USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Revisión e comparativa de metodoloxías para a formación de grupos

Mariña Vila Blanco

2024/2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Revisión e comparativa de metodoloxías para a formación de grupos

Mariña Vila Blanco

Setembro, 2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Traballo proposto

Área de Coñecemento: Estatística e investigación operativa
Título: Revisión e comparativa de metodoloxías para formación de grupos
Breve descrición do contido
No ámbito da análise de datos, as técnicas de formación de grupos (clustering) destacan como metodoloxías fundamentais para identificar patróns e estruturas dentro de conxuntos de observacións e teñen aplicacións en numerosos campos. O obxectivo deste traballo é facer unha revisión e comparativa de diferentes metodoloxías de clustering existentes na literatura, desde os métodos baseados en distancias entre os que se atopa o algoritmo k-medias (un dos métodos de agrupamento máis populares e empregados), ata métodos baseados en modelos paramétricos, como mixturas de normais ou métodos baseados en densidade, como DBSCAN.
Recomendacións
Outras observacións

Índice

Resumo **VIII**

Introdución **XI**

1. Clustering particional **1**

1.1. k-medias 1

1.1.1. Descripción do k-medias 1

1.1.2. Problemas asociados ao k-medias 3

1.2. k-medoides 7

1.2.1. CLARA E CLARANS 7

2. Clustering xerárquico **9**

2.1. Métodos de enlace 11

2.2. Criterios de parada 13

2.3. Problemas asociados 15

3. Clustering baseado en modelos **17**

3.1. Mesturas finitas 18

3.2. Mesturas normais 19

3.2.1. Relación entre EM e k-medias 22

4. Clustering baseado en densidade **23**

4.1. DBSCAN 23

5. Aplicación práctica **29**

5.1. Descrición dos datos 29

5.2. Análise dos datos 32

5.3. Conclusións 36

Bibliografía **39**

Resumo

Co crecente volume de datos dispoñible, comprender a súa estrutura interna sen depender de información previa converteuse nunha tarefa crucial da análise de datos. Este traballo explora as metodoloxías de *clustering*, ou agrupamento non supervisado, que permite descubrir grupos naturais dentro dos datos sen coñecemento previo das súas categorías.

O obxectivo principal é analizar e comparar diferentes familias de algoritmos de *clustering*. Examínanse métodos particionais como o k-medias e o k-medoides, eficaces en escenarios onde os grupos teñen formas simples e ben definidas; métodos xerárquicos, que constrúen agrupamentos progresivos; enfoques baseados en modelos estatísticos, que permiten unha representación probabilística e máis flexible dos datos; e algoritmos baseados en densidade como DBSCAN, capaces de identificar formas de grupos irregulares así como observacións illadas.

Ao longo do traballo describiránse os fundamentos matemáticos, os algoritmos fundamentais, e as principais vantaxes e inconvenientes de cada método. Tamén se analizarán factores como a elección de diferentes parámetros e o tipo de grupos que é capaz de xerar cada un.

Finalmente, presentarase un conxunto de datos como exemplo sobre o que se aplican estas metodoloxías co obxectivo de visualizar e comparar os resultados de maneira práctica, facilitando así a comprensión dos distintos enfoques.

Abstract

With the growing volume of available data, understanding its internal structure without relying on prior information has become a crucial task in data analysis. This work explores clustering methodologies, or unsupervised grouping, which allow the discovery of natural groupings within the data without prior knowledge of their categories.

The main objective is to analyze and compare different families of clustering algorithms. We examine partitional methods such as k-means and k-medoids, which are effective in scenarios where the clusters have simple and well-defined shapes; hierarchical methods, which build progressive groupings; model-based approaches, which provide a probabilistic and more flexible representation of the data; and density-based algorithms such as DBSCAN, capable of identifying clusters with irregular shapes as well as isolated observations.

Throughout the work, we describe the mathematical foundations, core algorithms, and the main advantages and limitations of each method. We also analyze factors such as the choice of parameters and the types of groups each method is able to detect.

Finally, a dataset is presented as a case study in which these methodologies are applied, with the aim of visualizing and comparing the results in a practical manner, thus facilitating the understanding of the different approaches.

Introdución

A busca de patróns, a identificación de grupos e a clasificación de elementos son prácticas habituais do ser humano. Dende pequenos, aprendemos a categorizar todo o que nos rodea para facilitar a súa comprensión. Por iso, non é raro que no ámbito da análise de datos se desenvolveran diferentes técnicas estatísticas para organizar a información en grupos.

O conxunto destas técnicas coñécese co termo de *clustering*, e ten como obxectivo descubrir grupos naturais dentro dun conxunto de datos sen que exista un coñecemento previo desta clasificación. É importante recalcar isto último pois o clustering enmárcanse dentro do que se coñece como *aprendizaxe non supervisada*, na que non se dispón de etiquetas previas e que non se debe confundir coa *aprendizaxe supervisada*, na que se conta cunha variable resposta ou etiqueta xa fixadas. É o caso por exemplo da regresión.

Algúns exemplos de aplicacións do clustering [Everitt et al., 2009] son os estudos de mercado; para unha empresa pode resultar moi útil agrupar aos consumidores en función do tipo de produtos que consumen, as súas preferencias, idade, etc. Todo isto permite saber quen pode resultar un posible comprador e como achegarse a él. Estas técnicas tamén son útiles para agrupar obxectos astronómicos ou no ámbito da medicina. De feito, a primeira aparición do termo clustering aparece xa no 1954, no marco da investigación antropolóxica [Jain, 2010].

Formalmente, traballaremos cun conxunto de datos recollido nunha matriz $n \times p$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

onde cada fila $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^t \in \mathbb{R}^p$ representa un individuo ou observación e cada columna, unha variable ou característica medida. Deste xeito, podemos reescribir a nosa matriz de

datos como

$$\mathbf{X} = \begin{pmatrix} x_1^t \\ x_2^t \\ \vdots \\ x_n^t \end{pmatrix}.$$

O obxectivo xeral do clustering é asignar cada observación x_i a un subconxunto dos datos chamado *cluster* segundo algún criterio de similitude, facendo que os elementos dentro dun mesmo cluster sexan o máis similares posible entre eles e o máis diferentes posible aos elementos do resto de clusters. Estes grupos denótanse por C e o número total de clusters nos que se reparten os datos, por k. Deste xeito, cando dividamos os datos totais en k clusters, podemos usar a notación $C_1, ..., C_k$. A cada un destes clusters C_j asociámoslle o seu cardinal, que é o número de observacións que contén. Denotámolo por $|C_j|$ e é unha información moi relevante para saber o tamaño dos grupos e detectar se hai algún moi grande ou moi pequeno, feito que pode afectar á interpretación dos resultados.

Neste traballo, dedicarémonos a analizar e comparar distintos métodos de clustering, describindo o seu funcionamento, as súas vantaxes e inconvenientes, e a súa aplicabilidade a distintos tipos de datos.

Tipos de clustering. Existe unha gran variedade de métodos de clustering que foron surxindo ao longo dos anos para tratar diferentes tipos de datos ou para mellorar outros métodos xa existentes. Nos seguintes capítulos centrarémonos nalgúns deles pero primeiro imos clasificalos segundo o enfoque que usan para determinar que puntos pertencen ao mesmo cluster [Aggarwal and Reddy, 2013]. Estes enfoques diferénzanse principalmente no emprego de diferentes medidas de similitude ou disimilitude: funcións que permiten cuantificar a relación entre dous obxectos. As medidas de similitude asignan valores altos a obxectos que comparten características e, as de disimilitude fan o contrario, asignan valores máis altos canto máis diferentes son os obxectos que se están comparando.

Métodos baseados en distancia. Un exemplo moi común de medida de disimilitude para formar os clusters é a distancia. Sexa $\mathbf{X}$ un conxunto de datos como definimos anteriormente, unha distancia definida sobre el é unha función $d : \mathbf{X} \times \mathbf{X} \to \mathbb{R}^+$, sendo $\mathbb{R}^+$ o conxunto de números reais non negativos, que cumple as seguintes propiedades para todo $x_i, x_j, x_l \in \mathbf{X}$:

- Non negatividade: $d(x_i, x_j) \geq 0$.

- Separación: $d(x_i, x_j) = 0 \iff x_i = x_j$.

- Simetría: $d(x_i, x_j) = d(x_j, x_i)$.
- Desigualdade triangular: $d(x_i, x_l) \leq d(x_i, x_j) + d(x_j, x_l)$.

Unha das máis empregadas nos métodos de clustering será a distancia euclidiana

$$d_e(x_i, x_j) = \sqrt{\sum_{l=1}^{p} (x_{il} - x_{jl})^2}.$$

Para abreviar, usaremos en ocasións a notación $d_{ij} = d(x_i, x_j)$ para representar a distancia entre x_i e x_j.

A idea principal dos métodos baseados en distancia é medir canto se parecen dous obxectos mediante unha función de distancia: canto máis pequena sexa a distancia entre eles, máis parecidos se consideran e máis probabilidade hai de que formen parte do mesmo grupo. Son unha ferramenta moi empregada pois poden adaptarse a moitos tipos de datos, sempre que se escolla unha distancia compatible con eles. Por exemplo, a distancia euclidiana é axeitada para traballar con datos numéricos, mentres que outras permiten traballar con datos categóricos ou mixtos, o que lles proporciona unha gran flexibilidade. Todos os métodos baseados en distancia comparten o uso da mesma función de disimilitude pero a forma concreta na que esta se utiliza varía dun método a outro; algúns empregan a distancia para asignar cada punto ao representante dun grupo e outros emprégana como criterio para decidir que grupos fusionar ou separar.

Ao longo deste texto denotaremos por d a unha distancia xenérica definida sobre o noso conxunto de observacións sen especificar en xeral unha forma concreta. Esta función poderá ser tanto a distancia euclidiana como outras moitas, xa que o algoritmo en calquera caso será o mesmo. Iso si, o resultado pode verse afectado pola distancia empregada.

1. Clustering particional: consiste na división do conxunto de datos de xeito que cada cada observación pertenza a un único cluster [Tan et al., 2005]. É dicir, non hai solapamento nos clusters. Este enfoque coñécese como clustering duro (ou *hard clustering*). O contrario sería o clustering blando (ou *soft clustering*), no que un elemento pode pertencer a diferentes grupos, o que resulta moi útil cando os límites entre eles non son claros e pode darse solapamento.[Hastie et al., 2009].

   Os métodos particionais comezan cunha partición inicial e un representante asociado a cada cluster, para despois aplicar un proceso iterativo que busca optimizar unha función obxectivo, xeralmente minimizando a variabilidade dentro dos clusters e maximizando a separación entre eles. Deste xeito, a partición vai mellorando en cada iteración e os representantes vanse actualizando para facilitar esta mellora. Estes representantes poden ser puntos non pertencentes á mostra orixinal; no caso do k-medias por exemplo emprégase

a media dos puntos do cluster (centroide). No k-medoides pola contra, sí que se escolle un punto da mostra, concretamente aquel que minimiza a suma das distancias aos demais puntos do cluster (medoide) [Aggarwal and Reddy, 2013]. Ambos exemplos serán tratados con máis profundidade no Capítulo 1.

2. Clustering xerárquico: [Everitt et al., 2009] trátase doutra técnica de clustering duro no que os grupos se van formando progresivamente, xa sexa por medio de fusión ou división. No caso do clustering aglomerativo, considérase inicialmente cada observación como un cluster particular e vanse fusionando para dar lugar a outros máis grandes; no caso do divisivo, pártese dun único cluster que contén todos os datos e vaise separando para crear outros máis pequenos. Caracterízanse por ser representados graficamente cun dendrograma, un diagrama en forma de árbore que reflicte como se van agrupando ou separando os datos ao longo do proceso.

   Aínda que poidamos aplicar estes métodos a calquer conxunto de datos, estes non teñen por que axustarse á estrutura xerárquica necesaria para que o método sexa axeitado. No Capítulo 2, veremos como saber se os datos se axustan a esta estrutura, así como a necesidade de escoller un criterio para parar o proceso de fusión ou división unha vez se alcanza un número de clusters axeitado.

Métodos baseados en modelos. Estos métodos empregan un enfoque probabilístico polo que, antes de describilos formalmente, é necesario introducir algúns conceptos clave. Para modelar un conxunto de datos $\mathbf{X}$, considérase que cada observación x_i é a realización dun vector aleatorio $\mathbf{x}$ con valores no espazo $\mathbb{R}^p$. Formalmente, unha variable aleatoria defínese sobre un espazo de probabilidade $(\chi, \mathcal{F}, P)$ onde χ é o espazo muestral. É dicir, o conxunto de todos os resultados posibles dun experimento aleatorio. $\mathcal{F}$ é unha σ-álxebra de conxuntos cuxos elementos se chaman sucesos, e P é unha medida de probabilidade que asigna a cada suceso un valor entre 0 e 1. A variable aleatoria $\mathbf{x} : \chi \longrightarrow \mathbb{R}^p$ asigna a cada resultado de χ un vector de valores reais en $\mathbb{R}^p$.

A distribución de probabilidade dunha variable aleatoria $\mathbf{x}$ é unha función que describe a repartición teórica desta variable no espazo muestral, asignándolle probabilidades a cada resultado posible, e pode representarse de diferentes xeitos segundo a natureza dos datos cos que se estea traballando:

- Se os datos son discretos, a distribución defínese mediante unha función de masa

  $$f : \chi \longrightarrow [0, 1]$$

  onde χ é o conxunto de valores que poden tomar os datos e debe cumplirse a condición

  $$\sum_{x \in \chi} f(x) = 1$$

  .

- Se os datos son continuos, úsase unha función de densidade $f : \mathbb{R}^p \longrightarrow [0, \infty)$ que satisfai

$$\int_{\mathbb{R}^p} f(x)dx = 1$$

En moitos casos, estas funcións dependen dun conxunto de parámetros θ que determinan propiedades como a forma, a posición e a dispersión da distribución. Neste caso, a función condicionada a este parámetros denótase como $f(x|\theta)$.

Un exemplo fundamental e o cal trataremos cando falemos dos métodos baseados en modelos é a distribución normal multivariante, que vén determinada por dous parámetros: a media dos datos $\mu \in \mathbb{R}^p$ e a matriz de covarianzas Σ, de dimensión $p \times p$. A súa función de densidade defínese para $x \in \mathbb{R}^p$ como

$$f(x|\mu, \Sigma) = (2\pi)^{-\frac{p}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right)$$

onde $|\Sigma|$ denota o determinante da matriz de covarianzas.

No Capítulo 3, abordaremos os métodos baseados en mesturas finitas, nos que se supón que cada cluster está asociado a unha distribución de probabilidade específica.

Métodos baseados en densidade. Finalmente, trataremos tamén no Capítulo 4 os métodos baseados en densidade, que identifican os clusters como zonas nas que hai moita concentración de observacións (densidade alta) e que están separadas entre elas por zonas vacías ou con poucos puntos (densidade baixa).

Aínda que tanto os métodos baseados en modelos como os métodos baseados en densidade empregan unha distancia como parte do seu funcionamento, non os consideramos métodos baseados en distancia no sentido estricto pois a distancia non é o criterio central que guía a formación de grupos. É dicir, a distancia é unha ferramenta auxiliar pero non o criterio principal e único.

Capítulo 1

Clustering particional

Tras ter introducido a idea do clustering particional previamente, neste capítulo imos profundizar máis neste enfoque a través de dous métodos: o k-medias e o k-medoides. Ambos algoritmos parten da mesma idea xeral de separar os datos en grupos segundo certa medida de centralidade pero difiren na forma de definir dita centralidade e na súa sensibilidade a valores atípicos entre outras cousas.

1.1. k-medias

O algoritmo k-medias é quizais o método máis coñecido e usado de todos os que imos mencionar. A súa formulación orixinal foi proposta por Lloyd en 1957, nun informe interno onde xa se recolle un proceso iterativo que serviría de base para o procedemento actual e que, posteriormente, publicou en 1982 [Lloyd, 1982]. Máis tarde, foi en [MacQueen, 1967] onde se formalizou como método para clasificar datos multivariantes.

1.1.1. Descripción do k-medias

Antes de comezar co método, é necesario fixar previamente o número de clusters k nos que se quere dividir a mostra. A continuación, selecciónanse de xeito aleatorio k puntos do conxunto de datos total, que actuarán como representantes iniciais de cada cluster, é dicir, son os centroides $c_1, ..., c_k \in \mathbb{R}^p$ asociados a cada cluster $C_1, ..., C_k$. A cada un destes centroides asígnanselle o subconxunto de datos máis próximo a el que aos demais, o que é equivalente a un problema de optimización onde o obxectivo é minimizar a suma residual de cadrados (RSS), que se define do

seguinte xeito:

$$RSS = \sum_{j=1}^{k} \sum_{x_i \in C_j} d_e(x_i, c_j)^2$$

onde x_i é un punto de datos pertencente ao cluster C_j, c_j o representante do cluster C_j e d_e a distancia euclidiana. Tamén é habitual denominala suma de cadrados intracluster e denotala por WCSS.

Unha vez feita a asignación dos puntos a un cluster, calcúlanse as medias de cada un, que pasarán a ser os novos centroides, e repítense os procesos de asignación e actualización dos centroides ata que os puntos se asignen aos mesmos clusters que na iteración anterior[Tan et al., 2005]. Esta é a condición teórica de parada, pero na práctica poden usarse outras máis débiles como, por exemplo, parar cando os puntos que cambian de cluster entre dúas iteracións supoñen só o 1 % ou menos do total.

Algoritmo 1: Algoritmo k-medias

Entrada: Conxunto de datos $\mathbf{X}$, número de clusters k

Saída : Conxunto de clusters $C_1, C_2, \ldots, C_k$ e centroides $c_1, c_2, \ldots, c_k$

Inicialización:

Seleccionar aleatoriamente k puntos do conxunto de datos $\mathbf{X}$ para usalos como centroides iniciais $c_1, c_2, \ldots, c_k$.

repeat

  Paso 1: Asignación dos puntos aos clusters

  foreach *punto* $x_i \in X$ **do**

    Asignar x_i ao cluster C_j tal que a distancia $d_e(x_i, c_j)$ sexa mínima;

  end

  Paso 2: Actualización dos centroides

  foreach *cluster* C_j **do**

    Calcular o novo centroide c_j como a media dos puntos asignados ao cluster C_j: $c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$

  end

until *as asignacións non cambien en dúas iteracións consecutivas*;

O algoritmo k-medias baséase no uso da distancia euclidiana e no cálculo de medias aritméticas. Estas dúas operacións implican unha restrición importante sobre o tipo de datos que se poden empregar. Concretamente, fai que este algoritmo só sexa aplicable a datos cuantitativos, é dicir, a datos nos que todas as variables se representen mediante números e son cantidades medibles. A distancia euclidiana require que os valores das variables sexan comparables, o cal non ocorre se estamos traballando con datos categóricos, pois non se pode establecer unha orde entre eles sen realizar algunha transformación. Outra consecuencia do uso da distancia é que convirte ao k-medias nun método pouco robusto, é dicir, moi sensible a datos atípicos. Isto débese a que,

por construción, a distancia dálle un peso moi grande aos puntos que están lonxe do centroide, o que pode arrastralos e afectar ao resultado[Bishop, 2006].

É importante ter en conta tamén que, como o método se basea na distancia ao centroide, tende a crear grupos esféricos ou circulares arredor deles. Por tanto, non resulta útil para identificar estruturas alongadas ou irregulares nos datos. Ademais, tende a formar clusters de tamaño e dispersión similares, o que implica que non se adapta ben a situacións nas que os grupos teñen varianzas moi diferentes entre si. Nestes casos pode ocorrer que o k-means divida un cluster máis disperso en varios subgrupos ou xunte indebidamente clusters pequenos e máis compactos. Na Figura 1.1 xeráronse dous conxuntos de datos que forman un anel interior e un outro exterior, ambos concéntricos, e aplicouse o algoritmo k-medias con dous centroides iniciais. Observamos como se realiza unha separación lineal entre os grupos, sen respetar a estrutura real dos datos e pondo de manifesto o problema mencionado.

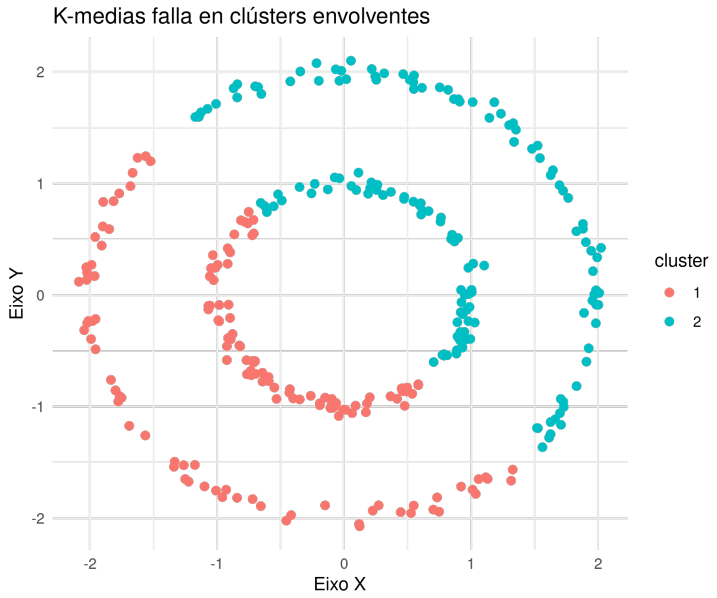


Figura 1.1: Aplicación do k-medias a datos con estrutura envolvente.

1.1.2. Problemas asociados ao k-medias

A pesar da súa popularidade, o k-medias presenta varios problemas como pode ser a sensibilidade a datos atípicos que acabamos de mencionar, pero tamén a sensibilidade aos valores iniciais e a necesidade de fixar de antemán o número de clusters.

Os valores iniciais

O algoritmo parte de k puntos escollidos de xeito aleatorio. Diferentes eleccións poden dar lugar a particións distintas dos datos, especialmente cando os clusters non están ben diferenciados ou hai solapamentos entre eles. Na Figura 1.2 móstranse os resultados de tres execucións do algoritmo k-medias con diferentes valores iniciais, onde cada cluster aparece dun color distinto. Pódese observar como a pesar de empregar sempre os mesmos datos e o mesmo valor de k, a distribución dos clusters varía.

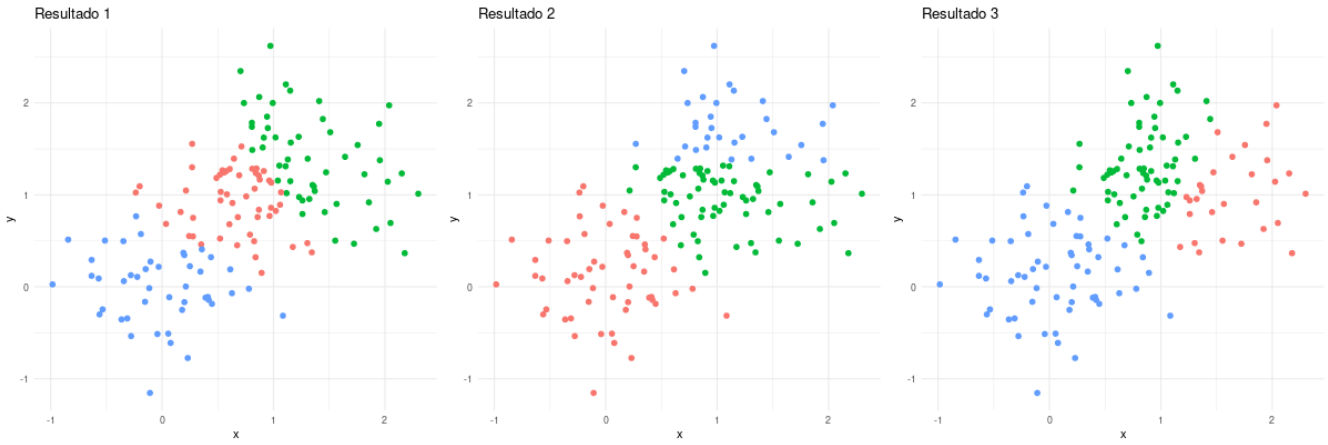


Figura 1.2: Variación dos resultados do algoritmo k-medias ao aplicar diferentes valores iniciais.

Por construcción, a función obxectivo vaise reducindo en cada iteración, polo que a converxencia está asegurada. Porén, non existe garantía de que o algoritmo converxa cara a un mínimo global, senón que pode quedarse nun mínimo local. Para solucionar este problema, é habitual iniciar o algoritmo múltiples veces con diferentes valores iniciais para, finalmente, quedarse con aquel resultado co que se obteña o RSS máis pequeno. Aínda que está moi extendido que os centroides iniciais se escollan aleatoriamente, xa que así foi como se enunciou por primeira vez, ao longo dos anos foron aparecendo diferentes métodos de inicialización co obxectivo de mellorar o algoritmo [Aggarwal and Reddy, 2013] e entre eles atópase o k-means ++ [Arthur and Vassilvitskii, 2007]. Esta variante selecciona os centroides iniciais de maneira incremental, escollendo o primeiro centroide aleatoriamente e cada un dos seguintes de xeito que se atopen o máis afastados posibles dos anteriores. Os pasos serían:

1. Escóllese o primeiro centroide ao azar.

2. Para cada $x_i \in X$, calcúlase a súa distancia ao centroide máis próximo entre os xa selec-

cionados e elévase ao cadrado. Denotando por C ao conxunto de centroides temos:

$$D(x_i)^2 = \min_{c_j \in C} \sum_{l=1}^{p} (x_{il} - c_{jl})^2.$$

3. Créase unha distribución de probabilidade onde x_j é elexido como próximo centroide con probabilidade:

$$P(x_j) = \frac{D(x_j)^2}{\sum_{x \in X} D(x)^2}.$$

Así, os puntos máis afastados dos xa seleccionados teñen unha maior probabilidade de ser escollidos.

Este proceso repítese ata obter os k centroides cos que se inicializa posteriormente o k-medias.

A elección de k

Nalgúns casos, o número de clusters k ven dado polo problema que queremos tratar. Máis outras moitas veces non, e debe estimarse. Para isto existen numerosos métodos que teñen como obxectivo medir canto diminúe a variabilidade dentro dos clusters ao aumentar k e así chegar a un k óptimo. Nós imos mencionar o método do codo, o método da silueta ([Rousseeuw, 1987]) e o Gap Statistic ([Tibshirani et al., 2001]).

O método do codo é un método visual que consiste en trazar unha gráfica onde se mostra a variación da suma de cadrados intracluster (WCSS) explicada en función do número de clusters k [Hastie et al., 2009].

$$WCSS(k) = \sum_{i=1}^{k} \sum_{x \in C_i} d_e(x - c_i)^2.$$

Por intuición, sabemos que WCSS diminuirá ao aumentar o número de clusters pois teremos clusters máis pequenos e compactos e os puntos estarán máis próximos aos seus centroides. Supondo que o número que mellor captura a estrutura dos grupos fose $K^\star$, unha vez alcanzado este punto, a diminución de WCSS producirase máis lentamente. Isto é porque, despois de chegar ao número óptimo de clusters, seguir aumentando k implicará dividir grupos que xa están ben definidos, o que non reduce tanto a disimilitude dentro de cada cluster. Por iso, este método escolle como k o punto onde se produce un cambio brusco na curva de WCSS, coñecido como o "codo" da gráfica.

Por outro lado, o método da silueta básase en calcular para cada punto da mostra o coeficiente da silueta, que nos indica como de ben ese punto se encontra situado dentro dun cluster ou se estaría mellor noutro. O coeficiente para un punto $x_i \in X$ e supondo k clusters defínese como:

$$s_k(x_i) = \frac{b(x_i) - a(x_i)}{\max(a(x_i), b(x_i))}$$

onde:

- $a(x_i)$ é a distancia promedio entre x_i e os demais puntos no seu mesmo cluster.
- $b(x_i)$ é a distancia promedio entre x_i e os puntos do cluster máis cercano ao que non pertence.

O valor de $s(i)$ varía entre 1 e -1. Se é cercano a 1, o punto está ben asignado ao cluster; se é cercano a 0, o punto está no borde de dous clusters; e se é cercano a -1, o punto debería estar noutro cluster. Para escoller k, calcúlase o promedio do coeficiente de silueta para diferentes valores de k:

$$S(k) = \frac{1}{n}\sum_{i=1}^{n} s_k(x_i)$$

e escóllese aquel k que maximice este promedio, pois é o que mellora a separación entre os clusters.

Por último, o Gap Statistic compara a variabilidade dentro dos clusters (expresada mediante a suma de cadrados dentro dos clusters $W(k)$) cunha referencia nula, é dicir, a variabilidade esperada nun conxunto de datos uniformemente distribuídos. Supomos que temos un conxunto de datos $\mathbf{X}$ agrupados en k clusters $C_1, ..., C_k$ con cardinalidade n_r e definimos a suma de cadrados dentro dos clusters como:

$$W(k) = \sum_{r=1}^{k} \frac{1}{2n_r} \sum_{x_i, x_j \in C_r} d_e(x_i, x_j).$$

Para obter o valor esperado da suma de cadrados nunha situación sen estrutura, xéranse de xeito aleatorio B conxuntos de datos cunha distribución uniforme no mesmo rango que os datos orixinais. Para cada un deles calcúlase:

$$W_b^*(k) = \sum_{r=1}^{k} \frac{1}{2n_r} \sum_{i,j \in C_r^b} d_e(x_i^b, x_j^b)$$

onde x_i^b representa os puntos do b-ésimo conxunto simulado e C_r^b os clusters resultantes da súa partición. Con isto, defínese o valor do estatístico Gap como:

$$\text{Gap}(k) = \frac{1}{B}\sum_{b=1}^{B} \log(W_b^*(k)) - \log(W(k)).$$

Se o valor de Gap é grande, quere dicir que os datos reais están moito mellor agrupados que os datos aleatorios, e iso é boa sinal. Escóllese como número óptimo de clusters aquel valor k que satisfai

$$\text{Gap}(k) \geq \text{Gap}(k+1) - s_{k+1}$$

onde s_{k+1} é unha estimación do erro estándar do logaritmo de $W_b^*(k+1)$. Isto quere dicir que o Gap para k xa é suficientemente alto e o aumento ao pasar a $k+1$ podería deberse ao azar.

1.2. k-medoides

Aínda que o algoritmo k-medias clásico está deseñado para minimizar a suma de distancias euclídeas ao cuadrado, esta elección non é obrigatoria. Tal como se sinala en [Hastie et al., 2009], "a única parte do algoritmo que asume distancia euclídea é o paso de minimización, onde os centroides se calculan como media". En contextos onde se desexa ter a oportunidade de empregar outra métrica, existen variantes do algoritmo que non requiren a media, senón que seleccionan un representante do cluster directamente entre os puntos asignados, como é o caso do k-medoides, tamén coñecido como algoritmo PAM (*Partitioning Around Medoids*) [Giordani et al., 2020b]. Isto fai que non se vexa tan afectado por valores extremos e convírteo nun método máis robusto.

O funcionamento de k-medoides é moi similar ao k-medias. Neste caso chamamos medoides aos representantes de cada cluster que, como xa dixemos, son puntos do propio conxunto de datos e denotámolos por $m_1, ..., m_k$. De novo estamos ante un proceso iterativo no que, en cada iteración, se escolle como medoide para cada cluster o punto que minimiza a distancia total aos demais puntos do cluster. Por tanto, a función obxectivo que busca minimizar este método é o criterio do erro absoluto que vén dado por:

$$S = \sum_{j=1}^{k} \sum_{x_i \in C_j} d_e(x_i, m_j)^2$$

onde m_j é o representante do cluster C_j. Aínda que se pode empregar calquera distancia d compatible cos datos, o máis habitual segue sendo usar a euclidiana.

O k-medoides mantén os problemas que xa mencionamos no k-medias relacionados coa escolla do número de clusters e os valores iniciais. As solucións propostas, como o uso do método do codo, o coeficiente de silueta e a inicialización múltiple tamén son válidas neste caso. Ademais, tamén comparten a creación de grupos esféricos.

A desvantaxe que ten este método se o comparamos co k-medias é que ten un maior custo computacional, o que pode ser prohibitivo [Aggarwal and Reddy, 2013]. Isto débese a que, en lugar de calcular os representantes como a media dos grupos, o algoritmo debe avaliar todas as posibles combinacións de puntos dentro de cada cluster para ver cal deles minimiza a distancia total. Ademais, este proceso repítese en cada iteración, o que resulta excesivamente costoso en conxuntos de datos moi grandes ou con moitas dimensións.

1.2.1. CLARA E CLARANS

A solución a este último problema ven dada a través de diferentes variantes do algoritmo PAM, como CLARA (*Clustering Large Applications*) [Kaufman and Rousseeuw, 1990] ou CLARANS

Algoritmo 2: Algoritmo k-medoides (PAM)

Entrada: Conxunto de datos $\mathbf{X}$, número de clusters k

Saída : Conxunto de clusters $C_1, C_2, \ldots, C_k$ e medoides $m_1, m_2, \ldots, m_k$

Inicialización:

Seleccionar aleatoriamente k puntos do conxunto de datos X para usalos como medoides iniciais $m_1, m_2, \ldots, m_k$.

repeat

  Paso 1: Asignación dos puntos aos clusters

  foreach *punto* $x_i \in X$ **do**

    Asignar x_i ao cluster C_j tal que a distancia $d_e(x_i, m_j)$ sexa mínima;

  end

  Paso 2: Actualización dos medoides

  foreach *cluster* C_j **do**

    Calcular o novo medoide m_j como o punto $x_i \in C_j$ que minimiza a distancia total aos demais puntos do cluster;

  end

until *as asignacións non cambien en dúas iteracións consecutivas*;

(*Clustering Large Applications based on RANdomized Search*) [Ng and Han, 1994] que empregan a mostraxe para reducir o custo computacional. Imos describir o seu funcionamento pero non entraremos máis en detalle porque se manteñen as características principais do k-medoides orixinal.

No caso de CLARA, a mostraxe ten lugar dende o principio pois extráense pequenas mostras do conxunto de datos orixinal e a cada unha aplícaselle o algoritmo PAM que vimos previamente. Dos diferentes grupos de medoides resultantes para cada mostra, elíxese aquel que minimize a función obxectivo. Este método non garante que o resultado obtido sexa o óptimo que se obtén aplicando PAM ao conxunto total dos datos, polo que é convinte aumentar o número de mostras tanto como se poida para mellorar a calidade do resultado [Hastie et al., 2009].

A diferenza de CLARA, CLARANS emprega unha motraxe dinámica en cada iteración, o que o fai máis eficiente [Aggarwal and Reddy, 2013]. En lugar de comprobar un a un os veciños dos medoides en cada iteración para ver se se encontra algún cun custo menor, elíxense de xeito aleatorio un número fixo de veciños cos que realizar esta comprobación.

Capítulo 2

Clustering xerárquico

En moitas ocasións cando temos un conxunto de datos queremos, ademais de atopar grupos naturais, coñecer a relación entre eses grupos. É dicir, entender a estrutura interna da mostra. Isto ocorre moi habitualmente en campos como a bioloxía, onde se busca clasificar especies, xenes ou proteínas para reconstruír a súa evolución e entender as relacións de descendencia. Os métodos de clustering xerárquico resultan especialmente útiles para esta labor xa que permiten ordear os clusters de xeito que uns se atopen dentro doutros. Ademais, teñen a vantaxe de que son métodos moi visuais grazas á súa representación mediante dendrogramas, dos cales daremos unha explicación detallada máis adiante.

Os métodos xerárquicos parten de cada elemento como un único cluster (aglomerativo) ou dun único elemento que contén a todos os clusters (divisivo) e van unindo ou dividindo os grupos según unha medida de disimilitude: a distancia entre clusters. Esta medida, que tamén denotaremos por d, non debe confundirse coa distancia entre observacións individuais da mostra, aínda que tamén se adoite usar a mesma notación. De feito, para calcular as distancias entre clusters empréganse os chamados métodos de enlace, que definen a disimilitude entre dous grupos a partir das distancias individuais entre os seus elementos, e dos que falaremos na Sección 2.1.

Centrarémonos no clustering xerárquico de tipo aglomerativo xa que é o máis empregado, pois o gran custo computacional que supoñen os métodos divisivos fai que os libros adoiten non profundizar neles [Kaufman and Rousseeuw, 1990]. Supondo definida con anterioridade unha medida de distancia entre clusters, o algoritmo xeral para un método aglomerativo empeza considerando cada observación como un cluster particular e, en cada iteración, une aqueles dous que minimicen dita distancia ata chegar a un cluster único.

Algoritmo 3: Algoritmo de Clustering Xerárquico Aglomerativo

Input: Conxunto de datos X

Output: Cluster final C

Inicializar cada punto como un cluster individual;

while *o número de clusters é maior que 1* **do**

  foreach *par de clusters* (C_i, C_j) **do**

    Calcular a distancia entre C_i e C_j;

  end

  Identificar os dous clusters C_i e C_j máis próximos;

  Fusionar C_i e C_j nun novo cluster C_{ij};

  Actualizar as distancias ou proximidades entre os clusters restantes;

end

Os dendrogramas son clave para entender o funcionamento destes métodos xa que reflicten visualmente as unións que van tendo lugar ao longo do proceso. Na Figura 2.1 atopamos un exemplo no que se van fusionando distintas cidades europeas en función da distancia xeográfica á que se atopan. Os nós do dendrograma son os puntos de unión das ramas e representan os clusters do algoritmo. Os nós terminais, situados na parte baixa do dendrograma, correspóndense coas observacións individuais e aparecen colocados de xeito que faciliten a representación polo que a orde na que aparecen non ten un significado estatístico. Por outra banda, os nós internos correspóndense cos clusters novos, onde as ramas que conflúen nese nó, proveñen dos clusters que se fusionan para a súa formación. O eixo de ordenadas mostra a altura á que se atopan eses nós internos, o cal se corresponde coa distancia á que se atopan os clusters fusionados.

A pesar da súa utilidade, cómpre ter precaución á hora de traballar cun dendrograma pois diferentes métodos xerárquicos poden producir dendrogramas distintos, pequenos cambios nos datos poden alteralos enormemente e incluso existe a posibilidade de que a estrutura xerárquica non reflexe realmente os datos [Hastie et al., 2009]. Para solucionar isto último podemos recurrir ao *coeficiente de correlación cofenética* (CCC) [Sokal and Rohlf, 1962] que mide como de ben un dendrograma representa os nosos datos. Para poder entendelo, primeiro temos que definir a *distancia cofenética* entre dous puntos x_i e x_j da mostra (c_{ij}), que é a distancia entre os clusters aos que pertencen x_i e x_j no momento xusto no que se unen por primeira vez no mesmo cluster, é dicir, coincide coa altura á que se atopa o nodo no dendrograma. O CCC é a correlación de Pearson que existe entre as distancias orixinais d_{ij} e as distancias cofenéticas. Ven dado por:

$$\mathrm{CCC} = \frac{\sum_{i<j}(d_{ij} - \overline{d})(c_{ij} - \overline{c})}{\sqrt{\sum_{i<j}(d_{ij} - \overline{d})^2 \sum_{i<j}(c_{ij} - \overline{c})^2}}$$

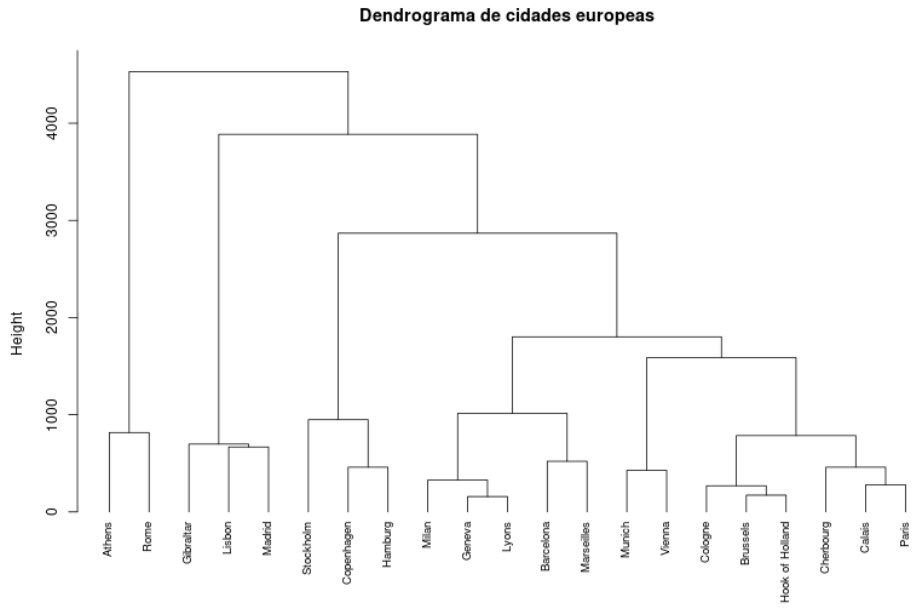


Figura 2.1: Dendrograma xerado mediante clustering xerárquico aglomerativo sobre as distancias entre cidades europeas.

sendo $\overline{d}$ e $\overline{c}$ as medias das distancias d_{ij} e c_{ij} respectivamente. Se toma un valor próximo a 1, o dendrograma é unha boa representación dos datos. Se é próximo a 0, non existe relación lineal entre as distancias orixinais e as cofenéticas polo que non é axeitado o uso do dendrograma. Se é próximo a -1, indica que existe relación lineal pero inversa, o cal non tería sentido pois suporía unha relación oposta entre dúas medidas de disimilitude.

A principal vantaxe respecto aos métodos particionais é que non é necesario fixar o número de clusters k de antemán. Pola contra, é habitual non continuar o proceso de fusión ata o final e cortar o dendrograma nalgún punto que resulte óptimo. Para iso, é necesario determinar un criterio que nos indique cando parar. Facelo en diferentes momentos pode xerar resultados moi distintos xa que o número de clusters non será o mesmo. Veremos algúns destes criterios na Sección 2.2.

2.1. Métodos de enlace

Nesta sección centrarémonos na mención de varias distancias entre clusters, que dan lugar a diferentes métodos. Todas elas empregan na súa definición a distancia entre individuos, podendo esta calcularse empregando calquera medida de distancia axeitada para os datos.

Se os datos cos que estamos traballando son compactos e ben definidos, os métodos que introduciremos a continuación darán lugar a resultados moi similares [Hastie et al., 2009]. Porén, se isto non ocorre, os resultados poden variar considerablemente debido ao diferente xeito de calcular as distancias, o que pode alterar o dendrograma.

Single e Complete linkage

O método *Single-Linkage Clustering* (SLC) caracterízase por unir en cada etapa os dous clusters con menor distancia mínima entre eles. É decir, une os clusters C_i, C_j que minimicen

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y).$$

Un fenómeno que ten lugar moi habitualmente con este método é o *encadenamento*, no cal dous puntos aparentemente moi afastados poden acabar agrupados no mesmo cluster porque existe unha cadea de observacións intermedias próximas entre si [Hastie et al., 2009]. Isto dá lugar a clusters alongados. Ademais, unha das súas principais desvantaxes é que se ve moi afectado por valores atípicos.

Por outro lado, temos o *Complete-Linkage Clustering* (CLC), que en cada etapa une os dous clusters con menor distancia máxima. É dicir, une os clusters C_i, C_j que minimicen

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y).$$

Neste caso obtéñense clusters máis compactos e é lixeiramente menos sensible a outliers que o *Single-Linkage* aínda que tamén se pode ver afectado.

Como o *Single Linkage* non ten en conta a estrutura total dos datos, senón que se centra só na menor distancia entre puntos de distintos grupos, dise que se trata dun método local. Pola contra, o Complete Linkage considérase un método global pois o uso da distancia máxima permítelle captar mellor a estruturas dos clusters [Aggarwal and Reddy, 2013].

Group Average

No caso do *Group Average* (UPGMA), defínese a distancia entre clusters como o promedio das distancias por pares de todos os puntos procedentes dos clusters. É dicir, a distancia entre dous clusters C_i e C_j defínese como:

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y).$$

Trátase dun método robusto máis o cálculo resulta moi costoso debido á gran cantidade de operacións a realizar especialmente cando temos un conxunto de datos moi grande. Adoita crear

clusters con varianzas pequenas, é dicir, con elementos moi próximos entre sí [Everitt et al., 2009].

Criterio de Ward

Por último, o método ou *Criterio de Ward*, usa o mesmo criterio que o k-medias; dous clusters únense se producen o menor incremento da varianza total, tendo en conta todas as posibles combinacións de grupos de dous clusters. É dicir, se minimizan:

$$d(C_i, C_j) = \frac{|C_i||C_j|}{|C_i| + |C_j|} d_e(c_i, c_j)^2$$

onde c_i e c_j representan os centroides de ditos clusters. Ao empregar a distancia euclidiana e o cálculo de centroides, este método só pode empregarse con datos cuantitativos [Giordani et al., 2020a]. O principal problema ligado a este criterio é que o factor $\frac{|C_i||C_j|}{|C_i|+|C_j|}$ tende a aumentar co tamaño dos clusters. Isto provoca que, a medida que se avanza no proceso de agrupamento e os grupos son cada vez máis grandes, a distancia entre eles (e por tanto a altura das fusións no dendrograma) aumente rapidamente. Este fenómeno pode dar lugar a factores de disimilitude moi elevados nas últimas fusións, o que produce un aspecto esaxerado no tramo final do dendrograma. Tal comportamento non sempre reflicte unha separación real tan pronunciada entre grupos, senón que é consecuencia da función de custo do modelo, polo que debe ser interpretada con cautela.

Para ilustrar que cada un destes métodos pode dar lugar a un resultado distinto, na Figura 2.2 atopamos o dendrograma asociado á agrupación de cidades europeas empregando esta vez os catro métodos mencionados. Podemos apreciar como, por exemplo, o *Criterio de Ward* une Roma e Atenas moi pronto, mentres que o *Single Linkage* non o fai ata o final pois, aínda que están relativamente cerca, hai outros pares que o algoritmo fusiona primeiro.

2.2. Criterios de parada

Encontrar o momento axeitado no que cortar o dendrograma non é doado e existen numerosos métodos elaborados para esta tarefa. Non hai un consenso claro sobre cal é mellor usar así que é habitual que a decisión a tome o investigador según lle pareza máis razoable. Existe a posibilidade de que o número de clusters xa esté fixado no problema que queremos resolver. Por exemplo, se temos unha mostra dunha poboación de peixes e sabemos de antemán que proceden de catro liñas xenéticas distintas podemos cortar o proceso cando teñamos catro grupos. Se non é o caso e non temos tal coñecemento, unha opción visual é buscar o salto de altura máis grande entre dúas fusións consecutivas xa que un salto brusco pode indicar que se están fusionando grupos non tan similares. Agora ben, non sempre ten que existir un cambio tan evidente e hai factores

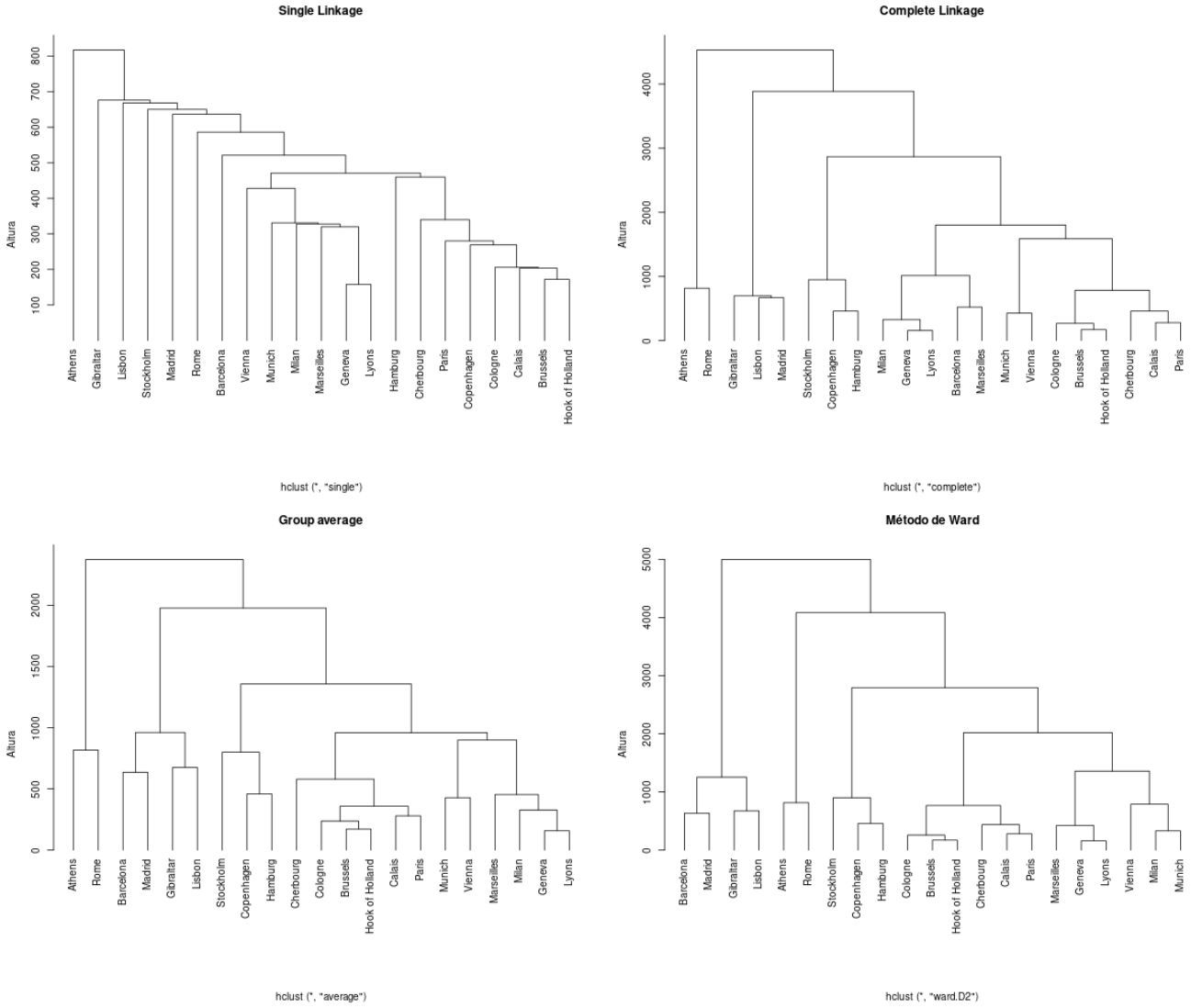


Figura 2.2: Dendrogramas xerados empregando diferentes métodos de enlace.

que poden interferir nestas alturas como a escala das distancias ou a presenza de valores atípicos [Giordani et al., 2020a]. Tamén existen outros métodos que permiten cortar distintas ramas do dendrograma a diferentes alturas e outros que usan técnicas estatísticas máis formais, como o cálculo das medias e desviación típica dos niveis de fusión [Everitt et al., 2009].

2.3. Problemas asociados

Un dos principais problemas dos métodos xerárquicos é que as unións ou divisións son irrevocables polo que un erro cometido nunha etapa non pode ser arranxado nas posteriores [Everitt et al., 2009]. Outro factor a ter en conta se os comparamos cos métodos particionais é que son máis costosos computacionalmente pois requiren un maior almacenamento de memoria e máis tempo de execución [Tan et al., 2005].

Nalgúns dos métodos mencionados anteriormente, como é o caso do *Single Linkage*, é frecuente o empate nas distancias de varios clusters. A maioría de programas (R, Python, etc.) teñen fixada unha orde predeterminada para escoller que clusters se fusionan en caso de empate. Moitas veces, este criterio ten que ver coa orde dos obxectos na matriz da mostra polo que se recomenda executar os algoritmos varias veces cambiando o orde dos datos [Everitt et al., 2009].

Capítulo 3

Clustering baseado en modelos

Os métodos descritos ata o momento son fundamentalmente heurísticos. Aínda que resultan moi útiles na práctica, a súa aplicación depende de decisións previas relevantes como son a determinación do número de clusters ou a selección dunha medida de distancia. Pola contra, os métodos baseados en modelos que presentamos neste capítulo propoñen un enfoque máis probabilístico para agrupar os datos. Neste traballo centrarémonos nos modelos de mesturas finitas. Estes modelos asumen que o conxunto dos datos se xera a partir dunha combinación de distintas distribucións probabilísticas, onde cada unha delas representa un subconxunto de datos cunha estrutura específica que dá lugar a un cluster diferenciado.

Deste xeito, o uso de modelos de mesturas finitas permite relaxar as restricións asociadas aos métodos anteriores como a igualdade do número de observacións por grupo, a homoxeneidade das varianzas ou a forma dos clusters. Ao permitir que os clusters teñan diferentes formas e tamaños, estes modelos ofrecen unha maior flexibilidade e permiten capturar mellor a estrutura dos datos. En xeral, podemos considerar que cada cluster posúe unha distribución totalmente diferente, máis tamén se adoita considerar que as distribucións pertencen á mesma familia paramétrica diferenciándose unicamente nos valores dos seus parámetros. Un exemplo moi empregado é o modelo de mesturas normais, onde cada cluster posúe unha distribución normal con diferentes medias e varianzas. Este será o caso particular que abordaremos na Sección 3.2.

Á diferenza dos métodos heurísticos onde cada punto queda asignado de forma ríxida a un só cluster, nos métodos de mesturas finitas asígnaselle a cada dato un conxunto de probabilidades de pertenza aos distintos clusters. É dicir, un mesmo punto pode ter, por exemplo, un 60 % de probabilidade de pertencer a un cluster e un 40 % de pertencer a outro. Esta característica resulta espacialmente vantaxosa pois permite o solapamento dos clusters e axuda a modelar situacións nas que os límites entre os clusters non están claros.

Para ilustrar esta diferenza, na Figura 3.1 móstrase un exemplo no que se aplicou tanto

o método k-medias como un método de mesturas finitas. Os clusters resultantes do primeiro caso aparecen representados con puntos de diferentes cores, onde se aprecia a realización dunha partición ríxida. Por outro lado, as elipses representan rexións de probabilidade alta asociadas aos clusters resultantes co método de mesturas finitas e, como vemos, permítennos visualizar o solapamento entre clusters.

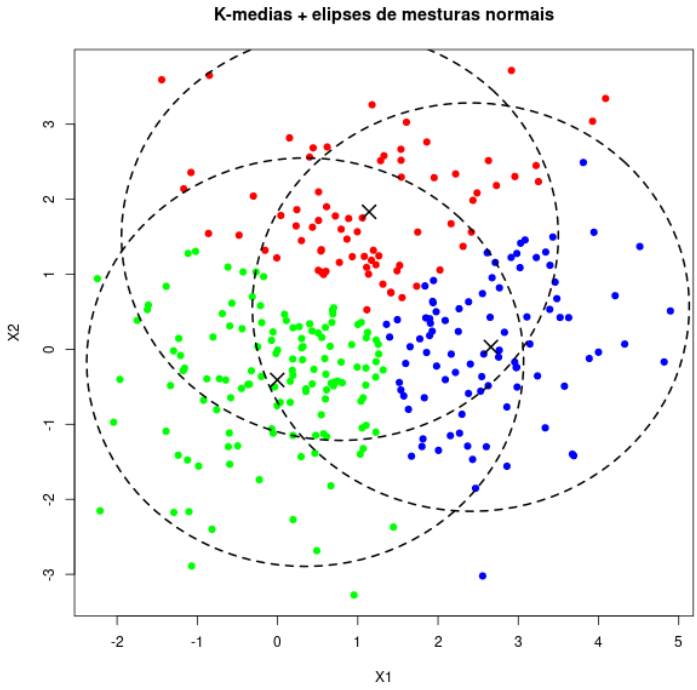


Figura 3.1: Comparación entre k-medias e mesturas finitas.

3.1. Mesturas finitas

Os modelos de mesturas finitas constitúen unha familia de modelos estatísticos onde a distribución dunha poboación ven dada como combinación lineal doutras distribucións máis simples [Everitt et al., 2009]. Dada una mostra $\mathbf{X}$, onde recordamos que cada unha das n observacións x_i está definida en $\mathbb{R}^p$, e supondo que os datos proveñen da mestura de k distribucións diferentes, podemos escribir a función de probabilidade conxunta como

$$f(x_i|\pi,\theta) = \sum_{j=1}^{k} \pi_j f_j(x_i|\theta_j)$$

onde f_j é a función de densidade ou de masa (dependendo de se os datos son continuos ou discretos) asociada ao cluster j-ésimo, aplicada ao individuo i e parametrizada por θ_j. O conxunto

de todos os parámetros das k compoñentes represéntase como $\theta = (\theta_1, ..., \theta_k)^t$. Os coeficientes π_j coñécense como proporcións de mestura e representan os pesos asociados a cada compoñente, indicando a fracción da poboación total que se espera que pertenza ao cluster j. Por tanto, cada π_j debe ser non negativo e a suma total verifica

$$\sum_{j=1}^{k} \pi_j = 1.$$

O vector completo destas proporcións denótase $\pi = (\pi_1, ..., \pi_k)^t$.

Unha vez definido o modelo, o obxectivo principal é estimar os parámetros, o que inclúe tanto ás proporcións de mestura π como o conxunto de parámetros específicos de cada compoñente θ. Denotaremos por $\Theta = (\pi_1, ..., \pi_k, \theta_1, ..., \theta_k)$ ao conxunto completo destes parámetros. Para realizar a estimación, trabállase coa chamada función de verosimilitude, que expresa a probabilidade de observar a mostra $\mathbf{X}$ en función dos parámetros do modelo:

$$l(\Theta|\mathbf{X}) = \prod_{i=1}^{n} \sum_{j=1}^{k} \pi_j f_j(x_i|\theta_i).$$

Debido á dificultade de traballar con produtos, é habitual aplicar o logaritmo para obter a función de log-verosimilitude

$$L(\Theta|\mathbf{X}) = \sum_{i=1}^{n} \log \sum_{j=1}^{k} \pi_j f_j(x_i|\theta_i).$$

O procedemento estándar para a estimación baséase en atopar o estimador de máxima verosimilitude (EMV), que é aquel Θ que maximiza a función de log-verosimilitude

$$\Theta_{EM} = \arg \max_{\Theta} L(\Theta|\mathbf{X}),$$

xa que deste xeito estamos seleccionando os valores dos parámetros que fan que a mostra observada sexa o máis probable posible.

A pesar de que a estimación por máxima verosimilitude é un enfoque teóricamente sólido, a súa aplicación práctica no caso de mesturas finitas pode presentar dificultades xa que a función de log-verosimilitude inclúe unha suma dentro dun logaritmo, o que impide obter derivadas explícitas. Para abordar esta situación, emprégase o algoritmo *Expectation-Maximization* (EM), que emprega un proceso iterativo para o seu cálculo. Na seguinte sección, explicaremos o funcionamento deste algoritmo para o caso particular das mesturas de distribucións normais.

3.2. Mesturas normais

O *Gaussian Mixture Model* (GMM) é un exemplo de mesturas finitas no que se asume que os datos observados foron xerados a partir dunha combinación de varias distribucións normais

multivariantes. É dicir, cada cluster C_j segue a súa propia distribución normal coa súa media $\mu_j \in \mathbb{R}^p$ e a súa matriz de covarianzas particulares $\Sigma_j \in \mathbb{R}^{p\times p}$

$$f(x_i|\pi,\theta) = \sum_{j=1}^{k} \pi_j f_j(x_i|\mu_j,\Sigma_j).$$

Debido a que as distribucións normais están definidas unicamente para variables aleatorias continuas, este modelo require que os datos sexan continuos.

Para a estimación dos parámetros (π_j,μ_j,Σ_j) usando o enfoque de máxima verosimilitude, está estendido o uso do algoritmo *Expectation-Maximization* (EM) [Dempster et al., 1977] tal e como mencionamos previamente. O algoritmo EM é un proceso iterativo que se divide en dúas etapas ben diferenciadas: *Expectation* (E) e *Maximitation* (M). Aínda que este proceso se pode definir de xeito xeral para aplicalo noutros modelos de mesturas finitas, imos explicar o seu funcionamento centrándonos no exemplo concreto de mesturas finitas normais.

Para poder levar a cabo o algoritmo é preciso definir antes un conxunto de variables vectoriais latentes. É dicir, un conxunto de variables que non se observan directamente no modelo pero que son importantes para entender os datos. Neste caso, as variables latentes $(z_1,...,z_n)$ indican a que compoñente pertence cada mostra. Cada $z_i = (z_{i1},...,z_{ip})$ é un vector de lonxitude p onde z_{ij} é unha variable binaria con $z_{ij} = 1$ se x_i pertence á compoñente j e $z_{ij} = 0$ en caso contrario [Peña, 2010]. Podemos reescribir a función de densidade de x_i empregando estas variables como

$$f(x_i|z_i) = \prod_{j=1}^{k} f_j(x_i)^{z_{ij}}$$

e a función de probabilidade de z_i, que coincide coa probabilidade de que a observación x_i pertenza á compoñente j que se lle asignou, ven dada por

$$p(z_i) = \prod_{j=1}^{k} \pi_j^{z_{ij}}.$$

Por tanto, a función de densidade conxunta queda do seguinte xeito

$$f(x_i,z_i) = f(x_i|z_i)p(z_i) = \prod_{j=1}^{k} (\pi_j f_j(x_i))^{z_{ij}}$$

e reescribimos tamén a función de log-verosimilitude

$$L(\Theta|X,Z) = \sum_{i=1}^{n} \log f(x_i,z_i) = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}\log\pi_j + \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}\log f_j(x_i).$$

Procedemos a substituir agora o valor da función de densidade para o noso caso concreto de mesturas normais

$$L(\Theta|\mathbf{X},Z) = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}\log\pi_j + \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}\log\left((2\pi)^{-\frac{p}{2}}|\Sigma_j|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(x_i-\mu_j)^t\Sigma_j^{-1}(x_i-\mu_j)\right)\right).$$

No caso ideal de que se coñecesen as variables z_i, poderíamos calcular de maneira directa os estimadores de máxima verosimilitude para os parámetros do modelo mediante esta funcíón de log-verosimilitude. Se derivamos esta función con respecto das medias e igualamos a cero

$$\frac{\partial L(\Theta|\mathbf{X}, Z)}{\partial \mu_j} = \sum_{i=1}^{n} z_{ij}\Sigma^{-1}(x_i - \mu_j) = 0.$$

podemos obter o estimador de máxima verosimilitude das medias despexando μ_j,

$$\hat{\mu_j} = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}x_i$$

De xeito análogo obtense o estimador para a matriz de covarianzas

$$\hat{\Sigma_j} = \frac{\sum_{i=1}^{n} z_{ij}(x_i - \hat{\mu_j})(x_i - \hat{\mu_j})^t}{\sum_{i=1}^{n} z_{ij}}$$

e para as proporcións de mestura

$$\hat{\pi_j} = \frac{1}{n}\sum_{i=1}^{n} z_{ij}.$$

Porén, hai que ter en conta que na práctica as variables latentes non son observables, polo que non se poden aplicar directamente as fórmulas anteriores. Por iso, o EM substitúe os z_{ij} polas súas esperanzas condicionais dado o conxunto de datos observados. Estas cantidades interprétanse como a probabilidade a posteriori de que a observación x_i pertenza ao cluster j tras observar os datos. Para poder calcular, necésitanse os parámetros do modelo polo que é necesario partir duns valores iniciais para eles $\hat{\Theta^{(0)}}$. Deste xeito, as probabilidades condicionadas son

$$\mathbb{E}(z_{ij}|\mathbf{X}, \hat{\Theta}^{(0)}) = p(x_{ij} = 1|\mathbf{X}, \hat{\Theta}^{(0)}) = \hat{\pi_{ij}}^{(0)}$$

onde os os $\hat{\pi_{ij}}^{(0)}$ se poden calcular a través do teorema de Bayes como

$$\hat{\pi_{ij}}^{(0)} = \frac{\pi_j^{(0)} f_j(x_i|\hat{\Theta}^{(0)})}{\sum_{j=1}^{k} \pi_j^{(0)} f_j(x_i|\hat{\Theta}^{(0)})}.$$

Este é o cálculo no que se fundamenta o paso E do algoritmo. Unha vez temos estas probabilidades calculadas, podemos proceder co paso M, no que se actualizan os parámetros do modelo de xeito que maximicen a función de log-verosimilitude empregando os valores estimados no paso E. Facendo uns cálculos similares aos que xa fixemos antes, derivar e igualar a cero para cada un dos parámetros, obtemos os estimadores

$$\hat{\pi_j}^{(1)} = \frac{1}{n}\sum_{i=1}^{n} \hat{\pi_{ij}}^{(0)}$$

$$\hat{\mu_j}^{(1)} = \sum_{i=1}^{n} \frac{\hat{\pi_{ij}}^{(0)}}{\sum_{i=1}^{n} \hat{\pi_{ij}}^{(0)}} x_i$$

$$\hat{\Sigma_j}^{(1)} = \sum_{i=1}^{n} \frac{\hat{\pi_{ij}}^{(0)}}{\sum_{i=1}^{n} \hat{\pi_{ij}}^{(0)}} (x_i - \mu_j)(x_i - \mu_j)^t.$$

Unha vez actualizados os parámetros, estes novos valores empréganse para repetir o paso E. Esta alternancia entre os pasos E e M continúa ata que se cumpre un criterio de converxencia predefinido, como pode ser que a diferenza entre o valor da función de log-verosimilitude nunha iteración e na anterior sexa menor que un certo número fixado. Neste caso, considérase que o algoritmo converxeu e detense o proceso tomando como parámetros os últimos calculados. En caso contrario, vólvese empezar o ciclo no paso E empregando os novos valores.

Cómpre ter en conta que o algoritmo EM pode converxer a un óptimo local da función de log-verosimilitude, polo que os resultados dependen en gran medida da elección inicial dos parámetros. Isto é algo que xa ocorría noutros métodos mencionados anteriormente e a solución é análoga: executar o algoritmo varias veces empregando distintas condicións iniciais e seleccionar aquela que produza unha maior verosimilitude final.

Ademais, o número de clusters k tamén debe especificarse previamente, ao igual que no caso de clustering particional, polo que se pode recorrer a métodos como os xa explicados na Sección 1.1.2.

Existen numerosas extensións e variantes do algoritmo EM que foron propostas para abordar estas desvantaxes mencionadas ou para adaptalo a casos máis complexos [Aggarwal and Reddy, 2013]. Con todo, estas variantes non serán abordadas neste traballo.

3.2.1. Relación entre EM e k-medias

O k-medias pode verse como un caso límite do algoritmo EM para mesturas normais [Bishop, 2006]. Supoñamos que cada compoñente segue unha normal multivariante na que asumimos que todos os clusters teñen a mesma matriz de covarianzas $\Sigma = \epsilon I$, onde I é a matriz identidade e ϵ, un escalar positivo. Esta forma particular representa clusters esféricos e de igual tamaño.

Se aplicamos o algoritmo EM veremos que, se facemos tender ϵ a 0, o cálculo das probabilidades condicionadas no paso E dá lugar a valores binarios que asignan o valor 1 a $\hat{\pi_{ij}}$ se x_i está máis próximo á μ_j que ás demais medias, e 0 en caso contrario. Isto implica que, neste caso, EM realiza unha asignación dura, ao igual que fai k-medias. Ademais, as actualizacións das medias no paso M correspóndense coas medias das observacións asignadas a dito cluster e a función de log-verosimilitude convírtese na función obxectivo de k-medias, a suma das distancias ao cadrado entre cada punto e o centro (media) do seu cluster, pero con signo negativo.

Capítulo 4

Clustering baseado en densidade

O tipo de clustering que imos tratar a continuación baséase na identificación de rexións con alta densidade de datos, as cales son interpretadas como clusters, rodeadas de zonas menos densas. Estes métodos pertencen á categoría de modelos non paramétricos xa que, a diferenza doutros modelos como as mesturas finitas, non supoñen que os datos seguen unha distribución de probabilidade determinada nin tampouco requiren especificar previamente o número de clusters. Esta característica proporciónalles unha maior capacidade para detectar valores atípicos e para tratar clusters con formas arbitrarias e irregulares.

Un exemplo onde estos métodos pode resultar moi útil é o caso dos datos espaciais; en áreas como a xeografía, a ecoloxía ou o urbanismo é habitual que os clusters adopten formas alongadas, curvadas e difíciles de capturar mediante métodos máis ríxidos. Esto débese a que a distribución dos puntos pode estar influída por condicións físicas como montañas, ríos ou rúas. Os métodos baseados en densidade, ao non depender de supostos paramétricos, teñen unha maior capacidade para detectar este tipo de estruturas complexas [Aggarwal and Reddy, 2013] .

4.1. DBSCAN

Un dos algoritmos máis empregados para a búsqueda de rexións de densidade alta é DBSCAN (*Density Based Spatial Clustering of Applications with Noise*) [Ester et al., 1996]. Este método, que emprega un enfoque baseado en veciñanzas, require definir dous parámetros fundamentais que se deberán fixar de antemán antes de comezar co algoritmo: Eps, que determina o radio de veciñanza dun punto e $MinPts$, que establece o número mínimo de veciños necesarios para considerar que un punto pertence a unha rexión densa.

Dado un conxunto de datos $\mathbf{X}$ e fixado uns valores para Eps e $MinPts$, para cada punto x_i,

cóntase o número de observacións que se encontran a unha distancia igual ou menor que Eps, o que é equivalente ao cardinal do conxunto

$$N_{Eps}(x_i) = \{x_j \in \mathbf{X} | d(x_i, x_j) \leq Eps\}$$

onde d é unha distancia definida en $\mathbb{R}^p$ (habitualmente a distancia euclidiana). En función do valor deste cardinal, podemos clasificar cada punto x_i da seguinte maneira:

- CENTRAL (core) se $|N_{Eps}(x_i)| > MinPts$. É decir, se a veciñanza de x_i contén un número suficiente de puntos.
- FRONTEIRA (*border*) se non é punto central pero ten polo menos un punto central a unha distancia menor ou igual que Eps.
- RUÍDO (*noise*) se non é nin central nin fronteira.

O funcionamento do algoritmo consiste en percorrer todos os puntos do conxunto de datos $\mathbf{X}$ e, para cada $x_i \in \mathbf{X}$, avaliar se se trata dun punto central. Se un punto cumpre a condición de ser central, considérase unha semente para iniciar un novo cluster no que, inicialmente, estarán todos aqueles puntos contidos na súa veciñanza $N_{Eps}(x_i)$. A continuación, o algoritmo explora iterativamente cada un destes veciños. Se algún deles é tamén punto central, a súa propia veciñanza engádese ao cluster. Isto segue de xeito iterativo ata que xa non queden máis puntos por recorrer. Se dentro de $N_{Eps}(x_i)$ temos algún x_j que non é central, considérarémolo como fronteira. É dicir, segue pertencendo ao cluster pero non aporta a súa propia veciñanza. Por último, se inicialmente x_i xa non é central, considerarémolo ruído e non formará parte de ningún dos nosos grupos. Este proceso aparece recollido no Algoritmo 4.

En termos xerais, o algoritmo DBSCAN é independente da orde na que se consideren os puntos agás no caso particular de solapamentos dos clusters. Se dous clusters están moi preto, pode ocorrer que compartan un punto fronteira. Este tipo de solapamento non altera a estrutura global dos clusters xa que cada punto fronteira só se asigna a un cluster, concretamente ao primeiro que o detecte durante o algoritmo como parte da súa veciñanza.

Non obstante, se dous clusters comparten un punto central, o algoritmo considera que están densamente conectados e, por tanto, forman un único cluster. Isto reflicte que DBSCAN non contempla o solapamento entre clusters, senón que realiza unha separación ríxida dos datos [Ester et al., 1996].

Unha das principais vantaxes deste algoritmo radica no tratamento dos valores atípicos, xa que non obriga a asignar todos os puntos dos datos a un cluster. En lugar diso, identifica explícitamente que puntos non se integran en ningunha rexión densa e trátaos como ruído, o que

Algoritmo 4: Algoritmo DBSCAN

Input: Conxunto de datos $\mathbf{X}$, Eps e $MinPts$
Output: Conxunto de clusters $\{C_1, C_2, \ldots, C_k\}$
Marcar todos os puntos como non visitados;
Inicializar $c = 0$;
foreach *punto* $x_i \in \boldsymbol{X}$ **do**
  if x_i *non foi visitado* **then**
    Marcar x_i como visitado;
    Obter $N_{Eps}(x_i)$;
    if $|N_{Eps}(x_i)| \geq MinPts$ **then**
      Crear un novo cluster C_{c+1} e engadir x_i;
      Engadir todos os puntos de $N_{Eps}(x_i)$ á lista de expansión;
      while *a lista de expansión non estea baleira* **do**
        Extraer y da lista;
        if y *non foi visitado* **then**
          Marcar y como visitado;
          Obter $N_{Eps}(y)$;
          if $|N_{Eps}(y)| \geq MinPts$ **then**
            Engadir todos os puntos de $N_{Eps}(y)$ á lista de expansión;
          end
        end
        Engadir y ao cluster C_{c+1};
      end
      Actualizar $c = c + 1$;
    end
    else
      Marcar x_i como ruído;
    end
  end
end

lle dá unha vantaxe significativa fronte a algoritmos que se ven obrigados a ter en conta eses datos podendo afectar á distribución final dos clusters.

Por outro lado, tamén é preciso mencionar algunha carencia do DBSCAN. Aínda que non é necesario especificar previamente o número de clusters, si é preciso fixar dous valores esenciais: Eps e $MinPts$. Como ocorre en calquera método no que se deban propoñer uns valores iniciais, unha elección inadecuada pode levar a resultados pouco satisfactorios. Para axudar na determinación axeitada de Eps pode empregarse o chamado gráfico da k-distancia. Esta técnica consiste en calcular para cada punto do conxunto de datos, á distancia ao seu k-ésimo veciño máis próximo, onde $k = MinPts$. Por definición, os puntos pertencentes ás mesmas rexións densas deberían presentar distancias pequenas, mentres que os puntos illados ou pertencentes ao ruído tenderán a mostrar distancias maiores. Ao representar estas distancias ordeadas de maior a menor, é habitual observar un cambio abrupto na curva. O punto de inflexión que marca este cambio serve como estimador de Eps [Ester et al., 1996]. Na Figura 4.1 móstrase un exemplo de gráfico de k-distancia aplicado a un conxunto de datos con clusters de distinta densidade, neste caso con $k = 4$. Pódese apreciar claramente unha rexión onde a curva se eleva bruscamente e o punto de inflexión xa aparece marcado como Eps.

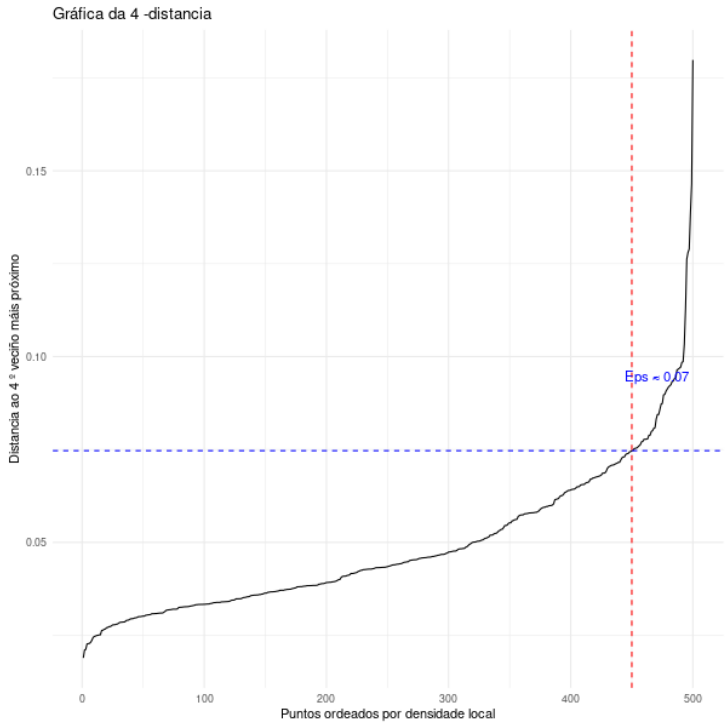


Figura 4.1: Exemplo dun gráfico de k-distancia.

Por outra parte, a elección dun único Eps que require o algoritmo pode resultar inadecuada cando os clusters teñen densidades moi diferentes [Aggarwal and Reddy, 2013]. Se o valor escolli-

do para *Eps* é demasiado pequeno, os puntos pertencentes a rexións menos densas poderían non cumprir o criterio de densidade mínima e ser clasificados erróneamente como ruído. Pola contra, se *Eps* é excesivamente grande, existe o risco de que puntos illados sexan agrupados dentro dun cluster.

Capítulo 5

Aplicación práctica

Nesta sección, abordaremos a aplicación práctica dos métodos expostos previamente. O obxectivo é demostrar a utilidade destas técnicas e as diferenzas entre elas mediante a súa aplicación a un conxunto de datos concreto. Para iso, partiremos dunha descrición xeral dos datos cos que imos traballar: características xerais e cal é o seu interese de estudo. A partir de aí, aplicaremos os algoritmos de clustering tratados ao longo do traballo e analizaremos o seu resultado axudándonos tamén de ferramentas gráficas. Todo este proceso levarase a cabo empregando a linguaxe de programación [R Core Team, 2024], que ofrece unha ampla variedade de ferramentas estatísticas e paquetes específicos para aplicar os métodos de *clustering*.

5.1. Descrición dos datos

Para levar a cabo esta práctica, empregarase o conxunto de datos *olive* do paquete *pgmm* de R. Este conxunto recolle información sobre a composición química de mostras de aceite de oliva procedentes de diferentes rexións xeográficas de Italia. En total, inclúe 572 observacións, cada unha correspondente a unha mostra de aceite.

A base de datos aporta información categórica como a rexión e área de procedencia e oito variables cuantitativas que recollen a proporción de ácidos graxos presentes en cada mostra: ácido palmítico, palmitoleico, esteárico, oleico, linoleico, linolénico, araquídico e eicosenoico. Estas variables cuantitativas son continuas e están expresadas en unidades por mil (‰).

Realizando un *summary* dos datos, o cal aparece recollido na Figura 5.1, observamos que o ácido oleico destaca como compoñente maioritario, cunha media de 7312‰ e valores que oscilan entre 6300‰ e 8410‰. Os ácidos palmítico e linoleico sitúanse despois en abundancia mentres que outros como o linolénico, o eicosenoico e o araquídico se encontran en menor proporción,

ningún por riba dos 6000‰ de media. En canto á dispersión, o ácido oleico presenta un rango total (valor máximo menos valor mínimo) elevado pero coherente tendo en conta que toma valores moi altos. Outros compostos como o palmitoleico, o linolénico e o eicosenoico amosan unha dispersión elevada en comparación ao seu valor medio, o que podería indicar unha maior variabilidade relativa ou unha asimetría na súa distribución. O resto, pola contra, presentan unha dispersión moderada. Esta disparidade no rango de valores de cada variable fai necesario aplicar unha estandarización previa á analise de agrupamento. Isto débese a que os métodos de *clustering* empregan distancias e estas diferenzas farían que os ácidos maioritarios dominasen o proceso de agrupamento, ocultando a influencia doutros compostos minoritarios.

```
Palmitic     Palmitoleic        Stearic          Oleic
Min.   : 610   Min.   : 15.00   Min.   :152.0   Min.   :6300
1st Qu.:1095   1st Qu.: 87.75   1st Qu.:205.0   1st Qu.:7000
Median :1201   Median :110.00   Median :223.0   Median :7302
Mean   :1232   Mean   :126.09   Mean   :228.9   Mean   :7312
3rd Qu.:1360   3rd Qu.:169.25   3rd Qu.:249.0   3rd Qu.:7680
Max.   :1753   Max.   :280.00   Max.   :375.0   Max.   :8410


Linoleic        Linolenic       Arachidic       Eicosenoic
Min.   : 448.0   Min.   : 0.00   Min.   :  0.0   Min.   : 1.00
1st Qu.: 770.8   1st Qu.:26.00   1st Qu.: 50.0   1st Qu.: 2.00
Median :1030.0   Median :33.00   Median : 61.0   Median :17.00
Mean   : 980.5   Mean   :31.89   Mean   : 58.1   Mean   :16.28
3rd Qu.:1180.8   3rd Qu.:40.25   3rd Qu.: 70.0   3rd Qu.:28.00
Max.   :1470.0   Max.   :74.00   Max.   :105.0   Max.   :58.00
```

Figura 5.1: Resumo estatístico das concentracións de ácidos graxos

Na Figura 5.2, obsérvase como a matriz de correlación mostra fortes relacións lineais entre os ácidos graxos. Destaca, por exemplo, a alta correlación positiva entre o ácido palmítico e o palmitoleico ou entre o linoleico e o palmitoleico, o que indica que estes compostos tenden a aparecer xuntos en proporcións similares. Pola contra, tamén se observa unha forte correlación negativa entre outros como o oleico e o linoleico indicando que, se un aumenta, o outro tende a diminuír. Esto mostra a existencia de información redundante o que pode afectar negativamente á análise.

Para paliar este efecto, realízase unha análise de compoñentes principais (ACP), coa finalidade de reducir a dimensionalidade dos datos mantendo ao mesmo tempo a maior parte da súa variabilidade. A Figura 5.3 mostra a proporción de varianza explicada por cada compoñente principal. Obsérvase que as catro primeiras compoñentes acumulan o 91,21 % da varianza total dos datos, o cal resulta suficiente para capturar os seus patróns principais. Isto permite substituír as oito variables orixinais por catro compoñentes principais non correlacionadas sobre as que rea-

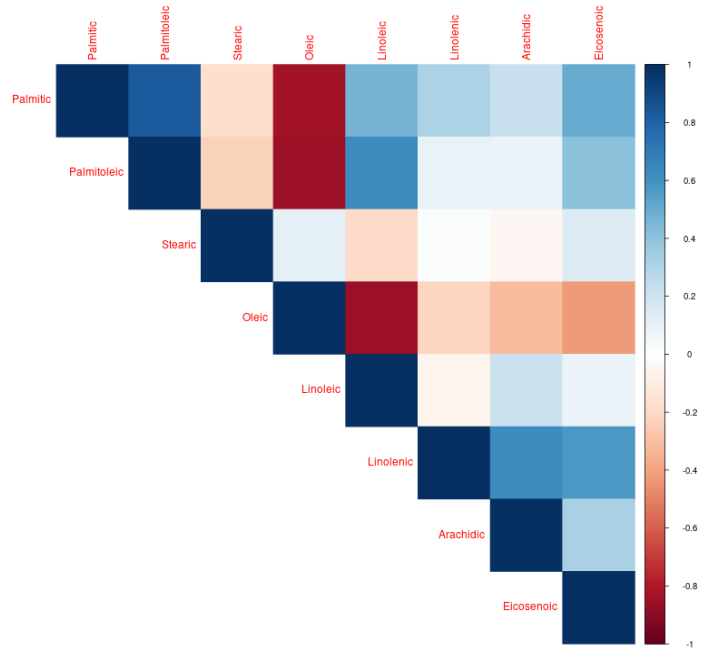


Figura 5.2: Matriz de correlación de Pearson entre os ácidos graxos.

lizar os distintos algoritmos de *clustering*. Posteriormente, os resultados obtidos representaranse no espazo bidimensional definido polas dúas primeiras compoñentes principais (PC1 e PC2), co obxectivo de facilitar a súa interpretación visual e comparar a estrutura dos clusters.

PC	Varianza	Acumulada
PC1	0.4652	0.4652
PC2	0.2207	0.6859
PC3	0.1270	0.8129
PC4	0.0991	0.9121
PC5	0.0417	0.9538
PC6	0.0311	0.9849
PC7	0.0149	0.9997
PC8	0.0003	1.0000

Figura 5.3: Proporción de varianza explicada por cada compoñente principal e acumulada

Mediante o tratamento destes datos, buscaremos identificar grupos naturais dentro da mostra baseados exclusivamente na composición química dos aceites, sen empregar información previa como a súa orixe. Isto permite que posteriormente estes grupos se poidan asociar a factores como a xeografía, o proceso de producción ou a variedade da oliva.

5.2. Análise dos datos

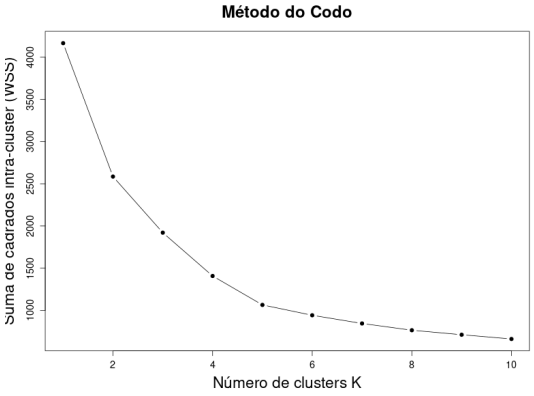
Antes de aplicar os diferentes algoritmos de *clustering*, imos realizar unha análise para estimar o número óptimo de clusters k presentes na estrutura dos datos. Para iso, recorremos ao método do codo, o método da silueta e o Gap Statistic, presentados na Sección 1.1.2.

Na Figura 5.4 obsérvanse os resultados obtidos empregando cada un destes métodos, onde cada un suxire un valor distinto de k, reflexo das distintas aproximacións que empregan para avaliar os datos. O método do codo mostra un punto de inflexión en $k = 3$, o que suxire que dividir os datos en tres grandes grupos permite capturar gran parte da estrutura dos datos sen necesidade de aumentar a complexidade. Por outro lado, o coeficiente de silueta que mide tanto a cohesión interna dos clusters como a separación con respecto aos demais, acadou o seu valor máximo en $k = 5$. Isto indica que, con cinco grupos, os elementos están ben agrupados cos seus semellantes e afastados doutros clusters, o que favorece unha estrutura ben definida. Finalmente, o GAP Statistic suxire un valor de $k = 8$, o que indica que existen oito grupos significativamente mellor definidos que os que se observarían baixo unha distribución aleatoria, sen patróns.

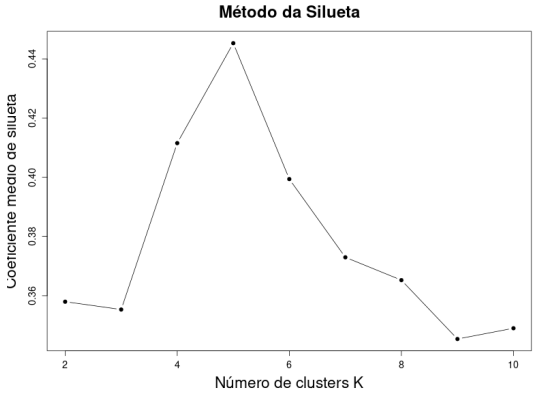
Estas diferenzas nos valores de k poden indicar que a distribución dos datos non presenta clusters completamente separados e uniformes senón que se pode corresponder cunha estrutura máis complexa onde existen subgrupos máis pequenos contidos noutros máis grandes, o que sería consistente coa natureza dos datos xa que factores como a rexión de producción, a variedade da oliva ou os métodos de producción poden variar tanto a gran escala (áreas grandes) como a pequena escala (áreas máis pequenas ou produtores).

A continuación, aplícanse os métodos particionais k-medias e k-medoides, ambos empregando un número de clusters fixado en $k = 3$. Dado que non existe un consenso sobre cal dos métodos de selección de k é preferible, poderíase continuar aplicando os métodos de *clustering* para calquera dos resultados previos, máis neste caso escolleremos un valor de $k = 3$, co obxectivo de escoller unha partición interpretable e non caer nunha complexidade excesiva. Na Figura 5.5 móstranse os resultados obtidos por ambos algoritmos. Observamos que k-medias produce grupos de forma máis compacta e simétrica arredor do seu céntroide, especialmente visible no cluster vermello; mentres que os grupos xerados polo k-medoides son algo máis alongados, aínda que moi similares. Tamén destaca o comportamento entre as fronteiras dos clusters, xa que k-medias presenta fronteiras máis ríxidas e perfectas, o que pode dar lugar a separacións artificiais especialmente nas zonas de baixa densidade. Porén, k-medoides tende a evitar estas asignacións abruptas dando lugar a unha separación potencialmente máis representativa.

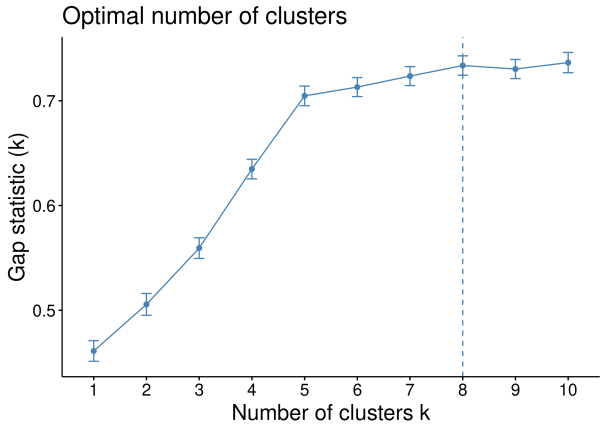
Ademais, tendo en conta a hipótese de que os datos poden poseer unha estrutura xerárquica, tamén se aplica o método de clustering xerárquico aglomerativo empregando o criterio de ligazón



(a) Método do Codo



(b) Método da Silueta



(c) GAP Statistic

Figura 5.4: Comparación dos tres métodos empregados para estimar o número óptimo de clusters.

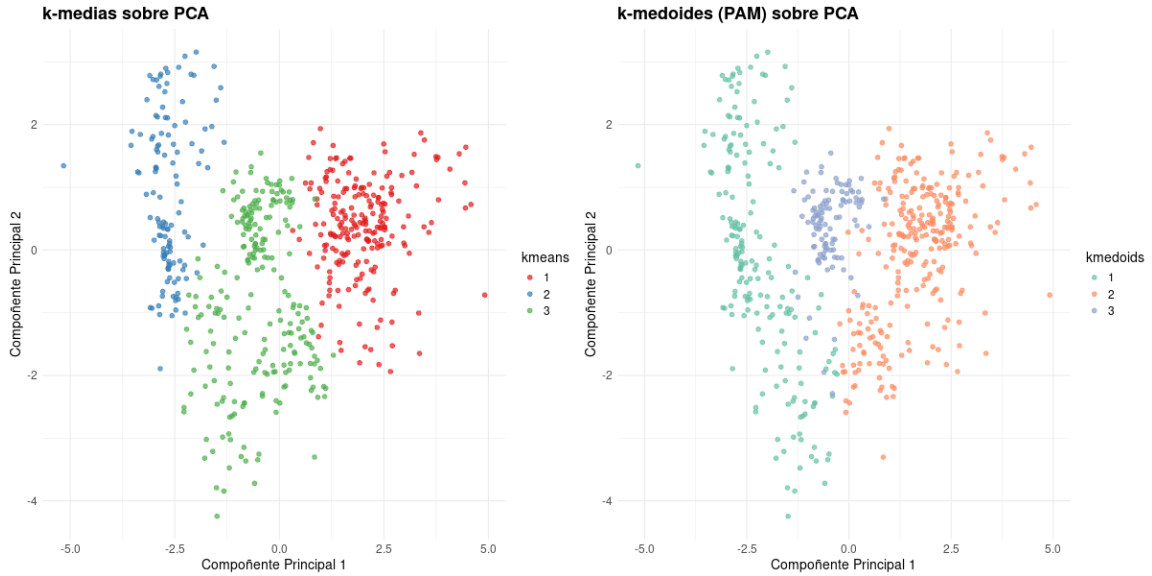


Figura 5.5: Comparativa entre k-medias e k-medoides.

de Ward. A Figura 5.6 mostra o dendrograma resultante incluíndo un corte a altura correspondente $k = 3$, dando lugar a tres grupos diferenciados. Esta elección baséase na observación dun salto apreciable na altura das fusións a partir dese punto, indicando que as unións posteriores implican grupos moito máis disimilares entre si.

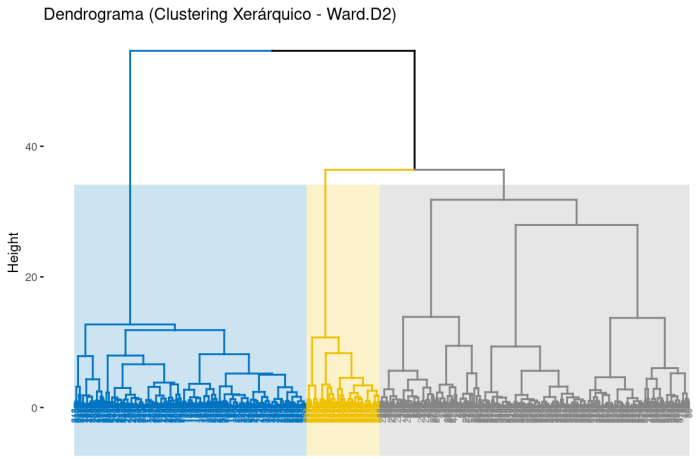


Figura 5.6: Dendrograma xerado cos datos *olive*.

Tal e como se mencionou no Capítulo 2, podemos ver como de ben este dendrograma representa os nosos datos mediante o *coeficiente de correlación cofenética* que neste caso concreto ten un valor de 0.658. Esto indícanos que, aínda que hai unha estrutura xerárquica presente nos datos, esta non é completamente dominante.

Para comparar enfoques baseados en modelos e baseados en densidade, aplicáronse os algoritmos EM mediante mesturas gaussianas e DBSCAN. Na Figura 5.7, obsérvase que o algoritmo EM, configurado para identificar tres compoñentes, modela os clusters como distribucións gaussianas elípticas. Cada elipse representa a zona na que se espera que estea a maior parte dos puntos do cluster e, como podemos observar, existe unha superposición considerable entre os tres clusters. Isto pode deberse a que os clusters presentan unha estrutura interna similar, a que non existen fronteiras claras entre eles ou incluso a unha mala elección do número de clusters. Pola súa parte DBSCAN distingue automaticamente catro clusters principais e un conxunto de puntos identificados como ruído e representados de cor gris. Este resultado dáse fixando $MinPts = 5$ e escollendo $Eps = 0{,}9$ mediante o gráfico da k-distancia explicado no Sección 4.1. Chama a atención un pequeno grupo de puntos en cor rosa (o cluster 4) que poderían parecer visualmente integrables no cluster 1 dada a súa proximidade. O motivo podería ser que DBSCAN non só ten en conta a distancia aos puntos veciños, senón tamén a súa densidade local; se un grupo de puntos é suficientemente denso pero non está conectado directamente co núcleo doutro cluster, DBSCAN pode formar un grupo por separado e, en ocasións, isto pode dar lugar a unha sobresegmentación.

Para analizar o efecto que estes parámetros iniciais poden ter no resultado, aplicouse o algoritmo con distintos valores de $MinPts$ (2 e 10) e axustouse o seu valor de Eps correspondente. Os resultados aparecen recollidos na Figura 5.11. Vemos que se tomamos $MinPts = 2$, resulta un valor tan pequeno que DBSCAN detecta moitos microclusters e prodúcese unha sobresegmentación. Tamén destaca a gran cantidade de datos que son clasificados como ruído. No caso de $MinPts = 10$, a esixencia de rexións moi densas fai que se detecte un único cluster, o cal resulta excesivamente restrictivo.

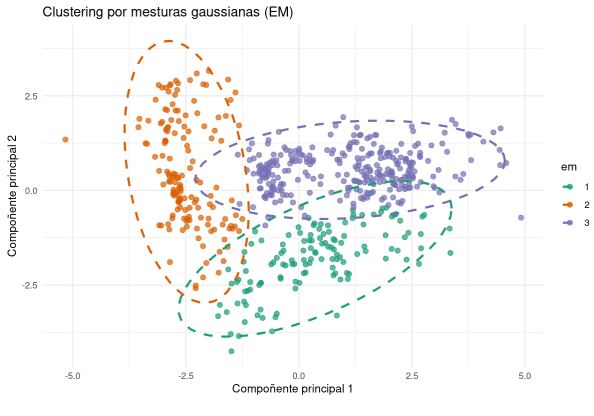


Figura 5.7: Clustering por mesturas gaussianas (EM).

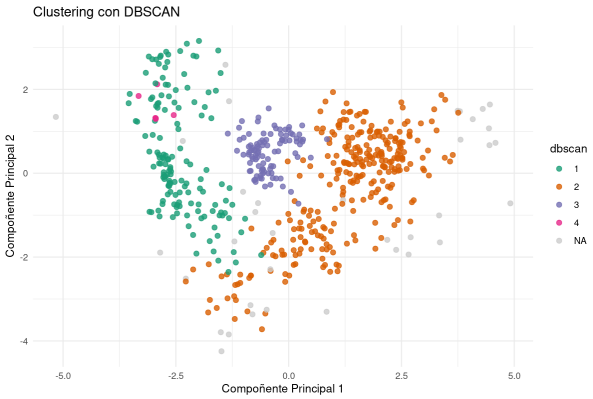


Figura 5.8: Clustering con DBSCAN.

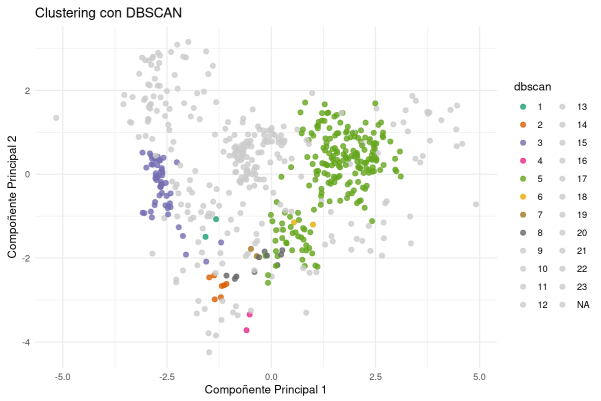


Figura 5.9: DBSCAN con MinPts = 2, Eps = 0,6

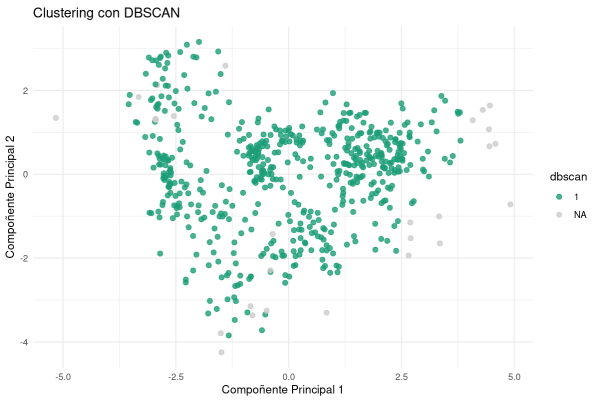


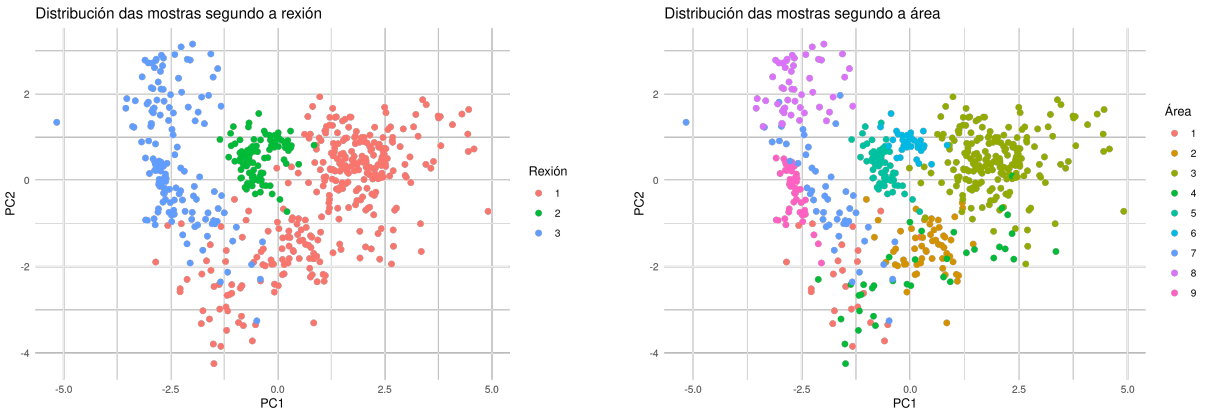
Figura 5.10: DBSCAN con MinPts = 10, Eps = 1,1

Figura 5.11: Comparación dos resultados de DBSCAN con diferentes configuracións de parámetros.

Unha vez identificados os grupos naturais mediante técnicas de *clustering*, resulta útil contrastar estes resultados coa información categórica dispoñible na base de datos, a rexión e a área de procedencia de cada mostra. Aínda que estes datos non foron empregados durante a análise, poden servir para a coherencia e validez dos grupos obtidos. Na Figura 5.12 móstranse as diferentes rexións e áreas nas que as variables categóricas dividen a mostra. Se nos fixamos especialmente no caso da variable rexión, que divide os datos en tres grupos, observamos unha forte semellanza co resultado obtido aplicando DBSCAN (ver figura 5.8). Isto reforza a validez da análise e mostra que a información química recollida reflicte patróns relacionados coa procedencia das mostras. Ademáis, tamén resalta o feito de que a variable área de procedencia divide á mostra en nove grupos, o que se achega moito ao valor de $k = 8$ suxerido polo método GAP Statistic.

5.3. Conclusións

A análise práctica do conxunto de datos *olive* permitiunos comparar os resultados de distintos algoritmos de clustering. Vemos así que cada un ofrece unha perspectiva diferente sobre a estrutura dos datos. Mentres uns favorecen agrupacións compactas e ben separadas, outros permiten identificar formas máis complexas ou detectar ruído. Este feito resalta a importancia de combinar diferentes enfoques para así obter unha visión máis completa dos datos, tendo en conta tamén o efecto que pode ter a selección de parámetros no resultado.



(a) Mostras coloreadas segundo a rexión.

(b) Mostras coloreadas segundo a área.

Figura 5.12: Visualización das mostras no espazo PCA, coloreadas segundo as variables categóricas reais.

Bibliografía

[Aggarwal and Reddy, 2013] Aggarwal, C. C. and Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition.

[Arthur and Vassilvitskii, 2007] Arthur, D. and Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, USA. Society for Industrial and Applied Mathematics.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231.

[Everitt et al., 2009] Everitt, B. S., Landau, S., and Leese, M. (2009). *Cluster Analysis*. Wiley Publishing, 4th edition.

[Giordani et al., 2020a] Giordani, P., Ferraro, M., and Martella, F. (2020a). *An Introduction to Clustering with R*. Springer Singapore.

[Giordani et al., 2020b] Giordani, P., Ferraro, M. B., and Martella, F. (2020b). *An Introduction to Clustering with R*. Behaviormetrics: Quantitative Approaches to Human Behavior Ser. ; v.1. Springer Singapore Pte. Limited, Singapore.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer.

[Jain, 2010] Jain, A. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31:651–666.

[Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2).

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pages 281–298. University of California press.

[Ng and Han, 1994] Ng, R. and Han, J. (1994). Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14:1003– 1016.

[Peña, 2010] Peña, D. (2010). *Analisis de datos multivariantes*. McGraw-Hill, Madrid.

[R Core Team, 2024] R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[Rousseeuw, 1987] Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65.

[Sokal and Rohlf, 1962] Sokal, R. and Rohlf, F. (1962). Sokal rr, rohlf fj. the comparison of dendrograms by objective methods. taxon 11: 33-40. *Taxon*, 11:33–40.

[Tan et al., 2005] Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., USA.

[Tibshirani et al., 2001] Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B*, 63:411–423.