dos poblaciones diferentes en los distintos casos de estudio analizados. Para ello, se ha tomado como referencia el esquema de simulación seguido en (Li, 2015), y que está recogido en la tabla 1 de este capítulo. Además se emplea para la comparación de los resultados, las métricas externas ya explicadas, junto a los procedimientos univariantes y multivariantes de bondad de ajuste explicados. Con dichos contrastes puede establecerse, si existen diferencias estadísticamente significativas entre los diferentes métodos estudiados. El umbral de significación fijado será del 5 por ciento, y en todos los casos, se aplican dos de los procedimientos de corrección por comparaciones múltiples explicados en el capítulo 4, como son Bonferroni (Bonferroni, 1936), tratado por(Advi, 2007), y Benjamini-Hochberg-BH-(Benjamini,1995), también conocido como Tasa de Falsos Descubrimientos-FDR-.

Para cada distribución especificada en la tabla anterior, se variará la dimensión de los datos generados entre 1, 2, 5, 10, 20 y 40 ($D = \{1, 2, 5, 10, 20, 40\}$) y para cada uno de los casos anteriores, se generarán 500

N^o	Distribución	Asimetría	Curtosis
1	$0{,}5N_d(0, I) + 0{,}5N_d(3, I)$	Simetría	Cola normal
2	$0{,}5N_d(0, I) + 0{,}5N_d(0, 4I)$	Simetría	Cola normal
3	$0{,}5T_d(4) + 0{,}5(T_d(4) + 3)$	Simetría	Cola pesada
4	$0{,}5T_d(4) + 0{,}5(T_d(4) + 1)$	Simetría	Cola pesada
5	$0{,}5T_d(2) + 0{,}5(T_d(2) + 3)$	Simetría	Cola pesada
6	$0{,}5T_d(2) + 0{,}5(T_d(2) + 1)$	Simetría	Cola pesada
7	$0{,}5Cubic^p(0, 1) + 0{,}5Cubic^p(0{,}3, 0{,}7)$	Larga asimetría	Cola pesada
8	$0{,}5Lognormal(0, I) + 0{,}5(Lognormal(3, I))$	Simetría	Cola pesada
9	$0{,}5Lognormal(0, I) + 0{,}5(Lognormal(0, 4I))$	Larga asimetría	Cola pesada

Cuadro 4.1: Distribuciones simuladas y sus características

escenarios ($B = 500$) donde se simulan 200 datos ($N = 200$) para cada uno de ellos. En el apéndice de este memoria, se muestra el código programado para generar una lista de R que contenga todos los datos simulados, bajo las condiciones anteriores, para cada una de las 9 distribuciones empleadas y recogidas en la tabla 4.1.

4.4.2. Clasificación de los resultados

Una vez obtenidos los resultados de los distintos algoritmos, se ha elaborado una comparación. Para ello se han generado tablas, mediante los contrastes de Kolmogorov-Smirnov (Massey, 1951) y de Wilcoxon-Mann-Whitney. Estas contienen los p valores para las métricas internas consideradas. Se evalua así la igualdad de las distintas métricas empleadas en cada cluster, en mediana y distribución.

Para cada una de las 5 métricas se detallan respectivamente las siguientes tablas:

- P-valores ks.test.
- P-valores Wilcoxon.test.
- P-valores ks.test con correcciones múltiples por el método Bonferroni.
- P-valores ks.test con correcciones múltiples por el método BH(FDR).
- P-valores Wilcoxon.test con correcciones múltiples por el método Bonferroni.
- P-valores Wilcoxon.test con correcciones múltiples por el método BH(FDR).

Se realiza aquí la discusión de los resultados obtenidos para la métrica de Jaccard y la métrica de Rand.

4.4.3. Métrica de Jaccard

Para el contraste Kolmogorov-Smirnov los p valores para fuzzy-k grupos, fuzzy-k medias, así como para EM-k grupos y EM-k medias son demasiado extremos, pues son iguales a 1. En el caso de fuzzy-EM y Hclust-espectral su $p$valor es 0.18. Para el contraste de Wilcoxon-Mann-Whitney todos los p valores son iguales a 1, siendo dichos resultados todavía más extremos. Se hace necesario pues aplicar a ambos contrastes algún método de corrección que permita reducir el número de falsos negativos obtenidos en ambos contrastes.

Contraste Kolmogorov-Smirnov con corrección de Bonferroni

Las comparaciones para Hclust-espectral y EM-fuzzy son significativas pues el contraste en ambos casos da un p valor de 0.01. Las comparaciones k medias-fuzzy y k grupos-fuzzy están en el límite de la significancia que se ha fijado, con un p valor de 0.05. Los p valores de las comparaciones entre los algoritmos k medias-EM y EM-k grupos son 0.19 y 0.27 respectivamente. No obstante para todas las comparaciones del algoritmo espectral con los algoritmos restantes se obtiene un p valor de 0, debido en parte a que el método de Bonferroni es muy conservador pues los p valores son multiplicados por el número de comparaciones.

Contraste Kolmogorov-Smirnov con corrección de BH

Con este método no se obtienen p valores extremos. La comparativa Fuzzy-EM es significativa, con p valor de 0.01.El p valor de k medias-fuzzy y de k grupos-fuzzy es 0.05, estando al límite de la significancia. La comparativa k means-EM no es significativa con un p valor de 0.19. Los p valores de las siguientes comparativas son: EM-fuzzy y el EM- spectral p valor de 0.61, k grupos-espectral, k medias-fuzzy y hclust-fuzzy 0.68. El p valor de hclust-EM es 0.69. Para k medias-EM su p valor es de 0.7. El fuzzy-espectral y el k medias-espectral tienen un p valor de 0.83, el hclust-k grupos lo tienen de 0.96. Finalmente el hclust-espectral tiene un valor de 0.99.

Contraste Wilcoxon-Mann-Whitney con corrección de Bonferroni

El contraste es significativo,para la comparativa k medias-fuzzy y k grupos-fuzzy con un p valor de 0.03. Las comparativas por orden creciente, tienen los siguientes p valores EM-k medias 0.14 , EM-k grupos 0.21, y k grupos-k medias 0.89. Dichos resultados son coherentes con lo que a priori cabe esperar de la comparación de dichos algoritmos, no obstante hay resultados que siguen requiriendo un estudio más pormenorizado.

Contraste Wilcoxon-Mann-Whitney con corrección de BH

Detallamos a continuación los p valores de las siguientes comparativas: EM-espectral es de 0.05, EM-fuzzy 0.07, k medias-fuzzy 0.18, k grupos-espectral 0.19, k grupos-fuzzy 0.21, hclust-fuzzy 0.23, EM-hclust 0.28, EM-k grupos 0.3, k medias-EM 0.35, tanto la comparación espectral-fuzzy como la k medias-espectral tienen un p valor de 0.51. La k grupos-hclust 0.64, hclust-espectral 0.71, k medias-hclust 0.84, y finalmente k medias-k grupos con un p valor de 0.95. Este resultado es coherente con el hecho de que el método k grupos es una generalización del k medias aplicado a distribuciones de probabilidad.

4.4.4. Métrica de Rand

Discutimos ahora los resultados obtenidos para la métrica de Rand. Es destacable el hecho de que los resultados tanto para el contraste Kolmogorov-Smirnov, como para el contraste Wilcoxon-Mann-Whitney son idénticos con p valores iguales a 1 para fuzzy-k medias, fuzzy-k grupos, fuzzy-EM, k medias-EM, k medias-k grupos y k grupos-EM. El resto de los valores son 0. Es necesario aplicar métodos de correcciones múltiples para diferenciar con mayor acierto.

contraste Kolmogorov-Smirnov con corrección de Bonferroni

Se obtiene resultados extremos para todas las comparaciones del algoritmo hclust con los restantes da un p valor de 0, mientras que el k medias-EM y el k medias-k grupos da un p valor igual a 1. Para los demás algoritmos se obtienen en orden ascendente los siguientes resultados. EM-fuzzy 0.23, fuzzy-k grupos 0.36, k medias-fuzzy 0.47, k medias-EM 0.82.

contraste Kolmogorov-Smirnov con corrección de BH

k means-EM tiene un p valor de 1. EM-fuzzy 0.1, k grupos-fuzzy 0.3, k medias-EM 0.69, k medias-k grupos 0.7, fuzzy-k medias y k grupos-EM 0.89.

contraste Wilcoxon-Mann-Whitney con corrección de Bonferroni

Los p valores de las comparaciones del algoritmo espectral con el resto de algoritmos son iguales a 0. EM-fuzzy es de 0.14, fuzzy-k medias 0.31, k medias-EM 0.59. Finalmente k medias-k grupos y k grupos-EM un p valor de 0.89.

contraste Wilcoxon-Mann-Whitney con corrección de BH

Los p valores de las comparaciones del algoritmo espectral con el resto de algoritmos son iguales a 0. Fuzzy-k grupos tiene un p valor de 0.17, fuzzy-EM de 0.24, k medias-EM 0.46, k medias-k grupos 0.51, fuzzy-k medias 0.71 y k grupos-EM 0.74.

4.5. Uso del valor del estadístico para la elaboración de un ranking

A continuación en las tablas del Apéndice 2, se muestran los resultados de los siguientes estadísticos en bruto, que analizamos brevemente.

4.5.1. Métrica Jaccard

Es la métrica con valores del estadístico más bajos y homogéneos para todas las comparaciones. El algoritmo que mejores resultados presenta al compararse con los algoritmos restantes es el fuzzy seguido del espectral y el HCLUST, siendo el fuzzy-EM y espectral-EM los valores más bajos, 1134 y 1160 respectivamente.

4.5.2. Métrica MA

Según los valores del estadístico, los algoritmos que mejores resultados presentan al compararse con los algoritmos restante son el espectral, HCLUST, fuzzy. Con peores resultados el k medias, el EM y k grupos.

4.5.3. Métrica Rand

Los algoritmos que mejores resultados presentan son el espectral, HCLUST, fuzzy. Siguen por orden k medias, k grupos y EM.

4.5.4. Métrica FM

Los algoritmos que mejores resultados presentan son el fuzzy, el espectral y el HCLUST. A continuación k grupos, k medias y EM.

4.5.5. Métrica HA

Los algoritmos que mejores resultados presentan son espectral, HCLUST y fuzzy. K medias, k grupos y EM presentan las peores comparativas.

En base a los valores de los estadísticos en bruto, los mejores resultados se alcanzan con el espectral, el HCLUST y el fuzzy respectivamente. Debido que la probabibilidad, en general no es transitiva, consecuentemente resulta difícil establecer cuáles métodos son mejores que otros. Por este motivo se ha pretendido establecer un marco de comparación, para así establecer cuales algoritmos son mejores. Con el fin de superar la limitación indicada, se ha optado por el uso de la aproximación asintótica del p valor mediante el contraste de Kolmogorov-Smirnov y de Wilcoxon. No obstante, dicho método tiene la fuerte restricción de que las aproximaciones dadas son bajas ante tamaños muestrales pequeños. Ello hace que la potencia de los contrastes calculados sea baja.

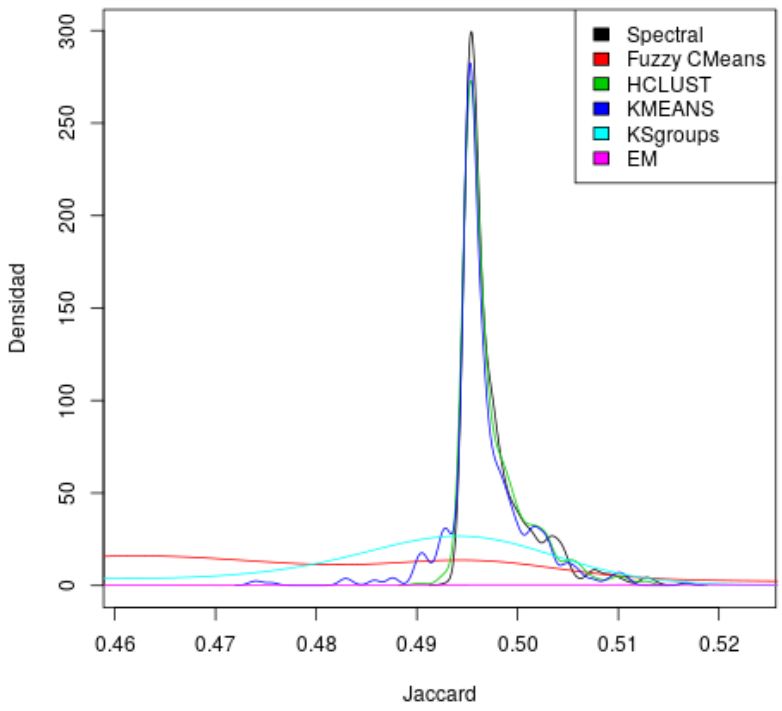

Figura 4.1: Estimación de la función de densidad para la métrica de Jaccard

4.6. El caso las mixturas de dos variables log normales con larga asimetría y colas pesadas

En esta sección vamos a analizar lo que ocurre con las 500 muestras en el caso concreto de las variables lognormales (caso de estudio 9 de la tabla anterior). En las figuras 1 y 2 se muestra la estimación no paramétrica de la función de densidad tanto para la métrica de Jaccard, como para la de Rand. En la primera gráfica se observa una gran variabilidad para el algoritmo k-grupos, quizás por los problemas para inicializar correctamente este algoritmo. Mientras que para los otros algoritmos destaca principalmente que el algoritmo EM ofrece unos resultados muy estables, donde toda la distribución de probabilidad se concentra en torno a un valor de 0,50. Finalmente para el gráfico asociado a la métrica de Rand se aprecia que el algoritmo k-grupos tiene un rendimiento un poco superior al resto, lo que es lógico al ser el único algoritmo específico para capturar la distribución subyacente bajo los datos.

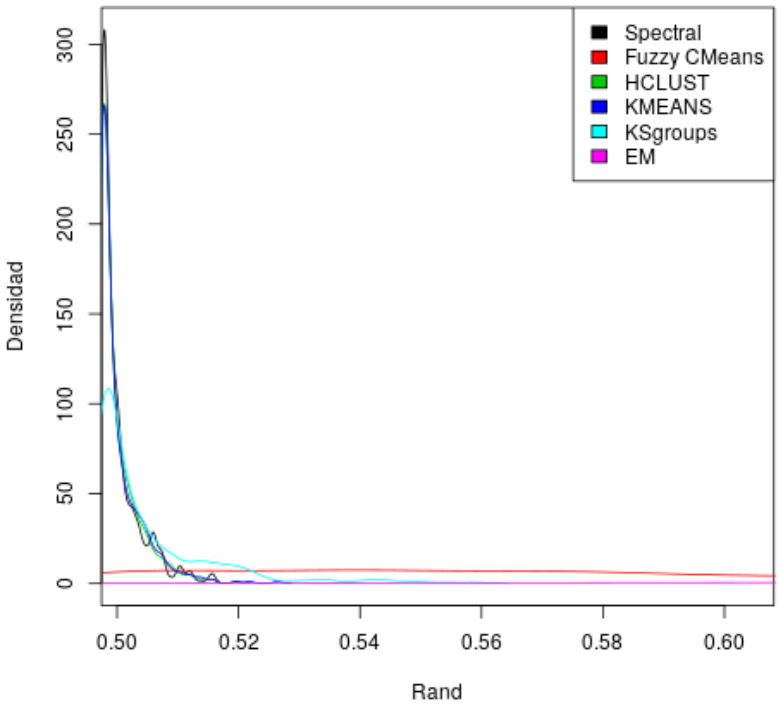

Figura 4.2: Estimación de la función de densidad para la métrica Rand

Conclusiones

En esta memoria se han presentado, por una parte los tópicos básicos del análisis cluster, se han explicado algunos de los principales algoritmos, para finalmente comparar varios de los algoritmos incluidos en el trabajo. Del estudio comparativo pueden extraerse las siguientes conclusiones:

- No existe un método superior al resto al evaluar los algoritmos para el conjunto de bases de datos que han sido simuladas.
- El k medias presenta por lo general resultados no destacables, debido a que han sido tratados datos de naturaleza no esférica.
- La dependencia del resultado de los algoritmos de clustering hace necesario la adopción de nuevas estrategias para evaluar el rendimiento de los algoritmos de clasificación y clustering.
- Quizás sea necesario en lugar de analizar la media de las métricas de estudio en diferentes escenarios, analizar separadamente cada escenario para estimar condiciones más sólidas acerca del rendimiento de los algoritmos de clustering usados.
- En todo caso, dados los objetivos introductorios de esta trabajo de fin de máster. En un futuro se contempla la posibilidad de estudiar con mayor profundidad algunos de los tópicos aquí tratados.

Apéndice A

Apéndice I

A.1. Scripts de R empleados

A.1.1. Simulación 1 $(0{,}5N_d(0, I) + 0{,}5N_d(3, I))$

```
w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){

sigma= matrix(0,ncol= j,nrow= j)
diag(sigma)= 1
mu1=  numeric(j)
mu2= numeric(j)
mu2[1:j]= 3


listaaux= 1:B
listaaux= as.list(listaaux)

indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
```

```
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rmvn(1,mu1,sigma)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
generados= rmvn(1,mu2,sigma)
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones




saveRDS(lista,"normal1.RDS")
```

A.1.2. Simulación 2 $(0{,}5N_d(0, I) + 0{,}5N_d(0, 4I))$

```
w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")
```

```
for(i in N){

for(j in d){

sigma1= matrix(0,ncol= j,nrow= j)
diag(sigma1)= 1

sigma2= matrix(0,ncol= j, nrow= j)

diag(sigma2)= 4

mu=  numeric(j)



listaaux= 1:B
listaaux= as.list(listaaux)

indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rmvn(1,mu,sigma1)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
etiqueta= 2
generados= rmvn(1,mu,sigma2)
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones



saveRDS(lista,"normal2.RDS")
```

A.1.3. Simulación 3 $0{,}5T_d(4) + 0{,}5(T_d(4) + 3)$

```
w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){

sigma= matrix(0,ncol= j,nrow= j)
diag(sigma)= 1
mu1=  numeric(j)
mu2= numeric(j)
mu2[1:j]= 3
grados= 4


listaaux= 1:B
listaaux= as.list(listaaux)

indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rmvt(1,delta=mu1,sigma=sigma,df= grados)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
generados= rmvt(1,mu2,sigma= sigma,df= grados)
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos
```

```
}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones




saveRDS(lista,"student1.RDS")
```

A.1.4. Simulación 4 $0{,}5T_d(4) + 0{,}5(T_d(4) + 1)$

```
w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){

sigma= matrix(0,ncol= j,nrow= j)
diag(sigma)= 1
mu1=  numeric(j)
mu2= numeric(j)
mu2[1:j]= 1
grados= 4


listaaux= 1:B
listaaux= as.list(listaaux)
```

```
indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rmvt(1,delta=mu1,sigma=sigma,df= grados)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
generados= rmvt(1,mu2,sigma= sigma,df= grados)
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones




saveRDS(lista,"student2.RDS")
```

A.1.5. Simulación 5 $0{,}5T_d(2) + 0{,}5(T_d(2) + 3)$

```
w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
```

```
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){

sigma= matrix(0,ncol= j,nrow= j)
diag(sigma)= 1
mu1=  numeric(j)
mu2= numeric(j)
mu2[1:j]= 3
grados= 2


listaaux= 1:B
listaaux= as.list(listaaux)

indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rmvt(1,delta=mu1,sigma=sigma,df= grados)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
generados= rmvt(1,mu2,sigma= sigma,df= grados)
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones
```

```
saveRDS(lista,"student3.RDS")
```

A.1.6. Simulación 6 $0{,}5T_d(2) + 0{,}5(T_d(2) + 1)$

```
w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){

sigma= matrix(0,ncol= j,nrow= j)
diag(sigma)= 1
mu1=  numeric(j)
mu2= numeric(j)
mu2[1:j]= 1
grados= 2


listaaux= 1:B
listaaux= as.list(listaaux)

indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rmvt(1,delta=mu1,sigma=sigma,df= grados)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
```

```
generados= rmvt(1,mu2,sigma= sigma,df= grados)
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones




saveRDS(lista,"student4.RDS")
```

A.1.7. Simulación 7 $0{,}5Lognormal(0, I) + 0{,}5(Lognormal(3, I))$

```
library("compositions")


w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){
```

```
sigma= matrix(0,ncol= j,nrow= j)
diag(sigma)= 1
mu1=  numeric(j)
mu2= numeric(j)
mu2[1:j]= 3


listaaux= 1:B
listaaux= as.list(listaaux)


indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rlnorm.rplus(1,mu1,sigma)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
generados= rlnorm.rplus(1,mu2,sigma)
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones




saveRDS(lista,"lognormal1.RDS")
```

A.1.8. Simulación 8 $0{,}5Lognormal(0, I) + 0{,}5(Lognormal(0, 4I))$

```
w= c(0.5)
B= 1000
```

```
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){

sigma1= matrix(0,ncol= j,nrow= j)
diag(sigma1)= 1
sigma2= matrix(0,ncol= j, nrow= j)
diag(sigma2)= 4
mu= numeric(j)

listaaux= 1:B
listaaux= as.list(listaaux)

indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= rlnorm.rplus(1,mu,sigma1)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
generados= rlnorm.rplus(1,mu, sigma2)
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux
```

```
}

}

names(lista)= textocombinaciones




saveRDS(lista,"lognormal2.RDS")
```

A.1.9. Simulación 9 $0{,}5Cubic^p(0,1) + 0{,}5Cubic^p(0{,}3, 0{,}7)$

```
w= c(0.5)
B= 1000
N= c(200,500)
d= c(1,2,5,10,20,40)

N= N[1]

iter= length(N)*length(d)

lista= 1:iter

lista= as.list(lista)
combinaciones= expand.grid(N,d)
orden= order(combinaciones[,1])
combinaciones=  combinaciones[orden,]

textocombinaciones= paste(combinaciones[,1],combinaciones[,2],sep="/")

for(i in N){

for(j in d){


listaaux= 1:B
listaaux= as.list(listaaux)

indice= 6*(which(N==i)-1)+which(d==j)

for(z in 1:B){
auxdatos= matrix(0,ncol=j+1,nrow= i)
for(s in 1:i){

unif= runif(1)
if(unif<w){
etiqueta= 1
generados= runif(j,0,1)
auxdatos[s,j+1]= etiqueta
auxdatos[s,1:j]= generados
}else{
generados=  runif(j,0.3,0.7)
```

```
etiqueta= 2
auxdatos[s,1:j]= generados
auxdatos[s,j+1]= etiqueta
}

listaaux[[z]]= auxdatos

}
}
lista[[indice]]= listaaux




}

}

names(lista)= textocombinaciones




saveRDS(lista,"uniforme.RDS")
```

A.1.10. Realización de contrastes y comparaciones múltiples

```
library("dgof")
library("xtable")
setwd("C:/Users/Usuario/Desktop/tablasVie27/SALIDA")
archivos=c("Jaccardfinal.csv","MAfinal.csv","Randfinal.csv","FMfinal.csv","HAfinal.csv")
archivos2= paste("nuevo",archivos,sep="")



contar=0
for(i in archivos){
datos= read.csv(i,sep=",")
dim(datos)
print(dim(datos))

nombres= datos[,1]
indices= c(seq(0,324,by=54)+1)[1:6]
nombres[indices]

nombres= c("spectral","fuzzycmeans","HCLUST","KMEANS","ksgroups","EM")



print(i)

indices= 1:6
const= 54
lista= indices*const
ind= 1:const
```

```
prim=c(ind)
seg= c(ind+lista[1])
ter= c(ind+lista[[2]])
cuar= c(ind+lista[[3]])
cinc= c(ind+lista[[4]])
sex= c(ind+lista[[5]])


info=data.frame(datos[prim,502],datos[seg,502],datos[ter,502],datos[cuar,502],
datos[cinc,502],datos[sex,502])\\
info2=data.frame(datos[prim,503],datos[seg,503],datos[ter,503],datos[cuar,503],
datos[cinc,503],datos[sex,503])\\




pvalores= matrix(0,ncol=6,nrow=6)
pvalores2= matrix(0,ncol=6,nrow=6)
for(z in nombres){
for(i in 1:6){
for(j in 1:6){
pvalores[i,j]= ks.test(info[,i],info[,j])$p.value
pvalores2[i,j]= wilcox.test(info[,i],info[,j])$p.value

}
}
}
colnames(pvalores)= nombres
rownames(pvalores)= nombres


colnames(pvalores2)= nombres
rownames(pvalores2)= nombres




method=c("bonferroni","BH")
bonferronicorrecionespvalores=matrix(p.adjust(as.vector(pvalores),method=method[1]),ncol=6,
nrow=6, byrow = TRUE)\\
fdrcorrecionespvalores=matrix(p.adjust(as.vector(pvalores),method=method[2]),ncol=6,nrow=6,
byrow = TRUE)\\

bonferronicorrecionespvalores2=matrix(p.adjust(as.vector(pvalores2),method = method[1]),
ncol=6,nrow=6, byrow = TRUE)\\
fdrcorrecionespvalores2=matrix(p.adjust(as.vector(pvalores2),method = method[2]),
ncol=6,nrow=6,byrow = TRUE)\\

colnames(bonferronicorrecionespvalores)= nombres
rownames(fdrcorrecionespvalores)= nombres


colnames(bonferronicorrecionespvalores2)= nombres
rownames(fdrcorrecionespvalores2)= nombres
```

```
print(xtable(bonferronicorrecionespvalores))
print(xtable(bonferronicorrecionespvalores2))

print(xtable(fdrcorrecionespvalores))

print(xtable(fdrcorrecionespvalores2))



print(xtable(pvalores))

print(xtable(pvalores2))

print("-------------------------------------------------------")

}
```

A.1.11. Obtención de estadísticos para comparativa final

```
library("xtable")
setwd("C:/Users/Usuario/Desktop/tablasVie27/SALIDA")
archivos=c("Jaccardfinal.csv","MAfinal.csv","Randfinal.csv","FMfinal.csv","HAfinal.csv")\\
archivos2= paste("nuevo",archivos,sep="")



contar=0
for(i in archivos){
datos= read.csv(i,sep=",")
dim(datos)
print(dim(datos))

nombres= datos[,1]
indices= c(seq(0,324,by=54)+1)[1:6]
nombres[indices]

nombres= c("spectral","fuzzycmeans","HCLUST","KMEANS","ksgroups","EM")


print(i)

indices= 1:6
const= 54
lista= indices*const
ind= 1:const
prim=c(ind)
seg= c(ind+lista[1])
ter= c(ind+lista[[2]])
cuar= c(ind+lista[[3]])
cinc= c(ind+lista[[4]])
sex= c(ind+lista[[5]])


info= data.frame(datos[prim,502],datos[seg,502],datos[ter,502],datos[cuar,502],
datos[cinc,502],datos[sex,502])\\
```

```
info2=data.frame(datos[prim,503],datos[seg,503],datos[ter,503],datos[cuar,503],
datos[cinc,503],datos[sex,503])\\




pvalores= matrix(0,ncol=6,nrow=6)
pvalores2= matrix(0,ncol=6,nrow=6)
for(z in nombres){
for(i in 1:6){
for(j in 1:6){
pvalores[i,j]= ks.test(info[,i],info[,j])$statistic
pvalores2[i,j]= wilcox.test(info[,i],info[,j])$statistic

}
}
}
colnames(pvalores)= nombres
rownames(pvalores)= nombres


colnames(pvalores2)= nombres
rownames(pvalores2)= nombres


print(xtable(pvalores))
print(xtable(pvalores))

method=c("bonferroni","BH")
bonferronicorrecionespvalores=matrix(p.adjust(as.vector(pvalores),method = method[1]),
ncol=6,nrow=6, byrow = TRUE)\\
fdrcorrecionespvalores=matrix(p.adjust(as.vector(pvalores),method = method[2]),
ncol=6,nrow=6,byrow = TRUE)\\

bonferronicorrecionespvalores2=matrix(p.adjust(as.vector(pvalores2),method = method[1]),
ncol=6,nrow=6, byrow = TRUE)\\
fdrcorrecionespvalores2=matrix(p.adjust(as.vector(pvalores2),method = method[2]),
ncol=6,nrow=6,byrow = TRUE)\\

colnames(bonferronicorrecionespvalores)= nombres
rownames(fdrcorrecionespvalores)= nombres


colnames(bonferronicorrecionespvalores2)= nombres
rownames(fdrcorrecionespvalores2)= nombres


print(xtable(bonferronicorrecionespvalores))
print(xtable(bonferronicorrecionespvalores2))

print(xtable(fdrcorrecionespvalores))

print(xtable(fdrcorrecionespvalores2))
```

```
print(xtable(pvalores))

print(xtable(pvalores2))

print("-----------------------------------------------------")

}

}
```

A.1.12. Comandos para generar la función de densidad

```
library("xtable")
setwd("~/SALIDA")
archivos=c("Jaccardfinal.csv","MAfinal.csv","Randfinal.csv","FMfinal.csv","HAfinal.csv")
archivos2= paste("nuevo",archivos,sep="")

# archivos= archivos[[1]]

titulo= c("Jaccard","MA","Rand","FM","HA")

# archivos
# título

contar=0
for(i in archivos){
        datos= read.csv(i,sep=",")
        dim(datos)
        print(dim(datos))

        nombres= datos[,1]
        indices= c(seq(0,324,by=54)+1)[1:6]
        nombres[indices]

        nombres= c("spectral","fuzzycmeans","HCLUST","KMEANS","ksgroups","EM")



        print(i)

        indices= 1:6
        const= 54
        lista= indices*const
        ind= 1:const
        prim=c(ind)
        seg= c(ind+lista[1])
        ter= c(ind+lista[[2]])
        cuar= c(ind+lista[[3]])
        cinc= c(ind+lista[[4]])
        sex= c(ind+lista[[5]])


        info=data.frame(datos[prim,502],datos[seg,502],datos[ter,502],datos[cuar,502],
        datos[cinc,502],datos[sex,502])\\
        info2= data.frame(datos[prim,503],datos[seg,503],datos[ter,503],datos[cuar,503],
```

```
        datos[cinc,503],datos[sex,503])\\


        indx=seq(54,324,by=54)
        datos2=data.frame(as.numeric(t(datos[indx[1],2:501])),as.numeric(t(datos[indx[2],
        2:501])),as.numeric(t(datos[indx[3],2:501])),as.numeric(t(datos[indx[4],2:501])),
        as.numeric(t(datos[indx[5],2:501])), as.numeric(t(datos[indx[6],2:501])))\\
        colnames(datos2)= c("Spectral","Fuzzy␣CMeans","HCLUST","KMEANS","KSgroups","EM")

        contar= contar+1


        dens <- apply(datos2, 2, density)

        auxtitulo= paste(titulo[contar],".png",sep="")
        dev.copy(png,auxtitulo)

        plot(NA, xlim=as.numeric(quantile(unlist(sapply(dens, "[", "x")),probs= c(0.2,0.8))),

        mapply(lines, dens, col=1:length(dens))

        legend("topright", legend=names(dens), fill=1:length(dens))


        dev.off()


}
```

Apéndice A

Apéndice II

A.1. Tablas obtenidas

Como ya se ha indicado en el capítulo 4, se detallan a continuación, para cada una de las 5 métricas las siguientes tablas:

- P-valores ks.test.
- P-valores Wilcoxon.test.
- P-valores ks.test con correcciones múltiples por el método Bonferroni.
- P-valores ks.test con correcciones múltiples por el método BH(FDR).
- P-valores Wilcoxon.test con correcciones múltiples por el método Bonferroni.
- P-valores Wilcoxon.test con correcciones múltiples por el método BH(FDR).

A.1.1. P valores de cada métrica

Cuadro A.1: Jaccardfinal.csv

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.18	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	0.18
HCLUST	0.18	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	1.00	0.00	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	0.18	0.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	1.00	1.00	1.00	1.00	1.00
fuzzycmeans	1.00	1.00	1.00	1.00	1.00	1.00
HCLUST	1.00	1.00	1.00	1.00	1.00	1.00
KMEANS	1.00	1.00	1.00	1.00	1.00	1.00
ksgroups	1.00	1.00	1.00	1.00	1.00	1.00
EM	1.00	1.00	1.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.01	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.05	0.05	0.01
HCLUST	0.01	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.05	0.00	1.00	1.00	0.19
ksgroups	0.00	0.05	0.00	1.00	1.00	0.27
EM	0.00	0.01	0.00	0.19	0.27	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.83	0.99	0.83	0.68	0.61
fuzzycmeans	0.83	1.00	0.68	0.68	0.68	0.61
HCLUST	0.99	0.68	1.00	1.00	0.96	0.69
KMEANS	0.83	0.68	1.00	1.00	1.00	0.70
ksgroups	0.68	0.68	0.96	1.00	1.00	0.69
EM	0.61	0.61	0.69	0.70	0.69	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.03	0.03	0.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.03	0.00	1.00	0.89	0.14
ksgroups	0.00	0.03	0.00	0.89	1.00	0.21
EM	0.00	0.00	0.00	0.14	0.21	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.51	0.71	0.51	0.19	0.05
fuzzycmeans	0.51	1.00	0.23	0.18	0.21	0.07
HCLUST	0.71	0.23	1.00	0.84	0.64	0.28
KMEANS	0.51	0.18	0.84	1.00	0.95	0.35
ksgroups	0.19	0.21	0.64	0.95	1.00	0.31
EM	0.05	0.07	0.28	0.35	0.31	1.00

Cuadro A.2: MAfinal.csv

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	1.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	1.00	0.00	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	1.00	0.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	1.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	1.00	0.00	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	1.00	0.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.47	0.06	0.23
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.47	0.00	1.00	1.00	0.82
ksgroups	0.00	0.06	0.00	1.00	1.00	1.00
EM	0.00	0.23	0.00	0.82	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.89	0.30	0.40
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.89	0.00	1.00	0.70	0.69
ksgroups	0.00	0.30	0.00	0.70	1.00	0.89
EM	0.00	0.40	0.00	0.69	0.89	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.31	0.03	0.14
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.31	0.00	1.00	0.89	0.59
ksgroups	0.00	0.03	0.00	0.89	1.00	0.89
EM	0.00	0.14	0.00	0.59	0.89	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.71	0.17	0.24
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.71	0.00	1.00	0.51	0.46
ksgroups	0.00	0.17	0.00	0.51	1.00	0.74
EM	0.00	0.24	0.00	0.46	0.74	1.00

Cuadro A.3: Randfinal.csv

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	1.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	1.00	0.00	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	1.00	0.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	1.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	1.00	0.00	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	1.00	0.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.32	0.06	0.14
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.32	0.00	1.00	0.97	0.82
ksgroups	0.00	0.06	0.00	0.97	1.00	1.00
EM	0.00	0.14	0.00	0.82	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.61	0.31	0.32
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.61	0.00	1.00	0.61	0.52
ksgroups	0.00	0.31	0.00	0.61	1.00	0.72
EM	0.00	0.32	0.00	0.52	0.72	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.21	0.03	0.09
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.21	0.00	1.00	0.75	0.59
ksgroups	0.00	0.03	0.00	0.75	1.00	0.89
EM	0.00	0.09	0.00	0.59	0.89	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.46	0.17	0.20
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.46	0.00	1.00	0.48	0.35
ksgroups	0.00	0.17	0.00	0.48	1.00	0.60
EM	0.00	0.20	0.00	0.35	0.60	1.00

Cuadro A.4: FMfinal.csv

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.04	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	0.09
HCLUST	0.04	0.00	1.00	0.02	0.00	0.01
KMEANS	0.00	1.00	0.02	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	0.09	0.01	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	1.00	1.00	1.00	1.00	1.00
fuzzycmeans	1.00	1.00	1.00	1.00	1.00	1.00
HCLUST	1.00	1.00	1.00	1.00	1.00	1.00
KMEANS	1.00	1.00	1.00	1.00	1.00	1.00
ksgroups	1.00	1.00	1.00	1.00	1.00	1.00
EM	1.00	1.00	1.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.05	0.05	0.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.05	0.00	1.00	1.00	0.19
ksgroups	0.00	0.05	0.00	1.00	1.00	0.27
EM	0.00	0.00	0.00	0.19	0.27	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.80	0.83	0.83	0.83	0.83
fuzzycmeans	0.80	1.00	0.69	0.80	0.80	0.69
HCLUST	0.83	0.69	1.00	1.00	1.00	1.00
KMEANS	0.83	0.80	1.00	1.00	1.00	0.83
ksgroups	0.83	0.80	1.00	1.00	1.00	0.83
EM	0.83	0.69	1.00	0.83	0.83	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.03	0.03	0.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.03	0.00	1.00	0.89	0.14
ksgroups	0.00	0.03	0.00	0.89	1.00	0.21
EM	0.00	0.00	0.00	0.14	0.21	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.19	0.50	0.51	0.51	0.51
fuzzycmeans	0.19	1.00	0.08	0.15	0.22	0.06
HCLUST	0.50	0.08	1.00	0.88	0.93	0.98
KMEANS	0.51	0.15	0.88	1.00	0.97	0.40
ksgroups	0.51	0.22	0.93	0.97	1.00	0.31
EM	0.51	0.06	0.98	0.40	0.31	1.00

Cuadro A.5: HAfinal.csv

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	1.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	1.00	0.00	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	1.00	0.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	1.00	1.00	1.00
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	1.00	0.00	1.00	1.00	1.00
ksgroups	0.00	1.00	0.00	1.00	1.00	1.00
EM	0.00	1.00	0.00	1.00	1.00	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.47	0.06	0.14
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.47	0.00	1.00	1.00	0.82
ksgroups	0.00	0.06	0.00	1.00	1.00	0.97
EM	0.00	0.14	0.00	0.82	0.97	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.62	0.31	0.37
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.62	0.00	1.00	0.62	0.62
ksgroups	0.00	0.31	0.00	0.62	1.00	0.83
EM	0.00	0.37	0.00	0.62	0.83	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.31	0.03	0.09
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.31	0.00	1.00	0.89	0.59
ksgroups	0.00	0.03	0.00	0.89	1.00	0.75
EM	0.00	0.09	0.00	0.59	0.75	1.00

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1.00	0.00	0.00	0.00	0.00	0.00
fuzzycmeans	0.00	1.00	0.00	0.47	0.17	0.23
HCLUST	0.00	0.00	1.00	0.00	0.00	0.00
KMEANS	0.00	0.47	0.00	1.00	0.48	0.43
ksgroups	0.00	0.17	0.00	0.48	1.00	0.69
EM	0.00	0.23	0.00	0.43	0.69	1.00

Estadísticos finales

Cuadro A.6: "Jaccardfinal.csv"

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1458.00	1566.00	1518.00	1350.00	1242.00	1134.00
fuzzycmeans	1350.00	1458.00	1261.00	1239.00	1251.50	1160.00
HCLUST	1398.00	1655.00	1458.00	1424.00	1382.00	1280.00
KMEANS	1566.00	1677.00	1492.00	1458.00	1446.50	1305.00
ksgroups	1674.00	1664.50	1534.00	1469.50	1458.00	1291.00
EM	1782.00	1756.00	1636.00	1611.00	1625.00	1458.00

Cuadro A.7: "MAfinal.csv"

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1458.00	0.00	380.00	0.00	0.00	161.00
fuzzycmeans	2916.00	1458.00	2469.00	1398.00	1232.50	1267.00
HCLUST	2536.00	447.00	1458.00	411.00	367.00	509.00
KMEANS	2916.00	1518.00	2505.00	1458.00	1349.50	1337.00
ksgroups	2916.00	1683.50	2549.00	1566.50	1458.00	1403.00
EM	2755.00	1649.00	2407.00	1579.00	1513.00	1458.00

Cuadro A.8: Randfinal.csv"

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1458.00	266.00	532.00	4.00	0.00	216.00
fuzzycmeans	2650.00	1458.00	2141.00	1336.00	1235.50	1248.00
HCLUST	2384.00	775.00	1458.00	429.00	385.00	516.00
KMEANS	2912.00	1580.00	2487.00	1458.00	1341.50	1304.00
ksgroups	2916.00	1680.50	2531.00	1574.50	1458.00	1372.00
EM	2700.00	1668.00	2400.00	1612.00	1544.00	1458.00

Cuadro A.9: "FMfinal.csv"

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1458.00	1674.00	1568.00	1350.00	1350.00	1350.00
fuzzycmeans	1242.00	1458.00	1169.00	1223.00	1258.50	1149.00
HCLUST	1348.00	1747.00	1458.00	1433.00	1472.00	1453.00
KMEANS	1566.00	1693.00	1483.00	1458.00	1464.50	1321.00
ksgroups	1566.00	1657.50	1444.00	1451.50	1458.00	1291.00
EM	1566.00	1767.00	1463.00	1595.00	1625.00	1458.00

Cuadro A.10: "HAfinal.csv"

	spectral	fuzzycmeans	HCLUST	KMEANS	ksgroups	EM
spectral	1458.00	257.00	394.00	0.00	0.00	216.00
fuzzycmeans	2659.00	1458.00	2120.00	1339.00	1234.50	1261.00
HCLUST	2522.00	796.00	1458.00	425.00	385.00	523.00
KMEANS	2916.00	1577.00	2491.00	1458.00	1342.50	1328.00
ksgroups	2916.00	1681.50	2531.00	1573.50	1458.00	1393.00
EM	2700.00	1655.00	2393.00	1588.00	1523.00	1458.00

Bibliografía

[1] Hervé Abdi. Bonferroni and šidák corrections for multiple comparisons. *Encyclopedia of measurement and statistics*, 3:103–107, 2007.

[2] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.

[3] Alan Agresti. *Categorical data analysis*, volume 482. John Wiley & Sons, 2003.

[4] Michael R Anderberg. Cluster analysis for applications. Technical report, Office of the Assistant for Study Support Kirtland AFB N MEX, 1973.

[5] Adelchi Azzalini and Nicola Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80, 2007.

[6] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300. ACM, 2010.

[7] R Bellman. Curse of dimensionality. *Adaptive control processes: a guided tour. Princeton, NJ*, 1961.

[8] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[9] Michael W Berry and Malu Castellanos. Survey of text mining. *Computing Reviews*, 45(9):548, 2004.

[10] James C Bezdek, Chris Coray, Robert Gunderson, and James Watson. Detection and characterization of cluster substructure i. linear structure: Fuzzy c-lines. *SIAM Journal on Applied Mathematics*, 40(2):339–357, 1981.

[11] Emanuel Joseph Bijnen. *Cluster analysis: Survey and evaluation of techniques*, volume 1. [Tilburg]: Tilburg University Press, 1973.

[12] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[13] Hans Hermann Bock. *Automatische Klassifikation: Theoret. u. prakt. Methoden z. Gruppierung u. Strukturierung von Daten (Cluster-Analyse)*, volume 24. Vandenhoeck & Ruprecht, 1974.

[14] Hans-Hermann Bock. On some significance tests in cluster analysis. *Journal of classification*, 2(1):77–108, 1985.

[15] Dibya Jyoti Bora, Dr Gupta, and Anil Kumar. A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *arXiv preprint arXiv:1404.6059*, 2014.

[16] Fang Chang, Weiliang Qiu, Ruben H Zamar, Ross Lazarus, Xiaogang Wang, et al. clues: an r package for nonparametric clustering based on local shrinking. *Journal of Statistical Software*, 33(4):1–16, 2010.

[17] Matthew C Clark, Lawrence O Hall, Dmitry B Goldgof, Laurence P Clarke, Robert P Velthuizen, and Martin S Silbiger. Mri segmentation using fuzzy clustering techniques. *IEEE Engineering in Medicine and Biology Magazine*, 13(5):730–742, 1994.

[18] William G Cochran. The $\chi 2$ test of goodness of fit. *The Annals of Mathematical Statistics*, pages 315–345, 1952.

[19] Douglas R Cox. Note on grouping. *Journal of the American Statistical Association*, 52(280):543–547, 1957.

[20] Tingting Cui and Fangshi Li. Weight computing in competitive k-means algorithm. In *Computing, Communications and Applications Conference (ComComAp), 2012*, pages 430–435. IEEE, 2012.

[21] David Dohan, Stefani Karp, and Brian Matejek. K-median algorithms: theory in practice, 2015.

[22] William E Donath and Alan J Hoffman. Lower bounds for the partitioning of graphs. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 437–442. World Scientific, 2003.

[23] Richard C Dubes and Anil K Jain. Algorithms for clustering data, 1988.

[24] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[25] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, 4(2):81–173, 2011.

[26] Laszlo Engelman and John A Hartigan. Percentage points of a test for clusters. *Journal of the American Statistical Association*, 64(328):1647–1648, 1969.

[27] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[28] E Diday et Collaborateurs. Optimisation en classification automatique. *INRIA, Le Chesnay, France*, 1979.

[29] Sabhia Firdaus and Md Ashraf Uddin. A survey on clustering algorithms and complexity analysis. *International Journal of Computer Science Issues (IJCSI)*, 12(2):62, 2015.

[30] Ronald A Fisher. On the interpretation of χ 2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

[31] Walter D Fisher. On grouping for maximum homogeneity. *Journal of the American statistical Association*, 53(284):789–798, 1958.

[32] Chris Fraley and Adrian E Raftery. Mclust version 3: an r package for normal mixture modeling and model-based clustering. Technical report, WASHINGTON UNIV SEATTLE DEPT OF STATISTICS, 2006.

[33] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[34] Martin Grötschel and Yoshiko Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1-3):59–96, 1989.

[35] Sudipto Guha and Nina Mishra. Clustering data streams. In *Data Stream Management*, pages 169–187. Springer, 2016.

[36] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1-3):191–215, 1997.

[37] John A Hartigan. Cluster algorithms. *John Wiley & Sons). IRF Scientific Report*, 214:1993, 1975.

[38] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[39] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[40] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.

[41] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

[42] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[43] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[44] Mohammad Imdad and Javid Ali. A general fixed point theorem in fuzzy metric spaces via an implicit function. *Journal of Applied Mathematics & Informatics*, 26(3_4):591–603, 2008.

[45] Nicholas Jardine and Robin Sibson. Mathematical taxonomy. *London etc.: John Wiley*, 1971.

[46] Robert E Jensen. A dynamic programming algorithm for cluster analysis. *Operations Research*, 17(6):1034–1057, 1969.

[47] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[48] Ie Lerman. Les bases de la classification automatique. 1970.

[49] Songzi Li. *K-groups: A generalization of K-means by energy distance*. Bowling Green State University, 2015.

[50] Maxwell W Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321, 2015.

[51] Sebastian Lühr and Mihai Lazarescu. Incremental clustering of dynamic data streams using connectivity based representative points. *Data & Knowledge Engineering*, 68(1):1–27, 2009.

[52] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45, 2004.

[53] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[54] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[55] Leslie C Morey and Alan Agresti. The measurement of classification agreement: An adjustment to the rand statistic for chance agreement. *Educational and Psychological Measurement*, 44(1):33–37, 1984.

[56] Daniel Müllner et al. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(9):1–18, 2013.

[57] John M Mulvey and Harlan P Crowder. Cluster analysis: An application of lagrangian relaxation. *Management Science*, 25(4):329–340, 1979.

[58] Georgios Paltoglou and Mike Thelwall. Twitter, myspace, digg: Unsupervised sentiment analysis in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):66, 2012.

[59] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

[60] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[61] David Pollard et al. A central limit theorem for k-means clustering. *The Annals of Probability*, 10(4):919–926, 1982.

[62] Girish Punj and David W Stewart. Cluster analysis in marketing research: Review and suggestions for application. *Journal of marketing research*, pages 134–148, 1983.

[63] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[64] MR Rao. Cluster analysis and mathematical programming. *Journal of the American statistical association*, 66(335):622–626, 1971.

[65] Kwangil Ro, Changliang Zou, Zhaojun Wang, and Guosheng Yin. Outlier detection for high-dimensional data. *Biometrika*, 102(3):589–599, 2015.

[66] Stephen J Roberts. Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2):261–272, 1997.

[67] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[68] Wolfgang Sodeur. *Empirische Verfahren zur Klassifikation*, volume 42. Springer-Verlag, 2013.

[69] RR Sokal and PHA Sneath. ?principles of numerical taxonomy? freeman. *San Francisco–London*, 1963.

[70] R Suganya and R Shanthi. Fuzzy c-means algorithm-a review. *International Journal of Scientific and Research Publications*, 2(11):1, 2012.

[71] Ryota Suzuki and Hidetoshi Shimodaira. Pvclust: an r package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, 22(12):1540–1542, 2006.

[72] Mitchell D Swanson, Mei Kobayashi, and Ahmed H Tewfik. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 86(6):1064–1087, 1998.

[73] Gábor J Székely and Maria L Rizzo. The energy of data. *Annual Review of Statistics and Its Application*, 4:447–479, 2017.

[74] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The annals of statistics*, pages 2769–2794, 2007.

[75] Alexandre B Tsybakov. Introduction to nonparametric estimation. revised and extended from the 2004 french original. translated by vladimir zaiats, 2009.

[76] Bart Jan Van Os. *Dynamic programming for partitioning in multivariate data analysis*. 2000.

[77] Hrishikesh D Vinod. Integer programming and the theory of grouping. *Journal of the American Statistical association*, 64(326):506–519, 1969.

[78] F Vogel. Probleme und verfahren der numerischen klassifikation, göttingen 1975. *Google Scholar*.

[79] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[80] Ulrike Von Luxburg, Robert C Williamson, and Isabelle Guyon. Clustering: Science or art? In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 65–79, 2012.

[81] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[82] Michel Wedel and Wagner A Kamakura. *Market segmentation: Conceptual and methodological foundations*, volume 8. Springer Science & Business Media, 2012.

[83] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[84] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[85] Makoto Yamada, Koh Takeuchi, Tomoharu Iwata, John Shawe-Taylor, and Samuel Kaski. Localized lasso for high-dimensional regression. *arXiv preprint arXiv:1603.06743*, 2016.

[86] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in neural information processing systems*, pages 1569–1576, 2005.

[87] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006.

[88] Székely, Gabor J and Rizzo, Maria L Hierarchical clustering via joint between-within distances: Extending Ward's minimum variance method. *Journal of classification*, 22(2):151–183, 2005.

[89] Alejandro Cholaquidis, Ricardo Fraimand, and Mariela Sued. Semi-supervised learning: When and why it works. *arXiv preprint arXiv:1805.09180*,2018.

[90] Spath, Helmuth. Cluster analysis algorithms for data reduction and classification of objects. *Horwood*,1980

[91] Dempster, Arthur P and Laird, Nan M and Rubin, Donald B Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society*:1–38,1977.

[92] MacKay, David JC and Mac Kay, David JC Information theory, inference and learning algorithms. *Cambridge university press* ,2003.

[93] Bonferroni,C.E Teoría statistica delle classi e calcolo delle probabilitá . *Pubblicazioni del R Instituto Superiore di Scienze Echonomiche e Commerciali di Firenze*:8, 3-62, 1936.