USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Clustering basado en modelos

Nuria Gómez Sánchez del Valle

Junio, 2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Clustering basado en modelos

Nuria Gómez Sánchez del Valle

Junio, 2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Coñecemento: Estadística e Investigación Operativa
Título: Clustering basado en modelos
Breve descrición do contido
El análisis clúster consiste en la búsqueda automatizada de grupos de observaciones relacionadas dentro de un conjunto de datos. Si bien los métodos tradicionales, como aquellos basados en distancias, son intuitivos, presentan una limitación importante en la dificultad para definir de manera precisa qué constituye un clúster. El objetivo de este trabajo se centra en utilizar un enfoque estadístico que se apoya en la distribución subyacente de los datos. Específicamente, se explorará el clustering basado en modelos, el cual permite asociar clústeres a subpoblaciones mediante características, como las componentes de una mixtura, de la función de densidad.
Recomendacións
Outras observacións

Índice

Resumen

El clustering es una técnica estadística no supervisada que busca identificar automáticamente grupos homogéneos de observaciones dentro de un conjunto de datos. Su utilidad se ha consolidado en múltiples disciplinas, especialmente en el contexto actual de generación masiva de datos, gracias a su capacidad para identificar grupos en datos complejos y de alta dimensión. Aunque tradicionalmente se han utilizado métodos heurísticos como k-medias o técnicas jerárquicas, estos enfoques presentan limitaciones, como la falta de una base teórica sólida o la dificultad para determinar el número óptimo de grupos. En contraste, el clustering basado en modelos (*Model-Based Clustering*, MBC) ofrece una alternativa estadísticamente fundamentada al modelar los datos como una mixtura finita de distribuciones de probabilidad. Este enfoque permite realizar inferencias rigurosas, seleccionar modelos apropiados, elegir el número de grupos de manera justificada y evaluar la incertidumbre en la asignación de observaciones. En este trabajo, se presentan los fundamentos teóricos del clustering basado en modelos, con un enfoque en los modelos de distribuciones gaussianas, que son los más utilizados, así como el algoritmo EM para la estimación de parámetros y criterios de selección de modelos, incluyendo la elección del número de clústeres. Además, se presentan ejemplos prácticos utilizando el paquete `mclust` en R.

Resumo

O clustering é unha técnica estatística non supervisada que busca identificar automaticamente grupos homoxéneos de observacións dentro dun conxunto de datos. A súa utilidade consolidouse en múltiples disciplinas, especialmente no contexto actual de xeración masiva de datos, grazas á súa capacidade para identificar grupos en datos complexos e de alta dimensión. Aínda que tradicionalmente empregábanse métodos heurísticos como k-medias ou técnicas xerárquicas, estes enfoques presentan limitacións, como a falta dunha base teórica sólida ou a dificultade para determinar o número óptimo de grupos. Pola contra, o clustering baseado en modelos (*Model-Based Clustering*, MBC) ofrece unha alternativa estatisticamente fundamentada ao modelar os datos como unha mestura finita de distribucións de probabilidade. Este enfoque permite realizar inferencias rigorosas, seleccionar modelos axeitados, elixir o número de grupos de maneira xustificada e avaliar a incerteza na asignación de observacións. Neste traballo, preséntanse os fundamentos teóricos do clustering baseado en modelos, cun enfoque nos modelos de distribucións gaussianas, que son os máis empregados, así como o algoritmo EM para a estimación de parámetros e criterios de selección de modelos, incluíndo a elección do número de clústeres. Ademais, preséntase exemplos prácticos utilizando o paquete `mclust` en R.

Abstract

Clustering is an unsupervised statistical technique that aims to automatically identify homogeneous groups of observations within a dataset. Its usefulness has been consolidated across various disciplines, particularly in the current context of massive data generation, thanks to its ability to identify groups in complex and high-dimensional data. Although heuristic methods such as k-means or hierarchical techniques have traditionally been used, these approaches present limitations, such as the lack of a solid theoretical foundation or the difficulty in determining the optimal number of groups. In contrast, model-based clustering (MBC) offers a statistically grounded alternative by modeling the data as a finite mixture of probability distributions. This approach allows for rigorous inferences, the selection of appropriate models, justifiable determination of the number of groups, and the evaluation of uncertainty in the assignment of observations. This work presents the theoretical foundations of model-based clustering, with a focus on Gaussian mixture models, which are the most widely used, as well as the EM algorithm for parameter estimation and model selection criteria, including the choice of the number of clusters. Additionally, practical examples are presented using the `mclust` package in R.

Introducción

El clustering es un conjunto amplio de métodos y técnicas estadísticas multivariantes que buscan identificar de forma automática subconjuntos homogéneos de observaciones dentro un conjunto de datos, es decir, particionar observaciones similares en grupos o clústeres significativos y útiles. El objetivo es encontrar grupos cuyos miembros compartan algo en común que no comparten con los miembros de otros grupos. El clustering es un ejemplo de aprendizaje no supervisado, ya que la presencia y el número de grupos pueden no conocerse de antemano y no se dispone de etiquetas para las observaciones.

El clustering puede entenderse como un proceso de categorización automática de objetos en grupos basados en sus características observadas. Agrupar objetos según lo que tienen en común es una práctica universal humana. Platón formalizó esta idea con su Teoría de las Formas, y Aristóteles pudo haber sido el primero en implementarla empíricamente, al clasificar animales en grupos según sus características en su Historia de los animales, labor ampliada por Linneo con su sistema de clasificación biológica o taxonomía de animales y plantas. Aristóteles y Linneo clasificaban objetos de manera subjetiva, sin embargo, el clustering va más allá, utilizando métodos numéricos sistemáticos.

En la era actual, caracterizada por un aumento masivo en la generación de datos, el clustering se ha convertido en una herramienta fundamental para extraer conocimiento útil de grandes volúmenes de información. Su importancia radica en su capacidad para identificar automáticamente grupos homogéneos dentro de datos complejos y de alta dimensión, lo que resulta crucial en numerosos campos como la salud, la biología, el comercio, el análisis web o el procesamiento de imágenes. Desde sus aplicaciones iniciales en investigación de mercados y taxonomía, el clustering ha cobrado aún más relevancia con la aparición de nuevos tipos de datos, como los genéticos o los derivados de Internet, y ha demostrado su utilidad en tareas tan diversas como segmentar imágenes médicas, analizar patrones de consumo, agrupar documentos o usuarios en línea, y comprimir imágenes mediante la agrupación de colores similares. Para conocer la historia del clustering hasta 1988, consultar Blashfeld y Aldenderfer (1988).

Se han propuesto muchos enfoques de clustering en la literatura a lo largo del último siglo, en gran parte desarrollados al margen de la estadística tradicional, que se basa en especificar un modelo probabilísticos para los datos. Los métodos tradicionales son de naturaleza combinatoria, y pueden ser jerárquicos (aglomerativos o divisivos) o de particionado, generalmente basados en distancias (por ejemplo, k-medias). Para una revisión, véase Saxena *et al.* (2017). Aunque algunos están ligeramente relacionados con modelos estadísticos, en general, se basan en procedimientos heurísticos que no hacen suposiciones explícitas sobre la estructura de los grupos, como ocurre en el caso de k-medias, que simplemente busca particionar los datos minimizando la varianza dentro de cada grupo sin asumir una distribución subyacente, o los métodos jerárquicos aglomerativos, que construyen una jerarquía de agrupamientos basándose en distancias entre observaciones o grupos. La elección del método de agrupamiento, las medidas de similitud y la interpretación han tendido a ser informales y, con frecuencia, subjetivas. Generalmente, son métodos heurísticos y algorítmicos, y dejan sin responder varias preguntas clave, tales como: ¿Qué método de clustering deberíamos usar? ¿Cuántos clústeres hay?¿Cómo podemos evaluar la incertidumbre sobre una partición estimada?

En los años 60, se reconoció que el clustering podía fundamentarse estadísticamente al enmarcarse como un problema de inferencia sobre un modelo de mixtura finita, en el que cada grupo se modela mediante su propia distribución de probabilidad. Esto permitió que el clustering se beneficiara del marco inferencial de la estadística, proporcionando respuestas fundamentadas y reproducibles a las preguntas mencionadas anteriormente. El clustering basado en modelos (*Model-Based Clustering*, MBC) es un enfoque probabilístico en el que cada grupo corresponde a una componente de una mixtura descrita por una distribución de probabilidad con parámetros desconocidos. Basar el clustering en un modelo probabilístico tiene varias ventajas. En esencia, sitúa el clustering dentro del marco de la metodología estadística estándar y permite realizar inferencias de manera rigurosa. También proporciona una manera fundamentada de elegir el número de clústeres. De hecho, la elección del modelo y del número de clústeres puede verse como un único problema de selección de modelos. Existe un equilibrio entre ambas decisiones: a menudo, si se elige un modelo más simple, se necesitarán más clústeres para representar los datos adecuadamente. Este enfoque también permite evaluar la incertidumbre en la asignación de las observaciones a los grupos.

No obstante, este enfoque conlleva ciertos desafíos, como la necesidad de asumir una forma funcional específica para los datos (por ejemplo, la distribución gaussiana), lo cual puede ser restrictivo si esta no se ajusta bien a la realidad, o bien volverse computacionalmente costoso en contextos de alta dimensión o con grandes volúmenes de datos, lo que en algunos casos puede hacer preferibles métodos tradicionales más simples y eficientes, especialmente cuando se busca rapidez o cuando la estructura de los datos no justifica un modelo complejo. A pesar de estas limitaciones, los métodos basados en modelos son cada vez más preferidos frente a los

métodos heurísticos de clustering, ya que permiten abordar de manera formal y estadísticamente fundamentada las cuestiones clave que los enfoques heurísticos dejan sin resolver, como qué método utilizar, cuántos grupos hay o cómo evaluar la incertidumbre asociada a una partición estimada, reduciéndolas a problemas estadísticos estándar, como la estimación de parámetros y la selección de modelos. Además, la relación uno a uno entre componentes de la mixtura y grupos puede flexibilizarse, como se discute en la Sección 2.3.

El modelo probabilístico subyacente al clustering basado en modelos es una mixtura finita de distribuciones multivariadas, que se describirá en el Capítulo 1. El tipo de distribución suele especificarse a priori (la más común es la gaussiana), mientras que la estructura del modelo (incluyendo el número de componentes) debe determinarse mediante técnicas de estimación de parámetros y selección de modelos. En particular, en la Sección 1.3, se presenta el enfoque de máxima verosimilitud para la estimación de parámetros y se detalla el algoritmo EM, prestando atención a su inicialización, un aspecto clave en el clustering basado en modelos.La metodología para realizar clustering mediante modelos de mixtura se detalla en el Capítulo 2. Asimismo, se describe el procedimiento de máxima a posteriori (MAP) como una estrategia para obtener una partición probabilística (Sección 2.1), asignando cada observación a la componente con mayor probabilidad a posteriori, lo que además permite evaluar y representar la incertidumbre en la asignación. También se presentan métodos para la selección de modelos basados en criterios de información y pruebas de razón de verosimilitud (Sección 2.2).

La mixtura finita de distribuciones gaussianas, descrita en la Sección 1.1, constituye una de las clases más flexibles de modelos para variables continuas y, por ello, se ha convertido en un método de referencia en una gran variedad de contextos, siendo el modelo más popular para el clustering de datos continuos. Por esta razón, el caso gaussiano se analiza en profundidad. Se introduce una descomposición de las matrices de covarianza de las componentes, que permite regularizar el procedimiento de estimación e imponer restricciones geométricas, como se describe en la Sección 1.1.1. Al imponer que el volumen, la forma y/o la orientación de las matrices de covarianza sean iguales entre componentes, se obtienen modelos más simples y fácilmente interpretables, adecuados para diferentes contextos de clustering. Además, en la Sección 2.3, se aborda cómo flexibilizar la relación uno a uno entre componentes de la mixtura y clústeres para ampliar la aplicabilidad del enfoque. En el Capítulo 3, se realiza un análisis ilustrativo en R utilizando `mclust`, un paquete de software ampliamente utilizado en el entorno estadístico R para clustering basado en modelos de mixtura gaussiana, con el objetivo de ejemplificar y consolidar los conceptos tratados a lo largo del trabajo. Finalmente, en el Capítulo 4 se presentan las conclusiones.

Capítulo 1

Modelos de mixtura finita

Sea $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ una muestra aleatoria de tamaño n, en la que cada $\mathbf{Y}_j$ es un vector aleatorio p-dimensional con una función de densidad de probabilidad $f(\mathbf{y}_j)$ definida sobre $\mathbb{R}^p$. En términos prácticos, $\mathbf{Y}_j$ representa el vector aleatorio asociado a las p mediciones realizadas en el j-ésimo registro, relacionadas con las características del fenómeno bajo estudio. Definimos $\mathbf{Y} = \left(\mathbf{Y}_1^\top, \ldots, \mathbf{Y}_n^\top\right)^\top$, donde el superíndice $\top$ indica la transposición vectorial. Es importante notar que $\mathbf{Y}$ se utiliza para representar la muestra completa; es decir, $\mathbf{Y}$ es una n-tupla de puntos en $\mathbb{R}^p$. Denotaremos una muestra observada de un vector aleatorio utilizando la letra minúscula correspondiente, de modo que $\mathbf{y} = \left(\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top\right)^\top$ representa una muestra aleatoria observada de n observaciones multivariadas $\mathbf{y}_j = (y_{j,1}, \ldots, y_{j,p})$, donde cada $\mathbf{y}_j$ es el valor observado del vector aleatorio $\mathbf{Y}_j$.

Un modelo de mixtura finita representa la distribución de probabilidad o la función de densidad f de $\mathbf{Y}_j$ como una mixtura finita o promedio ponderado de g funciones de densidad de probabilidad:

$$f(\mathbf{y}_j) = \sum_{i=1}^{g} \pi_i f_i(\mathbf{y}_j), \tag{1.1}$$

donde $f_i(\mathbf{y}_j)$ son funciones de densidad y π_i son cantidades no negativas que suman uno; es decir,

$$0 < \pi_i < 1 \quad (i = 1, \ldots, g) \quad \text{y} \quad \sum_{i=1}^{g} \pi_i = 1. \tag{1.2}$$

Las cantidades $\pi_1, \ldots, \pi_g$, denominadas proporciones de mixtura o pesos, representan la probabilidad de que una observación haya sido generada por la i-ésima componente. Nos referimos a las f_i como densidades, las cuales pueden adaptarse al contexto en el que el vector aleatorio $\mathbf{Y}_j$ es discreto mediante la adopción de la medida de conteo. Dado que las funciones $f_1, \ldots, f_g$ son funciones de densidad, es obvio que la Ecuación (1.1) define una densidad. Las f_i, $i \in \{1, \ldots, g\}$,

se denominan las componentes de la mixtura. Nos referiremos a la densidad (1.1) como una densidad de mixtura finita con g componentes y a su función de distribución correspondiente $F(\mathbf{y}_j)$ como una función de distribución de mixtura finita con g componentes. Dado que nos centraremos exclusivamente en mixturas finitas de distribuciones, en lo sucesivo nos referiremos a los modelos de mixtura finita simplemente como modelos de mixtura. En esta formulación del modelo de mixtura, el número de componentes g se considera fijo. Pero, por supuesto, en muchas aplicaciones, el valor de g es desconocido y debe inferirse a partir de los datos disponibles, junto con las proporciones de mixtura y los parámetros para las densidades de las componentes.

Típicamente, las densidades componentes $f_i(\mathbf{y}_j)$ se especifican dentro de alguna familia paramétrica. En este caso, dichas densidades $f_i(\mathbf{y}_j)$ se definen como $f_i(\mathbf{y}_j;\boldsymbol{\theta}_i)$, donde $\boldsymbol{\theta}_i$ es el vector de parámetros que caracteriza la densidad de la i-ésima componente en la mixtura. La densidad de mixtura $f(\mathbf{y}_j)$ se puede expresar como:

$$f(\mathbf{y}_j;\boldsymbol{\Psi}) = \sum_{i=1}^{g} \pi_i f_i(\mathbf{y}_j;\boldsymbol{\theta}_i), \tag{1.3}$$

donde el vector $\boldsymbol{\Psi}$, que contiene todos los parámetros del modelo de mixtura, se define como:

$$\boldsymbol{\Psi} = \left(\pi_1, \ldots, \pi_{g-1}, \boldsymbol{\xi}^\top\right)^\top, \tag{1.4}$$

donde $\boldsymbol{\xi}$ es el vector que contiene todos los parámetros $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_g$, que se asume que son distintos a priori. Se denota Ω como el espacio de parámetros especificado para $\boldsymbol{\Psi}$. Como las proporciones de mixtura π_i suman 1, una de ellas es redundante. En la definición de $\boldsymbol{\Psi}$, se omite arbitrariamente la g-ésima proporción π_g.

En muchas aplicaciones, se asume que las densidades componentes pertenecen a la misma familia paramétrica, por ejemplo, la normal multivariada. En este caso, la densidad de mixtura evaluada en el vector $\mathbf{y}_j$, $f(\mathbf{y}_j;\boldsymbol{\Psi})$, tiene la forma:

$$f(\mathbf{y}_j;\boldsymbol{\Psi}) = \sum_{i=1}^{g} \pi_i g(\mathbf{y}_j;\boldsymbol{\theta}_i), \tag{1.5}$$

donde $g(\mathbf{y}_j;\boldsymbol{\theta})$ representa un miembro genérico de la familia paramétrica,

$$\{g(\mathbf{y}_j;\boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}, \tag{1.6}$$

siendo Θ el espacio de parámetros.

En la Sección 1.1, veremos uno de los modelos de mixturas más empleado en la práctica, el modelo de mixtura gaussiana.

1.1. Modelos de mixturas gaussianas

Las mixturas de distribuciones gaussianas son el modelo más popular para datos continuos, es decir, datos numéricos que, en teoría, pueden medirse en unidades infinitamente pequeñas. Los modelos de mixtura gaussiana (GMM, por sus siglas en inglés) se utilizan ampliamente en aprendizaje estadístico, reconocimiento de patrones y minería de datos (Celeux y Govaert, 1995; Fraley y Raftery, 2003; Stahl y Sallis, 2012).

Así, en la práctica a menudo se asume que las componentes pertenecen a la familia normal, dando lugar a mixturas normales o gaussianas. Para componentes normales multivariadas, la función de densidad de probabilidad de un GMM puede escribirse como, véase la Ecuación (1.5),

$$f\left(\mathbf{y}_j;\boldsymbol{\Psi}\right)=\sum_{i=1}^{g}\pi_i\phi\left(\mathbf{y}_j;\mu_i,\Sigma_i\right), \tag{1.7}$$

donde

$$\phi\left(\mathbf{y}_j;\mu_i,\Sigma_i\right)=(2\pi)^{-p/2}\left|\Sigma_i\right|^{-1/2}\exp\left(-\frac{1}{2}\left(\mathbf{y}_j-\mu_i\right)^{\top}\Sigma_i^{-1}\left(\mathbf{y}_j-\mu_i\right)\right), \tag{1.8}$$

denota la densidad de la normal multivariada con media (vector) μ_i y matriz de covarianza Σ_i $(i=1,\ldots,g)$. En este caso, el vector $\boldsymbol{\Psi}$ de parámetros desconocidos se define como:

$$\boldsymbol{\Psi}=\left(\pi_1,\ldots,\pi_{g-1},\boldsymbol{\xi}^{\top}\right)^{\top},$$

donde $\boldsymbol{\xi}$ está formado por las medias de las componentes, $\mu_1,\ldots,\mu_g$, y los distintos elementos de las matrices de covarianza de las componentes, $\Sigma_1,\ldots,\Sigma_g$, en el caso no restringido (heterocedástico) donde estas pueden ser desiguales; es decir, no se les imponen restricciones, más allá de las habituales (ser simétrica y definida positiva). En el caso de que las matrices de covarianza de las componentes Σ_i se restrinjan a ser iguales (homocedástico),

$$\Sigma_i=\Sigma \quad (i=1,\ldots,g),$$

$\boldsymbol{\xi}$ consta de las medias de las componentes, $\mu_1,\ldots,\mu_g$, y los distintos elementos de la matriz de covarianza común a todas las componentes, Σ. Esta es una de las restricciones que se pueden imponer para simplificar el modelo.

A continuación, en la Sección 1.1.1, presentaremos otras simplificaciones del espacio de parámetros que se suelen emplear en la práctica.

1.1.1. Parametrizaciones de la matriz de covarianza

El modelo completo de mixtura finita normal multivariada especificado por la Ecuación (1.7) ha demostrado ser útil en muchas aplicaciones. Sin embargo, cuenta con $(g-1)+gp+g\{p(p+1)/2\}$ parámetros, lo que puede ser un número considerable si la dimensión p o el número de componentes g son "grandes". Un gran número de parámetros puede generar tanto dificultades en la estimación, como falta de precisión o incluso degeneraciones, además de problemas en la interpretación de los resultados.

Para aliviar este problema, es común especificar versiones más simplificadas del modelo. Se expondrá más adelante una forma de lograrlo en la que se utiliza la descomposición en valores propios de las matrices de covarianza de las componentes Σ_i, en la forma

$$\Sigma_i = \lambda_i D_i A_i D_i^\top \tag{1.9}$$

donde D_i es la matriz de autovectores de Σ_i, $A_i = \text{diag}(A_{1,i}, \ldots, A_{p,i})$ es una matriz diagonal cuyos elementos son proporcionales a los autovalores de Σ_i en orden descendente, y λ_i es la constante de proporcionalidad asociada.

Los datos generados por un GMM se caracterizan por grupos o clústeres centrados en la media μ_i, con mayor densidad para los puntos más cercanos a la media. Las superficies de nivel de densidad constante son elipsoides cuyas características geométricas (como volumen, forma y orientación) están determinadas por las matrices de covarianza Σ_i. Cada factor en la descomposición (1.9) corresponde a una propiedad geométrica particular de la componente i de la mixtura. La matriz ortogonal de autovectores D_i determina su orientación en $\mathbb{R}^p$. La matriz diagonal de autovalores escalados A_i determina su forma. Por ejemplo, si $A_{1,i} \gg A_{2,i}$, entonces la componente i de la mixtura está concentrado alrededor de una línea en $\mathbb{R}^p$. Si $A_{1,i} \approx A_{2,i} \gg A_{3,i}$, entonces la componente i está concentrada alrededor de un plano bidimensional en $\mathbb{R}^p$. Si todos los valores de $A_{j,i}$ son aproximadamente iguales, entonces la componente i es aproximadamente esférica. La constante de proporcionalidad λ_i determina el volumen ocupado por la componente i en $\mathbb{R}^p$, que es proporcional a $\lambda_i^p|A_i|$. Es común restringir la matriz A_i para que su determinante sea igual a 1, en cuyo caso $\lambda_i = |\Sigma_i|^{1/p}$ determina directamente el volumen de la componente i, siendo este proporcional a λ_i^p.

Debido a estas interpretaciones geométricas, la descomposición (1.9) se denomina a veces descomposición geométrica o descomposición Volumen-Forma-Orientación (VFO).

La simplificación del modelo de mixtura finita dado por la Ecuación (1.7) puede lograrse de diversas formas utilizando la descomposición (1.9). Las características de las distribuciones de las componentes, como el volumen, la forma y la orientación, son desconocidas, ya que dependen

de la matriz de covarianza, y se estiman a partir de los datos. Estas características pueden variar entre los clústeres o restringirse para que sean iguales en todos ellos. En consecuencia, (λ_i, A_i, D_i) pueden tratarse como conjuntos independientes de parámetros. Si dos componentes i y i' de la mixtura cumplen que $\lambda_i = \lambda_{i'}$, tendrán el mismo volumen; si $A_i = A_{i'}$, tendrán la misma forma; y si $D_i = D_{i'}$, tendrán la misma orientación. Así, cualquiera o todas las propiedades geométricas (volumen, forma u orientación) pueden restringirse para ser iguales entre las componentes de la mixtura. Además, la matriz de covarianza de cualquier componente puede forzarse a ser esférica (es decir, proporcional a la matriz identidad I) o diagonal.

Esto permite dos posibles modelos univariados y 14 posibles modelos en el caso multivariado. En el caso univariado se contempla la opción de permitir que la varianza varíe entre las componentes, modelo etiquetado como "V", o, alternativamente, imponer que todas tengan la misma varianza, modelo etiquetado como "E" ("E" y "V" son las siglas de igual y variable en inglés). En el caso multivariado se pueden obtener 14 posibles modelos al variar la parametrización de las matrices de covarianza de las componentes, variando así las características geométricas de sus distribuciones. Estos posibles modelos multivariados se muestran en la Tabla 1.1. Cada uno de estos posibles modelos se etiqueta con un identificador de tres letras, donde la primera letra es "E" si los volúmenes de los clústeres están restringidos a ser iguales, y "V" si no lo están. De manera similar, la segunda letra es "E" si las matrices de forma A_i están restringidas a ser iguales entre los clústeres, de modo que $A_i = A$ para $i = 1, \ldots, g$, "V" si no están restringidas e "I" si los clústeres son esféricos, en cuyo caso $A_i = I$ para $i = 1, \ldots, g$, donde I es la matriz identidad. Finalmente, la tercera letra es igual a "E" si las matrices D_i de autovectores que especifican las orientaciones de los clústeres están restringidas a ser iguales, teniendo, por tanto, los clústeres la misma orientación, "V" si no lo están, e "I" si $D_i = I$ para $i = 1, \ldots, g$. Así, tal como se indica en la Tabla 1.1, cuando las matrices de covarianza de las componentes son esféricas (proporcionales a la identidad), es decir, cuando las matrices de forma A_i y las matrices de autovectores D_i son la matriz identidad, los clústeres son esféricos y, como la distribución es simétrica en todas las direcciones, no hay una orientación específica, ya que la uniformidad en la varianza hace que cualquier orientación sea equivalente. Además, cuando las matrices de covarianza de las componentes son diagonales, es decir, cuando las matrices de autovectores D_i son la matriz identidad, los clústeres tienen forma de elipses en las que la dispersión varía según la dirección, definiendo así una orientación alineada con los ejes coordenados. Por último, cuando las matrices de covarianza de las componentes tienen la forma completa, es decir, tanto las matrices de forma como las matrices de autovectores de las componentes no están restringidas a ser la matriz identidad, se permite que la distribución presente correlaciones entre las variables, lo que genera formas elipsoidales con posibles rotaciones y posibilita que los clústeres tengan cualquier orientación y elongación.

Modelo	$\boldsymbol{\Sigma_i}$	Distribución	Volumen	Forma	Orientación
EII	λI	Esférica	Igual	Igual	—
VII	$\lambda_i I$	Esférica	Variable	Igual	—
EEI	λA	Diagonal	Igual	Igual	Ejes coordenados
VEI	$\lambda_i A$	Diagonal	Variable	Igual	Ejes coordenados
EVI	λA_i	Diagonal	Igual	Variable	Ejes coordenados
VVI	$\lambda_i A_i$	Diagonal	Variable	Variable	Ejes coordenados
EEE	$\lambda DAD^\top$	Elipsoidal	Igual	Igual	Igual
VEE	$\lambda_i DAD^\top$	Elipsoidal	Variable	Igual	Igual
EVE	$\lambda DA_i D^\top$	Elipsoidal	Igual	Variable	Igual
VVE	$\lambda_i DA_i D^\top$	Elipsoidal	Variable	Variable	Igual
EEV	$\lambda D_i A D_i^\top$	Elipsoidal	Igual	Igual	Variable
VEV	$\lambda_i D_i A D_i^\top$	Elipsoidal	Variable	Igual	Variable
EVV	$\lambda D_i A_i D_i^\top$	Elipsoidal	Igual	Variable	Variable
VVV	$\lambda_i D_i A_i D_i^\top$	Elipsoidal	Variable	Variable	Variable

Tabla 1.1: Parametrizaciones de las matrices de covarianza $\Sigma_1 \dots \Sigma_g$ en el caso multivariado.

En la Figura 1.1, se representan gráficamente las características geométricas (volumen, forma y orientación) de las componentes de la mixtura para cada una de las 14 posibles parametrizaciones de la matriz de covarianza (descritas en detalle en la Tabla 1.1). Estas representaciones corresponden a un caso bidimensional ($d = 2$) con tres grupos ($g = 3$), lo que permite visualizar de manera intuitiva cómo se distribuyen las componentes en el plano. Junto a cada configuración, se incluye un conjunto de datos simulados generado a partir de una mixtura de distribuciones normales que sigue la parametrización correspondiente. De esta forma, se puede apreciar con claridad cómo las restricciones geométricas impuestas por cada una de las 14 parametrizaciones influyen directamente en la forma, volumen y orientación de los grupos resultantes. En particular, se hace evidente cómo ciertas parametrizaciones permiten una mayor flexibilidad en la modelización de estructuras de grupo complejas, mientras que otras introducen restricciones que simplifican el modelo, a costa de una menor capacidad de adaptación a determinadas formas de los datos.

La Tabla 1.2 muestra el número de parámetros necesarios para especificar la matriz de covarianza para cada modelo en dos casos: el caso bidimensional con dos componentes, $p = 2, g = 2$, y el caso de 27 dimensiones con tres componentes, $p = 27, g = 3$. Estos resultados se obtienen al notar que, para una componente de mixtura, el volumen se especifica con 1 parámetro, la forma con $(p - 1)$ parámetros, y la orientación con $p(p - 1)/2$ parámetros.

Es evidente que la ganancia potencial por simplificación, medida por el número de parámetros,

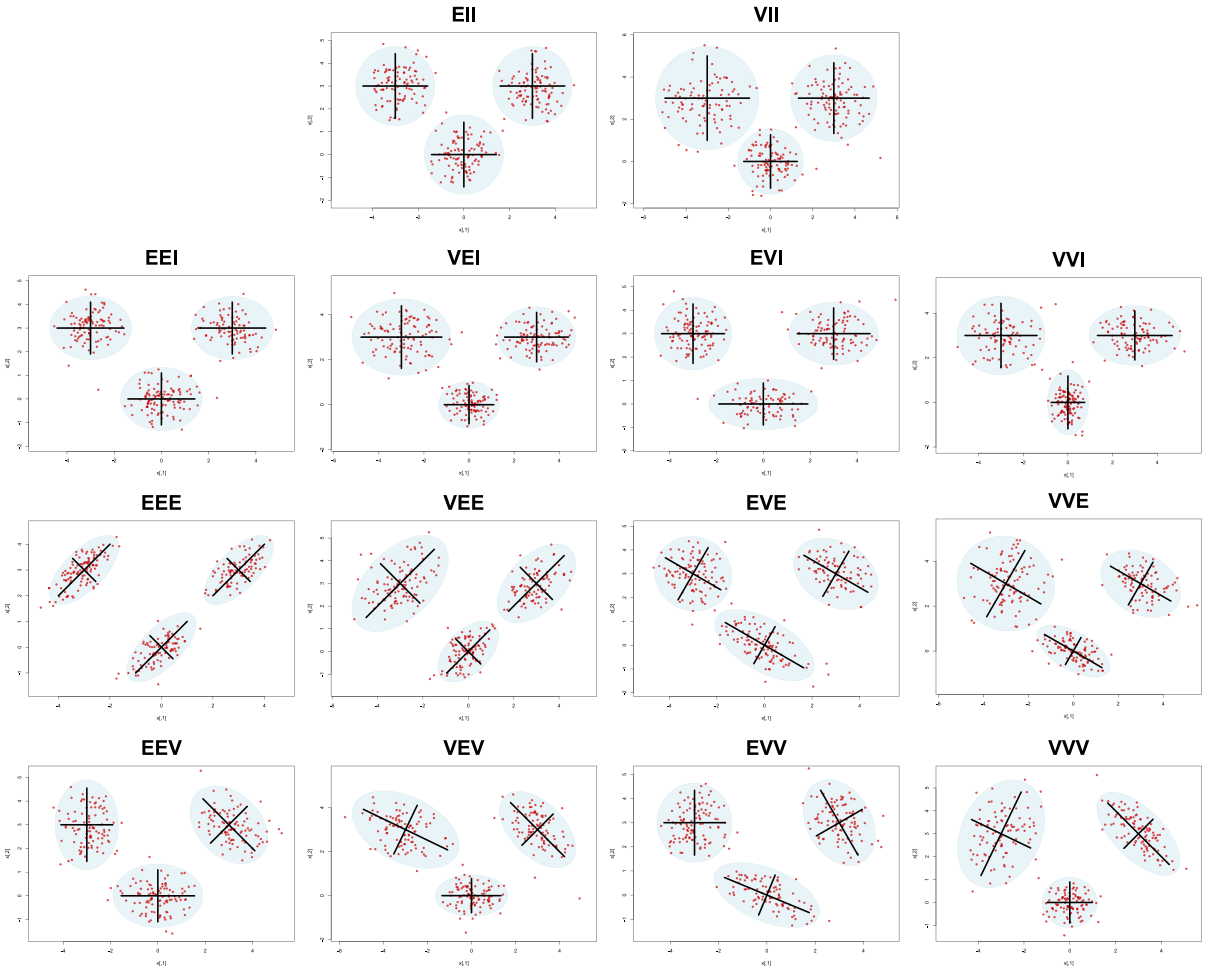

Figura 1.1: En rojo, datos generados de la mixtura de normales (1.7). En azul, superficies de nivel de densidad elípticas para cada uno de los 14 modelos gaussianos parametrizados mediante la descomposición espectral (1.9) de las matrices de covarianza de las componentes con las etiquetas de la Tabla 1.1, en el caso de tres grupos en dos dimensiones. La primera fila muestra los dos modelos esféricos *II, seguida de los cuatro modelos diagonales **I, luego los cuatro modelos de igual orientación **E, y finalmente los cuatro modelos de orientación variable **V. En las tres últimas filas, la primera columna muestra los modelos con igual volumen y forma para los tres grupos EE*; la segunda columna los modelos con igual forma para los tres grupos *E*;la tercera columna los modelos con igual volumen para los tres grupos E**; y la cuarta columna los modelos con volumen y forma variable VV*.

es pequeña para el caso bidimensional. Sin embargo, para casos de mayor dimensionalidad, la ganancia puede ser significativa. En el caso más extremo en la Tabla 1.2, en el caso de 27 dimensiones con 3 componentes de mixtura, el modelo VVV requiere 1,134 parámetros para representar las matrices de covarianza, mientras que el modelo EII requiere solo un parámetro.

Como se puede ver en la Tabla 1.2, se necesitan más parámetros para especificar la forma que el volumen, y más aún para especificar la orientación. Por lo tanto, se logran grandes ganancias

Modelo	Parámetros de Covarianza	$p = 2, g = 2$	$p = 27, g = 3$
EII	1	1	1
VII	g	2	3
EEI	p	2	27
VEI	$g + (p - 1)$	3	29
EVI	$1 + g(p - 1)$	3	79
VVI	gp	4	81
EEE	$p(p + 1)/2$	3	378
VEE	$g + (p + 2)(p - 1)/2$	4	380
EVE	$1 + (p + 2g)(p - 1)/2$	4	430
EEV	$1 + (p - 1) + g[(p - 1)(p)/2]$	5	1080
VVE	$g + (p + 2g)(p - 1)/2$	5	432
EVV	$1 + g[(p + 2)(p - 1)/2]$	5	1132
VEV	$g + (p - 1) + g[(p - 1)(p)/2]$	5	1082
VVV	$g[p(p + 1)/2]$	6	1134

Tabla 1.2: Número de parámetros necesarios para especificar las matrices de covarianza $\Sigma_1 \dots \Sigma_g$ en los diferentes modelos.

con los modelos que restringen que las orientaciones a ser iguales para todas las componentes de la mixtura, y las mayores ganancias provienen de imponer que las distribuciones de las componentes sean diagonales. Sin embargo, estos modelos más simplificados no siempre ajustan adecuadamente los datos. En particular, modelos como el EII (donde todas las componentes comparten la misma matriz de covarianza diagonal) restringen en exceso la forma de las distribuciones, ya que asumen que todas las componentes tienen la misma estructura (forma, volumen y orientación) y no capturan correlaciones entre las variables, lo que puede llevar a un ajuste inadecuado si los datos tienen una estructura más compleja o presentan correlaciones que no son capturadas por estas restricciones. De esta forma, aunque estos modelos son útiles por su simplicidad, pueden resultar inadecuados cuando la complejidad de los datos requiere una mayor flexibilidad en la forma de las distribuciones.

1.2. Interpretación de los modelos de mixtura finita

Los modelos de mixtura finita son una herramienta estadística efectiva para representar poblaciones heterogéneas, ya que permiten modelar datos que provienen de diferentes subgrupos. En este contexto, una situación obvia en la que el modelo de mixtura de g componentes (1.1) es directamente aplicable es cuando cada observación $\mathbf{Y}_j$ se extrae de una población G compuesta por g grupos, $G_1, \dots, G_g$, en proporciones $\pi_1, \dots, \pi_g$. Si la densidad de $\mathbf{Y}_j$ en el grupo G_i está dada por $f_i(\mathbf{y}_j; \boldsymbol{\theta}_i)$, entonces la densidad de $\mathbf{Y}_j$ tiene la forma de una mixtura de g componentes

(1.1). En esta situación, las g componentes de la mixtura pueden identificarse físicamente con los g grupos externamente existentes, $G_1, \ldots, G_g$.

Una forma obvia de generar un vector aleatorio $\mathbf{Y}_j$ con la densidad de mixtura de g componentes $f(\mathbf{y}_j)$ dada por (1.3) es a través de un modelo con variables latentes. Las variables latentes son variables auxiliares que no se observan directamente, pero que indican de qué grupo o componente proviene cada observación, permitiendo entender el modelo como una combinación de varios subgrupos dentro de los datos, donde cada observación proviene de una de estas subpoblaciones. A continuación, se analizará cómo los modelos de mixtura pueden entenderse en términos de variables latentes. Esta interpretación es clave ya que permite comprender el modelo de mixtura como una estructura que modela los datos de forma que cada observación proviene de una subpoblación, lo cual no solo proporciona una base probabilística sólida, sino que también facilita la estimación de los parámetros, la interpretación del modelo y su aplicación en tareas como la clasificación.

Sea $\mathbf{Z}_j$ una variable aleatoria discreta que toma los valores $1, \ldots, g$, con probabilidades $\pi_1, \ldots, \pi_g$, respectivamente, y supongamos que la densidad condicional de $\mathbf{Y}_j$ dado $\mathbf{Z}_j = i$ es $f_i(\mathbf{y}_j; \boldsymbol{\theta}_i)$ $(i = 1, \ldots, g)$. Entonces, la densidad marginal de $\mathbf{Y}_j$ se obtiene mediante la ley de probabilidad total y está dada por $f(\mathbf{y}_j; \boldsymbol{\psi})$ (véase la Ecuación (1.3)).

En este contexto, la variable $\mathbf{Z}_j$ puede interpretarse como la etiqueta de componente del vector de características $\mathbf{Y}_j$, es decir, $\mathbf{Z}_j$ indica a cuál de las g componentes pertenece $\mathbf{Y}_j$. En lo posterior, resulta conveniente trabajar con un vector de etiquetas g-dimensional $\mathbf{Z}_j$ en lugar de una única variable categórica $\mathbf{Z}_j$, donde el i-ésimo elemento de $\mathbf{Z}_j$, $Z_{ij} = (\mathbf{Z}_j)_i$, se define como uno o cero, según si la etiqueta de componente de $\mathbf{Y}_j$ en la mixtura es igual a i o no, es decir, cada Z_{ij} es 1 si $\mathbf{Y}_j$ pertenece al grupo i y 0 en caso contrario $(i = 1, \ldots, g)$. Así, $\mathbf{Z}_j$ se distribuye de acuerdo con una distribución multinomial, que denotaremos como $\mathbf{Z}_j \sim \text{Mult}_g(1, \boldsymbol{\pi})$, donde $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)^\top$, que consiste en una extracción sobre g categorías con probabilidades $\pi_1, \ldots, \pi_g$; es decir,

$$\mathbb{P}(\mathbf{Z}_j = \mathbf{z}_j) = \pi_1^{z_{1j}} \pi_2^{z_{2j}} \ldots \pi_g^{z_{gj}}, \quad \sum_{i=1}^{g} z_{ij} = 1.$$

A partir del valor observado $\mathbf{Y}_j = \mathbf{y}_j$, la probabilidad a posteriori de que provenga de la componente i puede obtenerse aplicando el Teorema de Bayes:

$$\tau_i(\mathbf{y}_j; \psi) = \mathbb{P}\{Z_{ij} = 1 \mid \mathbf{Y}_j = \mathbf{y}_j\} = \frac{\pi_i f_i(\mathbf{y}_j; \boldsymbol{\theta}_i)}{f(\mathbf{y}_j; \psi)} \quad (i = 1, \ldots, g;\ j = 1, \ldots, n). \tag{1.10}$$

donde π_i es la probabilidad inicial de pertenecer a la componente i, $f_i(\mathbf{y}_j;\boldsymbol{\theta}_i)$ es la densidad de la componente i y $f(\mathbf{y}_j;\psi)$ es la densidad de la mixtura. Tal como se verá en el siguiente capítulo, en la Sección 2.1, estas probabilidades son fundamentales para clasificar observaciones dentro de las componentes del modelo de mixtura.

Hay ejemplos en la práctica donde se sabe, a priori, que la población es una mixtura de g grupos distintos que existen en algún sentido físico. Sin embargo, también existen muchos ejemplos que implican el uso de modelos de mixtura en los que las componentes no pueden identificarse con grupos existentes externamente. En algunos casos, las componentes se introducen en el modelo de mixtura para permitir una mayor flexibilidad al modelar de una población heterogénea que, aparentemente, no puede ser modelada por una distribución de una sola componente. En el extremo, al aumentar el número de componentes hasta acercarse al número de observaciones, el modelo adquiere una flexibilidad tan grande que se aproxima a una representación no paramétrica de la distribución, véase la estimación tipo núcleo de la densidad (Silverman, 1986).

Por lo tanto, para valores del número de componentes g entre 1 y el tamaño de la muestra n, los modelos de mixtura pueden considerarse como un compromiso semiparamétrico entre el modelo completamente paramétrico, representado por una única familia paramétrica ($g = 1$), como la normal multivariante, y un enfoque altamente flexible que se aproxima a un modelo no paramétrico cuando g adopta un valor cercano a n.

Así, los enfoques basados en modelos de mixtura son paramétricos en el sentido de que se pueden especificar formas paramétricas $f_i(\mathbf{y}_j;\boldsymbol{\theta}_i)$ para las densidades de las componentes, pero también pueden considerarse no paramétricos al permitir que el número de componentes g aumente y se adapte a la propia estructura de los datos. Estos modelos poseen gran parte de la flexibilidad de los enfoques no paramétricos, al tiempo que conservan algunas ventajas de los enfoques paramétricos, como mantener la dimensión del espacio de parámetros en un tamaño razonable.

1.3. Estimación de los parámetros por máxima verosimilitud

A lo largo de los años, se han utilizado diversos enfoques para la estimación de los parámetros de los modelos de mixtura. Entre ellos se incluyen métodos gráficos, el método de momentos, enfoques basados en distancia mínima, máxima verosimilitud y enfoques bayesianos. La razón principal de la enorme literatura sobre la metodología de estimación para los modelos de mixtura se puede deber a que las fórmulas explícitas para las estimaciones de parámetros generalmente no están disponibles. Por ejemplo, en el caso de mixturas normales, el estimador de máxima verosimilitud (MLE) para las proporciones de mixtura, las medias y las matrices de covarianza

de las componentes no tiene una solución analítica explícita (McLachlan y Peel, 2000, Sección 1.13). Estos MLE deben obtenerse mediante métodos iterativos (McLachlan y Krishnan, 2008, Sección 1).

Sin embargo, como se verá a continuación, el cálculo de los MLE es sencillo utilizando el algoritmo de Esperanza-Maximización (EM) propuesto por Dempster *et al.* (1977). El algoritmo EM es un enfoque general para la estimación de máxima verosimilitud cuando los datos pueden considerarse como la realización de observaciones multivariantes $(\mathbf{y}_i, \mathbf{z}_i)$ para $i = 1, \ldots, n$, donde los $\mathbf{y}_i$ son observados y los $\mathbf{z}_i$ son variables latentes, no observadas. Desde su introducción, la máxima verosimilitud (ML) se ha consolidado como el enfoque más utilizado para el ajuste de modelos de mixtura. Por ello, esta sección se centra en la aplicación del algoritmo EM para calcular los estimadores de máxima verosimilitud en modelos de mixtura paramétricos. Se abordará primero el caso general de distribuciones arbitrarias para las componentes, para luego especializar los resultados en el caso de mixturas gaussianas. Antes de entrar en la estimación en modelos de mixtura finita, se introducirá brevemente el concepto general de estimación por máxima verosimilitud.

La estimación de máxima verosimilitud (ML) es un enfoque fundamental para estimar los parámetros de un modelo estadístico a partir de datos observados. En el caso de los modelos de mixtura, el objetivo es estimar un vector de parámetros $\boldsymbol{\Psi}$, que define la distribución de los datos.

Dado un conjunto de observaciones $\mathbf{y}_1, \ldots, \mathbf{y}_n$, se supone que cada una de ellas sigue una distribución con función de densidad $f(\cdot; \boldsymbol{\Psi})$, donde $\boldsymbol{\Psi}$ es el vector de parámetros a estimar. La función de verosimilitud para $\boldsymbol{\Psi}$ se define como:

$$L(\boldsymbol{\Psi}) = \mathbb{P}(\mathbf{y}_1, \ldots, \mathbf{y}_n \mid \boldsymbol{\Psi}) = \prod_{j=1}^{n} f(\mathbf{y}_j; \boldsymbol{\Psi}) = \prod_{j=1}^{n} \Big(\sum_{i=1}^{g} \pi_i f_i(\mathbf{y}_j; \boldsymbol{\theta}_i) \Big), \tag{1.11}$$

lo que representa la probabilidad conjunta de observar los datos $\mathbf{y}_1, \ldots, \mathbf{y}_n$ bajo un valor específico de los parámetros del modelo $\boldsymbol{\Psi}$. Para encontrar la estimación de máxima verosimilitud $\widehat{\boldsymbol{\Psi}}$, se busca el valor de $\boldsymbol{\Psi}$ que maximiza la función (1.11).

En la práctica, en lugar de maximizar directamente la función de verosimilitud, es más conveniente trabajar con su logaritmo, ya que es equivalente y transforma el producto en una suma, lo que facilita el cálculo de derivadas. Así, la función de log-verosimilitud correspondiente es

$$\ell(\boldsymbol{\Psi}) = \log L(\boldsymbol{\Psi}) = \log\Big(\prod_{j=1}^{n} f(\mathbf{y}_j; \boldsymbol{\Psi}) \Big) = \sum_{j=1}^{n} \log f(\mathbf{y}_j; \boldsymbol{\Psi}) = \sum_{j=1}^{n} \log\Big(\sum_{i=1}^{g} \pi_i f_i(\mathbf{y}_j; \boldsymbol{\theta}_i) \Big). \tag{1.12}$$

De esta forma, el estimador de máxima verosimilitud (MLE) $\widehat{\boldsymbol{\Psi}}$ del vector de parámetros $\boldsymbol{\Psi}$ se

define como el valor que maximiza la función de verosimilitud (1.11) o, de forma equivalente, su logaritmo (1.12) dentro del espacio de parámetros. En términos matemáticos, esto significa que es una raíz de la ecuación de verosimilitud $\partial L(\boldsymbol{\Psi})/\partial \boldsymbol{\Psi} = 0$ o, equivalentemente, de la ecuación de log-verosimilitud $\partial \ell(\boldsymbol{\Psi})/\partial \boldsymbol{\Psi} = 0$, que corresponde a un máximo local finito en el espacio de parámetros. La solución obtenida proporciona la estimación $\widehat{\boldsymbol{\Psi}}$.

No obstante, la función de log-verosimilitud dada en la Ecuación (1.12) resulta complicada de maximizar directamente, incluso mediante métodos numéricos (véase McLachlan y Peel, 2000, Sección 2.8.1). Debido a esta dificultad, el ajuste de modelos de mixtura se aborda generalmente reformulando el problema dentro del marco del algoritmo de Esperanza-Maximización (EM), donde se trata como un problema de datos incompletos.

En la Sección 1.2, se introdujo el vector de etiquetas de componentes $\mathbf{Z}_j$, que consiste en variables binarias indicadoras que determinan de qué componente proviene cada vector aleatorio de características $\mathbf{Y}_j$ en el modelo de mixtura (1.1). La formulación del modelo de mixtura en términos de $\mathbf{Y}_j$ y $\mathbf{Z}_j$ es especialmente útil, incluso si en algunos casos no siempre es apropiado interpretarlo de forma literal, ya que permite calcular el estimador de máxima verosimilitud de la distribución de la mixtura mediante una aplicación directa del algoritmo EM, al permitir tratarlo como un problema de datos incompletos. Para más información acerca de los problemas de datos incompletos, consultar Rubin (1976).

Como ya se ha mencionado anteriormente, el objetivo es la estimación de las distribuciones de mixtura basándose en los datos observados $\mathbf{y}_1, \ldots, \mathbf{y}_n$, que corresponden a una muestra aleatoria extraída de la densidad de mixtura (1.1). Así, los datos $\mathbf{y}_1, \ldots, \mathbf{y}_n$ representan valores observados de n vectores aleatorios independientes e idénticamente distribuidos (i.i.d.) $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ con densidad común f. En el marco del algoritmo EM, los vectores de características $\mathbf{y}_1, \ldots, \mathbf{y}_n$ se consideran incompletos, ya que los vectores de etiquetas de componentes $\mathbf{z}_1, \ldots, \mathbf{z}_n$ asociados no están disponibles. El vector de datos completos se define como

$$\mathbf{y}_c = \left(\mathbf{y}^\top, \mathbf{z}^\top\right)^\top,$$

donde

$$\mathbf{y} = \left(\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top\right)^\top$$

es el vector de datos observados o de datos incompletos y donde

$$\mathbf{z} = \left(\mathbf{z}_1^\top, \ldots, \mathbf{z}_n^\top\right)^\top$$

es el vector de datos no observados de etiquetas de componentes. Estos vectores $\mathbf{z}_1, \ldots, \mathbf{z}_n$ se consideran observaciones de los vectores aleatorios $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$, los cuales, bajo la suposición de

independencia de los datos, siguen la distribución:

$$\mathbf{Z}_1, \ldots, \mathbf{Z}_n \overset{\text{i.i.d.}}{\sim} \text{Mult}_g(1, \boldsymbol{\pi}).$$

Esta suposición garantiza que la distribución conjunta del vector de datos completos $\mathbf{Y}_c = (\mathbf{Y}, \mathbf{Z})$ determina de forma natural la distribución del vector de datos observados $\mathbf{Y}$, obtenida al marginalizar sobre las etiquetas Z. Es decir, si la densidad de una observación completa $\mathbf{y}_c = (\mathbf{y}, \mathbf{z})$ es $f_{\mathbf{y}_c}(\mathbf{y}, \mathbf{z}; \boldsymbol{\Psi})$, entonces la densidad de una observación incompleta es

$$f(\mathbf{y}; \boldsymbol{\Psi}) = \int f_{\mathbf{y},\mathbf{z}}(\mathbf{y}, \mathbf{z}; \boldsymbol{\Psi}) dz.$$

Así, al modelar $(\mathbf{Y}, \mathbf{Z})$ conjuntamente, se induce la distribución de mixtura de $\mathbf{Y}$, lo que justifica el uso de este enfoque de datos completos en la estimación.

La función de verosimilitud de los datos completos $(\mathbf{y}, \mathbf{z})$ para $\boldsymbol{\Psi}$, $L_c(\boldsymbol{\Psi})$, viene dada por

$$\begin{aligned} L_c(\boldsymbol{\Psi}) &= \mathbb{P}(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\Psi}) \\ &= \prod_{j=1}^{n} f_{\mathbf{y},\mathbf{z}}(\mathbf{y}_j, \mathbf{z}_j; \boldsymbol{\Psi}) \\ &= \prod_{j=1}^{n} f_{\mathbf{z}}(\mathbf{z}_j) f_{\mathbf{y}|\mathbf{z}}(\mathbf{y}_j \mid \mathbf{z}_j; \boldsymbol{\Psi}) \\ &= \prod_{j=1}^{n} \mathbb{P}(\mathbf{z}_j) f_{\mathbf{y}|\mathbf{z}}(\mathbf{y}_j \mid \mathbf{z}_j; \boldsymbol{\Psi}). \end{aligned} \tag{1.13}$$

La tercera igualdad se obtiene aplicando la fórmula de la densidad condicional, que describe cómo la densidad de probabilidad de una variable aleatoria continua Y dada $X = x$ se obtiene dividiendo la función de densidad conjunta de X e Y $f_{X,Y}(x, y)$ entre la densidad marginal de X $f_X(x)$, siempre que $f_X(x) > 0$:

$$f_{Y|X}(y \mid x) = \frac{f_{X,Y}(x, y)}{f_X(x)}.$$

Además, como los $\mathbf{Z}_j$ son i.i.d de acuerdo con una distribución multinomial con probabilidades $\pi_1, \ldots, \pi_g$, se tiene que

$$\mathbb{P}(\mathbf{z}_j) = \prod_{i=1}^{g} \pi_i^{z_{ij}},$$

y

$$f_{\mathbf{y}|\mathbf{z}}(\mathbf{y}_j \mid \mathbf{z}_j; \boldsymbol{\Psi}) = \prod_{i=1}^{g} f_i(\mathbf{y}_j; \boldsymbol{\theta}_i)^{z_{ij}},$$

ya que si $z_{ji} = 1$, es decir, si la observación $\mathbf{y}_j$ pertenece a la i-ésima componente, entonces

la densidad de $\mathbf{y}_j$ viene dada por $f_i(\mathbf{y}_j;\boldsymbol{\theta}_i)$. Por lo tanto, la función de log-verosimilitud de los datos completos para $\boldsymbol{\Psi}$, $\log L_c(\boldsymbol{\Psi})$, se expresa como

$$\ell_c(\boldsymbol{\Psi}) = \log L_c\left(\boldsymbol{\Psi}\right) = \sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}\Big(\log \pi_i + \log f_i\left(\mathbf{y}_j;\boldsymbol{\theta}_i\right)\Big). \tag{1.14}$$

La verosimilitud de los datos observados (incompletos) $L(\boldsymbol{\Psi})$, se puede obtener integrando los datos no observados z de la verosimilitud de los datos completos:

$$L(\boldsymbol{\Psi}) = \int L_c(\boldsymbol{\Psi})\, dz.$$

A continuación se mostrará cómo el algoritmo EM explota la simplificación que supone trabajar con la distribución conjunta de $\mathbf{Y}_j$ y $\mathbf{Z}_j$ para calcular los MLEs a partir de los datos observados $\mathbf{y}_j$. En este algoritmo, se utiliza la función de verosimilitud del vector de datos completos $\mathbf{y}_c$ y se supera el problema de que $\mathbf{z}_j$ sea desconocido iterando sobre la esperanza condicional de la log-verosimilitud de los datos completos, dados los datos observados $\mathbf{y}$, utilizando la estimación actual de los parámetros desconocidos.

Si el vector de datos completos $\mathbf{y}_c$ estuviera disponible, entonces la estimación de la distribución de la mixtura sería más sencilla que basándose únicamente en los datos observados $\mathbf{y}$, ya que cada densidad de componente $f_i(\mathbf{y})$ podría estimarse directamente a partir de los datos que se sabe que provienen de ella, es decir, a partir de aquellos datos $\mathbf{y}_j$ para los cuales $z_{ij} = (\mathbf{z}_j)_i = 1$. Esto sería una tarea trivial si, por ejemplo, se asumiera que las densidades de las componentes son normales multivariadas. En tal escenario, los únicos parámetros adicionales a estimar serían las proporciones de mixtura π_i, que en un diseño de muestreo basado en clasificación podrían obtenerse directamente como la proporción de datos asignados a cada componente:

$$\widehat{\pi}_i = \frac{1}{n}\sum_{j=1}^{n} z_{ij} \quad (i = 1, \ldots, g).$$

Para más detalles sobre este enfoque de análisis discriminante se recomienda consultar el Capítulo 2 de McLachlan (1992).

1.3.1. Algoritmo Esperanza-Maximización

El algoritmo Esperanza-Maximización, propuesto por Dempster *et al.* (1977), es un procedimiento iterativo que se aplica a este problema tratando los z_{ij} como datos faltantes. Procede iterativamente en dos pasos: E (para la esperanza) y M (para la maximización).

La adición de los datos no observados al problema (los $\mathbf{z}_j$) es manejada por el paso E, que toma la esperanza condicional de la verosimilitud logarítmica de los datos completos, $\ell_c(\mathbf{\Psi})$ definida en (1.14), dados los datos observados $\mathbf{y}$, utilizando el ajuste actual para $\mathbf{\Psi}$. Sea $\mathbf{\Psi}^{(0)}$ un valor establecido inicialmente por el usuario para $\mathbf{\Psi}$. Entonces, en la primera iteración del algoritmo EM, el paso E requiere el cálculo de la esperanza condicional de $\ell_c(\mathbf{\Psi})$ dado $\mathbf{y}$, usando $\mathbf{\Psi}^{(0)}$ para $\mathbf{\Psi}$, que se puede escribir como

$$Q\left(\mathbf{\Psi};\mathbf{\Psi}^{(0)}\right)=\mathbb{E}_{\mathbf{\Psi}^{(0)}}\Big[\ell_c(\mathbf{\Psi})\mid y\Big]=\mathbb{E}_{\mathbf{\Psi}^{(0)}}\Big[\log L_c(\mathbf{\Psi})\mid y\Big].$$

El operador de esperanza $\mathbb{E}$ tiene el subíndice $\mathbf{\Psi}^{(0)}$ para transmitir explícitamente que esta esperanza se está calculando utilizando $\mathbf{\Psi}^{(0)}$ para $\mathbf{\Psi}$. De igual manera, en la iteración $(k+1)$, el paso E requiere el cálculo de $Q\left(\mathbf{\Psi};\mathbf{\Psi}^{(k)}\right)$, donde $\mathbf{\Psi}^{(k)}$ es el valor de $\mathbf{\Psi}$ después de la k-ésima iteración del algoritmo EM. Como la verosimilitud logarítmica de los datos completos, $\ell_c(\mathbf{\Psi})$, es lineal en los datos no observados z_{ij}, la linealidad de la esperanza permite que, en el paso E de la iteración $(k+1)$, el cálculo de la esperanza condicional de $\ell_c(\mathbf{\Psi})$ dados los datos observados $\mathbf{y}$ se reduzca al cálculo de la esperanza condicional de las variables latentes Z_{ij}, donde Z_{ij} es la variable aleatoria correspondiente a z_{ij}. Ahora

$$\mathbb{E}_{\mathbf{\Psi}^{(k)}}(Z_{ij}\mid\mathbf{y})=\mathbb{P}_{\mathbf{\Psi}^{(k)}}(Z_{ij}=1\mid\mathbf{y})=\tau_i\left(\mathbf{y}_j;\mathbf{\Psi}^{(k)}\right), \tag{1.15}$$

donde, de acuerdo con la Ecuación (1.10),

$$\tau_i\left(\mathbf{y}_j;\mathbf{\Psi}^{(k)}\right)=\frac{\pi_i^{(k)}f_i\left(\mathbf{y}_j;\boldsymbol{\theta}_i^{(k)}\right)}{f\left(\mathbf{y}_j;\mathbf{\Psi}^{(k)}\right)}=\frac{\pi_i^{(k)}f_i\left(\mathbf{y}_j;\boldsymbol{\theta}_i^{(k)}\right)}{\sum_{h=1}^{g}\pi_h^{(k)}f_h\left(\mathbf{y}_j;\boldsymbol{\theta}_h^{(k)}\right)} \tag{1.16}$$

para $i=1,\ldots,g;j=1,\ldots,n$, donde la cantidad $\tau_i\left(\mathbf{y}_j;\mathbf{\Psi}^{(k)}\right)$ es la probabilidad a posteriori de que el j-ésimo miembro de la muestra con el valor observado $\mathbf{y}_j$ pertenezca a la i-ésima componente de la mixtura. Así, usando (1.15) y (1.16), al tomar la esperanza condicional de (1.14) tenemos que

$$\begin{aligned}
Q\left(\mathbf{\Psi};\mathbf{\Psi}^{(k)}\right)&=\mathbb{E}_{\mathbf{\Psi}^{(k)}}\Big[\ell_c(\mathbf{\Psi})\mid\mathbf{y}\Big]\\
&=\mathbb{E}_{\mathbf{\Psi}^{(k)}}\Big[\sum_{i=1}^{g}\sum_{j=1}^{n}z_{ij}\Big(\log\pi_i+\log f_i(\mathbf{y}_j;\boldsymbol{\theta}_i)\Big)\mid\mathbf{y}\Big]\\
&=\sum_{i=1}^{g}\sum_{j=1}^{n}\mathbb{E}_{\mathbf{\Psi}^{(k)}}\Big[z_{ij}\mid y\Big]\Big(\log\pi_i+\log f_i(\mathbf{y}_j;\boldsymbol{\theta}_i)\Big)\\
&=\sum_{i=1}^{g}\sum_{j=1}^{n}\tau_i\left(\mathbf{y}_j;\mathbf{\Psi}^{(k)}\right)\Big(\log\pi_i+\log f_i(\mathbf{y}_j;\boldsymbol{\theta}_i)\Big).
\end{aligned} \tag{1.17}$$

Por tanto, en esencia, el paso E reemplaza en $\ell_c(\boldsymbol{\Psi})$ las variables de etiquetas de componentes desconocidas z_{ij} por sus esperanzas condicionales actuales, dadas por las probabilidades a posteriori de pertenencia a un componente para los datos observados $\mathbf{y}_j$, $\tau_i\left(\mathbf{y}_j;\boldsymbol{\Psi}^{(k)}\right)$.

El paso M en la iteración $(k+1)$ requiere la maximización global de $Q\left(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(k)}\right)$ con respecto a $\boldsymbol{\Psi}$ sobre el espacio de parámetros Ω para dar la estimación actualizada $\boldsymbol{\Psi}^{(k+1)}$. Es decir,

$$\boldsymbol{\Psi}^{(k+1)} = \arg\max_{\boldsymbol{\Psi}} Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(k)}).$$

Las estimaciones actualizadas ${\pi_i}^{(k+1)}$ de las proporciones de mixtura π_i en el paso M se calculan independientemente de la estimación actualizada de los parámetros asociados a las componentes de la mixtura $\boldsymbol{\xi}^{(k+1)}$, donde el vector de parámetros $\boldsymbol{\xi}$ contiene los parámetros desconocidos de las componentes de la mixtura $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_g$.

Como se mencionó anteriormente, si z_{ij} fuera observable, entonces el MLE de las proporciones de mixtura π_i estaría dado simplemente por

$$\widehat{\pi}_i = \sum_{j=1}^{n} \frac{z_{ij}}{n} \quad (i=1,\ldots,g). \tag{1.18}$$

Como el paso E simplemente implica reemplazar cada z_{ij} por su esperanza condicional actual $\tau_i\left(\mathbf{y}_j;\boldsymbol{\Psi}^{(k)}\right)$ en la verosimilitud logarítmica de los datos completos, la estimación actualizada de π_i se obtiene reemplazando cada z_{ij} en (1.18) por $\tau_i\left(\mathbf{y}_j;\boldsymbol{\Psi}^{(k)}\right)$ para dar

$${\pi_i}^{(k+1)} = \sum_{j=1}^{n} \frac{\tau_i\left(\mathbf{y}_j;\boldsymbol{\Psi}^{(k)}\right)}{n} \quad (i=1,\ldots,g)\,. \tag{1.19}$$

Así, al formar la estimación de π_i en la iteración $(k+1)$, cada observación $\mathbf{y}_j$ contribuye con un valor igual a su probabilidad a posteriori (evaluada en la iteración actual) de pertenecer a la i-ésima componente del modelo de mixtura. Por lo tanto, esta solución del algoritmo EM tiene una interpretación intuitivamente atractiva.

Con respecto a la actualización de $\boldsymbol{\xi}$ en el paso M de la iteración $(k+1)$, los nuevos parámetros $\boldsymbol{\xi}^{(k+1)}$ deben maximizar $Q\left(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(k)}\right)$ con respecto a $\boldsymbol{\xi}$. Por tanto, teniendo en cuenta la Ecuación (1.17), $\boldsymbol{\xi}^{(k+1)}$ se obtiene como una raíz de

$$\sum_{i=1}^{g}\sum_{j=1}^{n} \tau_i\left(\mathbf{y}_j;\boldsymbol{\Psi}^{(k)}\right) \frac{\partial \log f_i\left(\mathbf{y}_j;\boldsymbol{\theta}_i\right)}{\partial \boldsymbol{\xi}} = 0 \tag{1.20}$$

que corresponde a un máximo de $Q\left(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}\right)$.

Una característica interesante del algoritmo EM es que la solución de la Ecuación (1.20) suele existir en forma cerrada, como se demostrará para el modelo de mixtura gaussiana en la Sección 1.3.3.

Los pasos E y M se alternan repetidamente hasta que la diferencia

$$L(\boldsymbol{\Psi}^{(k+1)}) - L(\boldsymbol{\Psi}^{(k)})$$

entre la log-verosimilitud de los datos incompletos (observados) de dos iteraciones consecutivas es inferior a una cantidad "pequeña" que puede ser prefijada, lo que indica la convergencia de la secuencia de valores de verosimilitud $\{L(\boldsymbol{\Psi}^{(k)})\}$, es decir, la convergencia del algoritmo EM a un valor L^*. Un umbral típico en la práctica para este criterio es de 10^{-5} (Bouveyron *et al.*, 2019, Sección 2.3). El paquete `mclust` (Scrucca *et al.*, 2016) de R utiliza este criterio por defecto.

Dempster *et al.* (1977) demostraron que la función de verosimilitud de los datos incompletos $L(\boldsymbol{\Psi})$ no disminuye después de una iteración del algoritmo EM, es decir,

$$L(\boldsymbol{\Psi}^{(k+1)}) \geq L(\boldsymbol{\Psi}^{(k)}),$$

$k = 0, 1, 2, \ldots$ Por lo tanto, la convergencia debe alcanzarse en una secuencia de valores de verosimilitud acotada superiormente. Dempster *et al.* (1977) también mostraron que si se cumple la condición de que $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$ es continua en ambas variables $\boldsymbol{\Psi}$ y $\boldsymbol{\Psi}^{(k)}$ $(k = 0, 1, 2, \ldots)$, entonces L^* es un máximo local de $L(\boldsymbol{\Psi})$, siempre que la secuencia no quede atrapada en un punto de silla. De esta forma, al alcanzar la convergencia, $\boldsymbol{\Psi}^{(k)}$, correspondiente a la última estimación calculada, representa la estimación de máxima verosimilitud de los parámetros del modelo.

En resumen, el algoritmo EM para mixturas finitas consta de los siguientes pasos:

- **Inicialización:** establecer $k = 0$ y elegir valores iniciales para los parámetros, $\boldsymbol{\Psi}^{(0)}$.

- **Paso E:** estimar la pertenencia a las componentes latentes:

$$z_{ij}^{(k)} = \mathbb{P}(z_{ij} = 1 \mid \mathbf{y}, \boldsymbol{\Psi}^{(k)}) = \tau_i\left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)}\right) = \frac{\pi_i^{(k)} f_i(\mathbf{y}_j; \boldsymbol{\theta}_i^{(k)})}{\sum_{h=1}^{g} \pi_h^{(k)} f_h(\mathbf{y}_j; \boldsymbol{\theta}_h^{(k)})}.$$

- **Paso M:** obtener las estimaciones actualizadas de los parámetros:

$$\boldsymbol{\Psi}^{(k+1)} = \arg\max_{\boldsymbol{\Psi}} Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}).$$

Para modelos de mixtura finita, en particular:

$$\pi_i^{(k+1)} = \sum_{j=1}^{n} \frac{z_{ij}^{(k)}}{n}$$

y se resuelve

$$\sum_{i=1}^{g} \sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right) \frac{\partial \log f_i \left(\mathbf{y}_j; \boldsymbol{\theta}_i \right)}{\partial \boldsymbol{\xi}} = 0.$$

- **Criterio de convergencia:** si los criterios de convergencia no se cumplen, establecer $k = k + 1$ y repetir un nuevo paso E seguido de un paso M.

La forma de asignar valores iniciales a los parámetros, es decir, la elección de $\boldsymbol{\Psi}^0$, se discute más adelante en la Sección 1.3.2.

Las propiedades del algoritmo EM han sido objeto de numerosos estudios; para una revisión detallada véase McLachlan y Krishnan (2008). Algunas de las principales ventajas de este algoritmo son las siguientes. En primer lugar, a menos que se alcance un punto estacionario de la log-verosimilitud, cada iteración de EM incrementa la log-verosimilitud. Por otro lado, en muchos casos de interés práctico, los pasos E y M son más manejables en términos de implementación que la maximización directa de la log-verosimilitud, y el coste por iteración suele ser relativamente bajo. Sin embargo, el algoritmo también presenta algunas desventajas. Las estimaciones de los parámetros pueden depender significativamente de los valores iniciales y de los criterios de convergencia. Además, la convergencia puede ser difícil de evaluar: no solo la tasa de convergencia asintótica puede ser lenta, sino que también el progreso puede ser lento incluso cuando el valor actual está lejos de un punto estacionario. Cabe señalar que, aunque el EM clásico no es necesariamente lento en términos absolutos, especialmente en problemas de dimensión moderada, existen alternativas optimizadas que aceleran su convergencia, como los algoritmos EM acelerados (vía métodos quasi-Newton, Louis o híbridos) (McLachlan y Peel, 2000, Sección 2.17). Estas alternativas pueden ser preferibles en escenarios con grandes volúmenes de datos o cuando se requieren soluciones en tiempo real.

1.3.2. Inicialización del algoritmo EM

La eficiencia del algoritmo EM puede depender en gran medida del punto de inicio, ya que la superficie de verosimilitud de una mixtura finita tiende a tener múltiples máximos locales. Por lo tanto, la inicialización del algoritmo EM suele ser crucial, aunque no se ha encontrado un método que supere consistentemente a los demás. No obstante, el algoritmo EM generalmente produce resultados razonables cuando se inicia con valores iniciales adecuados (Wu, 1983). Existen dos enfoques para iniciar el algoritmo EM.

El primer enfoque consiste en inicializar el algoritmo EM asignando valores iniciales para los parámetros $\boldsymbol{\Psi}^{(0)}$ dentro de la región factible, o, alternativamente, asignando probabilidades iniciales de pertenencia a las componentes para cada dato. Esta elección de valores iniciales se puede realizar mediante distintas estrategias. Una forma de inicializar estos valores es aleatoriamente. Dado que la estrategia de inicios aleatorios tiene una probabilidad considerable de no proporcionar valores iniciales óptimos, una recomendación común es ejecutar el algoritmo EM con varios inicios aleatorios y elegir la solución que alcance la mayor verosimilitud logarítmica. Una variación de este método es la estrategia emEM propuesta por Biernacki *et al.* (2003). Este procedimiento realiza varias ejecuciones cortas del algoritmo EM (reflejadas en el nombre por "em", en minúsculas), cada una con diferentes valores iniciales aleatorios para los parámetros, que se detienen tras un número fijo y reducido de iteraciones, sin esperar a la convergencia completa. Luego, se selecciona la solución con la mayor verosimilitud logarítmica para iniciar una ejecución prolongada y completa del EM (representada por "EM", en mayúsculas), que continua hasta cumplir los criterios de convergencia habituales. El paquete `Rmixmod` (Rémi Lebret *et al.*, 2015) de R para mixturas finitas de distribuciones gaussianas utiliza esta estrategia por defecto. Este método es más robusto que una simple inicialización aleatoria, pero es computacionalmente costoso y sufre, aunque en menor medida, de los mismos problemas que los inicios aleatorios, ya que sigue dependiendo de los valores iniciales.

En el segundo enfoque, en lugar de fijar directamente los parámetros, se selecciona una partición inicial de las observaciones correspondientes a las componentes de la mixtura, estableciendo $z_{ij}^{(0)}$ en 1 si la i-ésima observación se asigna a la j-ésima componente y 0 en caso contrario. Esta partición inicial se puede obtener mediante algún algoritmo de agrupamiento, como k-medias o clustering jerárquico, véase Saxena *et al.* (2017). En este caso, la clasificación final se usa para iniciar el algoritmo EM desde el paso M. No obstante, el uso de estos algoritmos de particionamiento para inicializar EM tiene algunas desventajas. Por ejemplo, algunos de estos métodos presentan sus propios problemas de inicialización y pueden tender a imponer artificialmente ciertas formas o patrones en los clústeres. La partición inicial también se puede obtener aleatoriamente, sin embargo, los valores aleatorios de los parámetros tienden a proporcionar mejores valores iniciales que las particiones aleatorias de las observaciones (Bouveyron *et al.*, 2019, Sección 2.4). Biernacki *et al.* (2003) también propusieron dos algoritmos de inicialización basados en este enfoque: el algoritmo *Classification EM* (CEM) y el algoritmo *Stochastic EM* (SEM). Estos algoritmos se basan en variantes de EM, que se ejecutan un número determinado de iteraciones, generalmente hasta que se estabilizan, obteniendo una partición inicial de las observaciones. En CEM, cada observación se asigna determinísticamente a la componente con mayor probabilidad posterior, generando así una partición completa de los datos en cada iteración. En el contexto del agrupamiento (clustering), el algoritmo CEM una alternativa eficaz a la estrategia *emEM*, ya que presenta una convergencia rápida y puede ejecutarse múltiples veces con un coste computacional

reducido (Biernacki *et al.*, 2003). Por su parte, SEM introduce aleatoriedad en la asignación de las observaciones a las componentes, lo que puede ayudar a evitar ciertos óptimos locales, muestreando las etiquetas latentes según las probabilidades posteriores; esta estrategia también produce una partición inicial en cada iteración. Otra estrategia ampliamente utilizada, especialmente en modelos de mixtura gaussiana, basada en este enfoque es la inicialización mediante clustering jerárquico basado en modelos (HMBC, por sus siglas en inglés), propuesta por Banfield y Raftery (1993). El clustering jerárquico basado en modelos (HMBC) fusiona sucesivamente los pares de clústeres cuya unión produce el mayor incremento en la verosimilitud de clasificación. En ausencia de información previa, el proceso comienza considerando cada observación como un clúster individual y se repite hasta que todos los datos quedan agrupados en un único clúster. En cada paso del algoritmo, para cada partición, los parámetros del modelo se estiman una sola vez, condicionados a la partición actual, sin necesidad de realizar iteraciones. Una de sus principales ventajas es su eficiencia computacional, ya que con una sola pasada sobre los datos se obtienen particiones iniciales para distintos modelos y números de clústeres. Proporciona una asignación inicial razonable de clústeres, por lo que se utiliza frecuentemente para inicializar métodos más precisos, como el EM. En el paquete `mclust` (Scrucca *et al.*, 2016) de R para mixturas finitas de distribuciones gaussianas, el algoritmo EM se inicializa ejecutando un clustering jerárquico basado en modelos usando un modelo sin restricciones (heterocedástico, modelo VVV en la Tabla 1.1). Este procedimiento genera una partición para cada número posible de componentes, las cuales se utilizan como puntos iniciales para el algoritmo EM.

1.3.3. Algortimo EM para modelos de mixturas gaussianas

Tal como se mencionó en la Sección 1.1, una suposición común en la práctica es tomar las densidades de las componentes como normales. Por esto, ahora se van a especializar los resultados descritos en la Sección 1.3.1 sobre la aplicación del algoritmo EM para el ajuste ML en el caso de una mixtura de componentes gaussianas, definida en la Ecuación (1.7).

En primer lugar, se va a considerar el caso no restringido (heterocedástico, modelo VVV en la Tabla 1.1) donde las matrices de covarianza de las componentes Σ_i pueden ser desiguales; es decir, no se les imponen restricciones, más allá de las habituales (ser simétrica y definida positiva). Con respecto al paso E en la iteración $k+1$, se ha visto en la Sección 1.3.1 que reemplaza las variables de etiquetas de componente desconocidas z_{ij} por sus esperanzas condicionales actuales dadas por las probabilidades a posteriori de pertenencia a componentes de los datos observados $\mathbf{y}_j$, $\tau_i\left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)}\right)$, cuya expresión es ahora

$$\tau_i\left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)}\right) = \frac{\pi_i^{(k)} \phi\left(\mathbf{y}_j; \mu_i^{(k)},\ \Sigma_i^{(k)}\right)}{\sum_{h=1}^{g} \pi_h^{(k)} \phi\left(\mathbf{y}_j; \mu_h^{(k)}, \Sigma_h^{(k)}\right)} \tag{1.21}$$

para $i = 1, \ldots, g; j = 1, \ldots, n$.

En el paso M para componentes normales, la estimación de las medias tiene una solución simple en forma cerrada (Bouveyron *et al.*, 2019, Sección 2.3). La actualización de las medias μ_i de las componentes se obtiene mediante la siguiente expresión:

$$\mu_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right) \mathbf{y}_j}{\sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right)} \tag{1.22}$$

para $i = 1, \ldots, g$. El cálculo de la estimación de las matrices de covarianza depende de la parametrización elegida, entre las definidas en la Sección 1.1.1. En el modelo más general VVV, véase la Tabla 1.1, donde no se imponen restricciones sobre los volúmenes, formas y orientaciones de las matrices de covarianza Σ_i de las componentes de la mixtura, la actualización de la covarianza se expresa como:

$$\Sigma_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right) \left(\mathbf{y}_j - \mu_i^{(k+1)} \right) \left(\mathbf{y}_j - \mu_i^{(k+1)} \right)^\top}{\sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right)} \tag{1.23}$$

para $i = 1, \ldots, g$. La estimación actualizada de la i-ésima proporción de mixtura $\pi_i^{(k+1)}$ es la dada en la Ecuación (1.19).

Desde el punto de vista computacional, es ventajoso expresar la actualización (1.23) de Σ_i directamente en términos de las esperanzas condicionales actuales de los estadísticos suficientes T_{i1}, T_{i2} y T_{i3} para $\boldsymbol{\Psi}$ en el marco de datos completos, dado por

$$T_{i1}^{(k)} = \sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right), \quad \boldsymbol{T}_{i2}^{(k)} = \sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right) \mathbf{y}_j \quad \text{y} \quad T_{i3}^{(k)} = \sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)} \right) \mathbf{y}_j \mathbf{y}_j^\top$$

Se tiene entonces que

$$\Sigma_i^{(k+1)} = \frac{T_{i3}^{(k)} - T_{i1}^{(k)^{-1}} T_{i2}^{(k)} T_{i2}^{(k)^\top}}{T_{i1}^{(k)}} \quad (i = 1, \ldots, g). \tag{1.24}$$

El uso de (1.24) en lugar de (1.23) para actualizar la estimación de la matriz de covarianza de la i-ésima componente proporciona una reducción en el tiempo de CPU de alrededor del 50 % (McLachlan y Peel, 2000, Sección 3.2).

Además, a menudo, en la práctica, las matrices de covarianza Σ_i de las componentes se limitan a ser iguales,

$$\Sigma_i = \Sigma \quad (i = 1, \ldots, g),$$

donde Σ es desconocida. En este caso de componentes normales homocedásticas, correspondiente al modelo EEE, la estimación actualizada de la matriz de covarianza Σ común a todas las componentes viene dada por

$$\Sigma^{(k+1)} = \sum_{i=1}^{g} \frac{T_{i1}^{(k)} \Sigma_i^{(k+1)}}{n},$$

donde $\Sigma_i^{(k+1)}$ está dado por (1.24), y las actualizaciones de π_i y μ_i son iguales al caso heterocedástico (véanse las Ecuaciones (1.21) y (1.22)).

Como se mencionó anteriormente, la estimación de Σ_i depende de la parametrización adoptada para las matrices de covarianza de las componentes de la mixtura gaussiana. Ya se ha presentado el caso general del modelo VVV y el caso homocédastico del modelo EEE. En la Tabla 1.3 se muestra también la estimación actualizada para otros casos simples, donde

$$W_i = \sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)}\right) \left(\mathbf{y}_j - \mu_i^{(k+1)}\right) \left(\mathbf{y}_j - \mu_i^{(k+1)}\right)^\top, \quad W = \sum_{i=1}^{g} W_i \quad \text{y} \quad n_i = \sum_{j=1}^{n} \tau_i \left(\mathbf{y}_j; \boldsymbol{\Psi}^{(k)}\right).$$

Para más detalles sobre el paso M para Σ_i parametrizada por la descomposición en valores propios (1.9), véase Celeux y Govaert (1995). Celeux y Govaert (1995) discuten el paso M para los 14 modelos definidos en la Sección 1.1.1, proporcionando métodos iterativos para los cinco modelos (VEI, VEE, VEV, EVE, VVE) para los cuales el paso M no tiene una forma cerrada. En estos casos, al no existir una solución analítica, es necesario recurrir a métodos numéricos iterativos para actualizar los parámetros en cada iteración del EM. Una ventaja de disponer de una forma cerrada es la mayor eficiencia computacional, ya que permite actualizar los parámetros directamente mediante expresiones explícitas, sin necesidad de recurrir a procedimientos iterativos internos. Esto reduce el tiempo de cómputo por iteración del EM, lo que puede ser especialmente relevante en conjuntos de datos grandes o en problemas de clasificación con muchas componentes.

Modelo	$\boldsymbol{\Sigma_i}$	$\boldsymbol{\Sigma_i^{(k+1)}}$
EII	λI	$\frac{\mathrm{tr}(W)}{pn} I$
VII	$\lambda_i I$	$\frac{\mathrm{tr}(W_i)}{pn_i} I$
EEE	$\lambda D A D^\top$	$\frac{W}{n}$
VVV	$\lambda_i D_i A_i D_i^\top$	$\frac{W_i}{n_i}$

Tabla 1.3: Estimaciones actualizadas de Σ_i en el paso M para cuatro parametrizaciones distintas de las matrices de covarianza de las componentes en un modelo de mixtura gaussiana.

Capítulo 2

Clustering a través de modelos de mixtura

Los modelos de mixtura permiten abordar el problema del clustering desde una perspectiva estadística bien fundamentada, conocida como clustering basado en modelos (*model-based clustering*). Este enfoque asume que los datos provienen de una población heterogénea compuesta por varias subpoblaciones, cada una de las cuales se modela mediante una distribución de probabilidad específica. El tipo de distribución se especifica a priori, mientras que la estructura del modelo (incluido el número de componentes) se determina mediante técnicas de estimación de parámetros y selección de modelos.

A diferencia de otros métodos de agrupamiento, el clustering basado en modelos permite asignaciones probabilísticas, asociando a cada observación una probabilidad de pertenencia a cada grupo, lo que permite representar la incertidumbre en el agrupamiento de los datos, posibilitando una asignación flexible y más realista que el agrupamiento estricto. Además, permite estimar conjuntamente los parámetros de cada componente, seleccionar el modelo óptimo mediante criterios estadísticos como el BIC y obtener la forma de las distribuciones que representan a cada componente.

Entre las principales ventajas del clustering basado en modelos frente a otros métodos de agrupamiento se encuentran su base estadística sólida, que favorece la interpretación rigurosa y la reproducibilidad de los resultados, la posibilidad de seleccionar el modelo de manera objetiva, su adaptabilidad a distintos tipos de datos y contextos de análisis, y la obtención de una descripción detallada de la distribución que siguen los datos dentro de cada grupo.

Disponiendo del modelo de mixtura ajustado con las estimaciones de sus parámetros, es posible utilizarlo no solo para describir la distribución subyacente de los datos, sino también

para asignar las observaciones a los diferentes grupos identificados por el modelo. Una estrategia habitual para realizar esta asignación es el procedimiento de clasificación conocido como Máxima A Posteriori (MAP), el cual permite obtener una partición "dura" de los datos basada en las probabilidades a posteriori estimadas para cada componente del modelo. A continuación, se presenta cómo se aplica este enfoque dentro del marco del clustering basado en modelos.

2.1. Clasificación Máxima A Posteriori

Una vez ajustado un modelo de mixtura finita y estimados sus parámetros, es posible realizar una asignación de las observaciones a los distintos grupos mediante la clasificación de máxima a posteriori (Scrucca *et al.*, 2023, Sección 2.2.4). Este procedimiento consiste en asignar cada observación a la componente del modelo que maximiza su probabilidad posterior condicional, introducida en la Sección 1.2 por la Ecuación (1.10).

Formalmente, dado un conjunto de datos $\mathbf{y}_1, \ldots, \mathbf{y}_n$ y una mixtura de g componentes, se calcula para cada observación $\mathbf{y}_j$ la probabilidad a posteriori estimada de pertenecer a la componente i, definida como:

$$\tau_i(\mathbf{y}_j; \hat{\boldsymbol{\psi}}) = \frac{\hat{\pi}_i f_i(\mathbf{y}_j; \hat{\boldsymbol{\theta}}_i)}{f(\mathbf{y}_j; \hat{\boldsymbol{\psi}})} = \frac{\hat{\pi}_i f_i(\mathbf{y}_j; \hat{\boldsymbol{\theta}}_i)}{\sum_{k=1}^{g} \hat{\pi}_k f_k(\mathbf{y}_j; \hat{\boldsymbol{\theta}}_k)} \quad (i = 1, \ldots, g;\ j = 1, \ldots, n), \tag{2.1}$$

donde $\hat{\pi}_i$ es el peso estimado de la componente i, $f_i(\mathbf{y}_j; \hat{\boldsymbol{\theta}}_i)$ es la densidad de la componente i evaluada en $\mathbf{y}_j$ utilizando los parámetros estimados $\hat{\boldsymbol{\theta}}$ y $f(\mathbf{y}_j; \hat{\boldsymbol{\psi}})$ es la densidad de la mixtura.

De esta forma, se puede realizar un agrupamiento probabilístico de las n observaciones $\mathbf{y}_1, \ldots, \mathbf{y}_n$ en función de sus probabilidades a posteriori ajustadas de pertenencia a cada componente. Para cada observación $\mathbf{y}_j$, las g probabilidades $\tau_1(\mathbf{y}_j; \hat{\boldsymbol{\Psi}}), \ldots, \tau_g(\mathbf{y}_j; \hat{\boldsymbol{\Psi}})$ representan las probabilidades a posteriori estimadas de que dicha observación pertenezca, respectivamente, a la primera, segunda, ..., y g-ésima componente de la mixtura (para $j = 1, \ldots, n$).

Mediante la regla MAP se puede realizar un agrupamiento estricto de los datos asignando cada observación $\mathbf{y}_j$ a la componente de la mixtura a la que pertenece con mayor probabilidad a posteriori. Siguiendo con la interpretación presentada en la Sección 1.2, se estima la etiqueta de componente $\mathbf{z}_j$ de $\mathbf{y}_j$ mediante $\hat{\mathbf{z}}_j$, que será la estimación de máxima a posteriori de $\mathbf{z}_j$, donde $\hat{z}_{ij} = (\hat{\mathbf{z}}_j)_i$ se define como:

$$\hat{z}_{ij} = \begin{cases} 1, & \text{si } i = \arg \max\limits_{k \in \{1,\ldots,g\}} \tau_k(\mathbf{y}_j; \hat{\boldsymbol{\Psi}}) \\ 0, & \text{en otro caso} \end{cases} \quad \text{para } i = 1, \ldots, g;\ j = 1, \ldots, n. \tag{2.2}$$

Así, cada observación $\mathbf{y}_j$ se asigna a uno de los g grupos, $G_1 \ldots, G_g$, de la mixtura de la siguiente forma:

$$\mathbf{y}_j \in G_{i^*} \quad \text{con} \quad i^* = \arg \max_{i \in \{1,\ldots,g\}} \tau_i(\mathbf{y}_j; \hat{\mathbf{\Psi}}).$$

Este procedimiento proporciona una partición estricta (hard clustering) de los datos, donde cada observación pertenece a un único grupo.

Además, este enfoque permite cuantificar la incertidumbre en la asignación de cada dato $\mathbf{y}_j$ al grupo al que se le ha asignado mediante una medida definida como:

$$u_j = 1 - \max_{i \in \{1,\ldots,g\}} \tau_i(\mathbf{y}_j; \hat{\mathbf{\Psi}}),$$

la cual toma valores en el intervalo $[0, 1 - 1/g]$. Valores de u_j cercanos a cero indican una asignación segura, mientras que valores próximos a $1 - 1/g$ reflejan mayor ambigüedad en la pertenencia del dato a dicho grupo, puesto que representan contextos de igual probabilidad para todos los grupos.

En el clustering, existen dos problemas principales: la elección del método de agrupamiento y la determinación del número de grupos. En el enfoque de modelos de mixtura, ambos problemas se pueden reducir a uno solo: la selección del modelo. Esto se debe a que cada combinación del número de clústeres y el modelo de agrupamiento corresponde a un modelo estadístico diferente para los datos. Por lo tanto, el problema se convierte en una tarea de comparar los diferentes modelos posibles. A continuación, se presentan distintos criterios para realizar dicha selección del modelo.

2.2. Elección del número de clústeres y del modelo de clustering

En los modelos de mixtura finita, dos cuestiones fundamentales son la elección del número de componentes que debe incluirse en la mixtura y la elección del modelo de clustering más apropiado. Por ejemplo, en el caso particular de los modelos de mixtura gaussiana (GMM), presentados en la Sección 1.1, además de determinar el número de clústeres es necesario seleccionar una parametrización adecuada para las matrices de covarianza, véase la Tabla 1.1. Ambas cuestiones pueden abordarse conjuntamente como un único problema de selección de modelo, ya que cada combinación de modelo y número de componentes define un modelo estadístico distinto. Para llevar a cabo esta selección, se emplean habitualmente criterios de información, como el criterio de información bayesiano (BIC) o el criterio de verosimilitud completa integrada (ICL). Estos criterios buscan equilibrar el ajuste del modelo a los datos con la complejidad del modelo, penalizando aquellos que tienen demasiados parámetros. De esta manera, se favorecen modelos que

explican adecuadamente los datos sin sobreajustarse a ellos. Alternativamente, también pueden utilizarse contrastes de hipótesis formales para determinar el número de componentes.

Una forma de abordar el problema de la selección de modelos en el contexto del clustering es a través de la selección bayesiana de modelos, utilizando factores de Bayes y probabilidades a posteriori de los modelos (Kass y Raftery, 1995). La idea es que, si se consideran varios modelos estadísticos $M_1, \ldots, M_K$, con probabilidades a priori $\mathbb{P}(M_k)$, $k = 1, \ldots, K$ (que a menudo se suponen iguales), entonces, según el teorema de Bayes, la probabilidad a posteriori del modelo M_k dado un conjunto de datos D es proporcional a la probabilidad de los datos bajo el modelo M_k, multiplicada por la probabilidad a priori del modelo, es decir:

$$\mathbb{P}(M_k \mid D) \propto \mathbb{P}(D \mid M_k)\mathbb{P}(M_k).$$

Cuando los modelos contienen parámetros desconocidos, la probabilidad de los datos bajo el modelo M_k, $\mathbb{P}(D \mid M_k)$, se obtiene aplicando la ley de la probabilidad total, lo que implica integrar (y no maximizar) sobre el espacio de parámetros:

$$\mathbb{P}(D \mid M_k) = \int \mathbb{P}(D \mid \boldsymbol{\Psi}_{M_k}, M_k)\, \mathbb{P}(\boldsymbol{\Psi}_{M_k} \mid M_k)\, d\boldsymbol{\Psi}_{M_k},$$

donde $\mathbb{P}(\boldsymbol{\Psi}_{M_k} \mid M_k)$ es la distribución a priori del vector de parámetros $\boldsymbol{\Psi}_{M_k}$ asociado al modelo M_k. La cantidad $\mathbb{P}(D \mid M_k)$ se conoce como la verosimilitud integrada o verosimilitud marginal del modelo M_k.

Un enfoque bayesiano natural para la selección de modelos es elegir el modelo que sea más probable a posteriori. Si las probabilidades a priori de los modelos, $\mathbb{P}(M_k)$, son iguales, esto equivale a elegir el modelo con la mayor verosimilitud integrada $\mathbb{P}(D \mid M_k)$. Para comparar dos modelos, M_1 y M_2, el factor de Bayes se define como la razón de las dos verosimilitudes integradas, $B_{12} = \mathbb{P}(D \mid M_1)/\mathbb{P}(D \mid M_2)$, con la comparación favoreciendo a M_1 si $B_{12} > 1$, y considerándose una evidencia muy fuerte a favor de M_1 si $B_{12} > 100$ (lo que significa que los datos son al menos 100 veces más probables bajo M_1 que bajo M_2), según una escala propuesta por Jeffreys (1961) que se ha convertido en una convención empírica ampliamente adoptada por su utilidad práctica. A menudo, se toma el valor de $2\log(B_{12})$ en lugar de B_{12}, y en esta escala, redondeando, una evidencia muy fuerte se corresponde a un umbral de 10 (Kass y Raftery, 1995).

La principal dificultad en el uso de los factores de Bayes es la evaluación de la integral que define la verosimilitud integrada $\mathbb{P}(D \mid M_k)$. Para los modelos regulares, la verosimilitud integrada puede aproximarse simplemente mediante el Criterio de Información Bayesiano (BIC), introducido originalmente por Schwarz (1978):

$$2\log \mathbb{P}(D \mid M_k) \approx 2\log \mathbb{P}(D \mid \hat{\boldsymbol{\Psi}}_{M_k}, M_k) - \nu_{M_k}\log(n) = \mathrm{BIC}_{M_k}, \tag{2.3}$$

donde n es el tamaño de la muestra y ν_{M_k} es el número de parámetros independientes a estimar en el modelo M_k. Se puede encontrar la justificación de emplear este criterio en Kass y Raftery (1995). Así, el Criterio de Información Bayesiano (BIC) proporciona una aproximación al factor de Bayes para comparar dos modelos, M_1 y M_2:

$$2 \log B_{12} \approx \mathrm{BIC}_{M_1} - \mathrm{BIC}_{M_2} = \Delta_{12}.$$

Suponiendo que M_2 tiene el valor más pequeño de BIC, la fuerza de la evidencia en su contra puede resumirse cómo se observa en la Tabla 2.1.

Δ_{12}	Evidencia a favor de M_1 sobre M_2
0 - 2	Insuficiente para hacer una afirmación sólida
2 - 6	Positiva
6 - 10	Fuerte
>10	Muy Fuerte

Tabla 2.1: Interpretación de la diferencia en el valor de BIC (Δ_{12}) entre dos modelos M_1 y M_2 (Scrucca *et al.*, 2023, Sección 2.3.1).

Para una revisión del BIC, su derivación, sus propiedades y aplicaciones, se recomienda consultar Neath y Cavanaugh (2012).

Los modelos de mixtura finita, en general, no satisfacen las condiciones de regularidad que sustentan las demostraciones publicadas de la Ecuación (2.3), pero varios resultados sugieren que es apropiada y que tiene un buen rendimiento en el contexto de clustering basado en modelos. Leroux (1992) mostró que basar la selección de modelos en la comparación de valores de BIC no subestima asintóticamente el número de grupos. Keribin (1998) demostró que el BIC es consistente para estimar el número de grupos, bajo el supuesto de que la verosimilitud está acotada. La verosimilitud, sin embargo, no está acotada en general en los modelos de mixtura gaussiana ya que puede crecer infinitamente si una componente de la mixtura colapsa sobre un único punto (lo que ocurre cuando la matriz de covarianza se vuelve singular). No obstante, una restricción muy débil, como un límite inferior muy pequeño sobre el valor propio más pequeño de las matrices de covarianza, es suficiente para garantizar la validez del supuesto. El software principal utilizado en clustering basado en modelos incorpora este tipo de restricciones, por lo que en la práctica el BIC se puede usar con confianza en estos modelos. Además, en una variedad de aplicaciones de clustering basado en modelos, la elección del modelo basada en el BIC ha dado buenos resultados (Campbell *et al.*, 1997; Dasgupta y Raftery, 1998; Fraley y Raftery, 1998; Stanford y Raftery, 2000).

El BIC es un criterio ampliamente adoptado para la selección de modelos en modelos de mixtura finita, tanto para estimación de densidad (Roeder y Wasserman, 1997) como para clustering (Fraley y Raftery, 1998). Para los modelos de mixtura, el BIC para el modelo M_k se define por la Ecuación (2.3), donde los datos D consisten en las observaciones $\mathbf{y}_1, \ldots, \mathbf{y}_n$ que se van a agrupar, $\hat{\boldsymbol{\Psi}}_{M_k}$ es el estimador de máxima verosimilitud (MLE) de los parámetros del modelo, que generalmente se estima mediante el algoritmo EM, y ν_{M_k} es el número de parámetros independientes en M_k. Así, $\mathbb{P}(D \mid \hat{\boldsymbol{\Psi}}_{M_k}, M_k)$ es la función de verosimilitud de la mixtura, definida por la Ecuación (1.11). De esta forma, para los modelos de mixtura, el BIC toma la siguiente forma, penalizando la log-verosimilitud del modelo al introducir un término de penalización que tiene en cuenta la complejidad del modelo:

$$\mathrm{BIC}_{M,G} = 2\ell_{M,G}(\hat{\boldsymbol{\Psi}}) - \nu_{M,G}\log(n),$$

donde $\ell_{M,G}(\hat{\boldsymbol{\Psi}})$ es la log-verosimilitud de la mixtura en el MLE $\hat{\boldsymbol{\Psi}}$ para el modelo M con G componentes, definida en la Ecuación (1.12), n es el tamaño de la muestra, y $\nu_{M,G}$ es el número de parámetros a estimar. El modelo M y el número de componentes G se eligen para maximizar $\mathrm{BIC}_{M,G}$.

Para los modelos de mixtura normal, presentados en la Sección 1.1, los modelos a comparar corresponden a diferentes números de clústeres y diferentes parametrizaciones de las matrices de covarianza (representadas en la Tabla 1.1), o modelos de clustering. Cada combinación de un número de clústeres y un modelo de clustering define un modelo distinto a comparar. Como ya se comentó anteriormente, el número de parámetros que es necesario estimar, denotado por $\nu_{M,G}$, varía en cada una de estas combinaciones. Los valores de $\nu_{M,G}$ para los distintos modelos de mixtura normal, en función del número de clústeres y la estructura de la matriz de covarianza, pueden consultarse en la Tabla 1.2. Típicamente, se consideran modelos con $G = 1, \ldots, G_{\max}$, para alguna elección razonable del número máximo de clústeres, $G_{\max}$. En el paquete `mclust` (Scrucca *et al.*, 2016) de R, el valor predeterminado de $G_{\max}$ es 9, pero en aplicaciones específicas el número apropiado podría ser mucho mayor. Además, de manera predeterminada, el software `mclust` considera los 14 modelos de covarianza definidos en la Tabla 1.1, lo que da lugar a un número predeterminado de $G_{\max} \times 14 = 126$ modelos considerados.

El BIC está diseñado para elegir el número de componentes en un modelo de mixtura, en lugar del número de clústeres en el conjunto de datos per se. La diferencia es sutil pero importante. El clustering basado en modelos se fundamentó inicialmente en la esperanza de que el número de componentes de la mixtura fuera el mismo que el número de clústeres, pero esto no siempre es cierto. Por ejemplo, en modelos de mixtura gaussiana, un clúster puede ser fuertemente no gaussiano y, en sí mismo, ser mejor representado por una mixtura de distribuciones normales. En este caso, el número de componentes de la mixtura sería mayor que el número de clústeres. Por esta razón, se han propuesto otros criterios para la selección de modelos en clustering. Cuando el

interés se centra principalmente en el clustering en lugar de encontrar el mejor modelo de mixtura para ajustar los datos, una solución a este problema es utilizar, en lugar del BIC, el Criterio de Verosimilitud Integrada de Datos Completos (ICL, por sus siglas en inglés) (Biernacki *et al.*, 2000). Esto se basa en la verosimilitud conjunta de los datos $\mathbf{y}$ y el clustering $\mathbf{z}$, es decir, en la función de verosimilitud de los datos completos $\mathbf{y}_c = (\mathbf{y}, \mathbf{z})$, definida por la Ecuación (1.13).

En este enfoque, la probabilidad de los datos completos bajo el modelo M_k, $\mathbb{P}(y, z \mid M_k)$, se calcula mediante la siguiente integral:

$$\mathbb{P}(\mathbf{y}, \mathbf{z} \mid M_k) = \int \mathbb{P}(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\Psi}_{M_k}, M_k)\, \mathbb{P}(\boldsymbol{\Psi}_{M_k} \mid M_k)\, d\boldsymbol{\Psi}_{M_k}.$$

Utilizando una aproximación similar al BIC para la integral se obtiene

$$2 \log \mathbb{P}(\mathbf{y}, \mathbf{z} \mid M_k) \approx 2 \log \mathbb{P}(\mathbf{y}, \mathbf{z} \mid \hat{\boldsymbol{\Psi}}_{M_k}, M_k) - \nu_{M_k} \log(n). \tag{2.4}$$

La Verosimilitud Integrada de Datos Completos (ICL) del modelo M_k se define como el lado derecho de (2.4), donde $\mathbf{z}$ es reemplazado por su estimación de máxima a posteriori (MAP) $\hat{\mathbf{z}}$, formada por las correspondientes estimaciones de máxima a posteriori (MAP) $\hat{\mathbf{z_j}}$ de cada $\mathbf{z}_j$, definidas por (2.2).

Así, el ICL toma la siguiente forma:

$$\text{ICL} = 2 \log \mathbb{P}(\mathbf{y}, \hat{\mathbf{z}} | \hat{\boldsymbol{\Psi}}_{M_k}, M_k) - \nu_{M_k} \log(n).$$

Esta expresión es análoga a la del BIC, pero incorpora explícitamente la asignación más probable de cada observación a un clúster, lo que permite capturar lo que permite capturar la incertidumbre asociada a la clasificación inducida por el modelo.

A partir de esta formulación, el ICL se puede reescribir como una corrección del BIC mediante un término de penalización adicional:

$$\text{ICL} = \text{BIC} - E(M_k),$$

donde $E(M_k)$ es la entropía esperada de la clasificación del modelo M_k (Biernacki *et al.*, 2000):

$$E(M_k) = -\sum_{j=1}^{n} \sum_{i=1}^{g_{M_k}} \tau_i(\mathbf{y}_j; \hat{\boldsymbol{\Psi}}) \log \left(\tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}}) \right). \tag{2.5}$$

Así, el ICL es igual al BIC penalizado por la entropía esperada de la clasificación. La entropía es alta cuando hay gran incertidumbre sobre la clasificación, y es aún más alta cuando todos los $\tau_i(\mathbf{y}_j; \hat{\boldsymbol{\Psi}})$ son iguales (es decir, todos iguales a $1/g_{M_k}$), momento en el que alcanza el valor

$n\log(g_{M_k})$. La entropía es más baja cuando todos los $\tau_i(\mathbf{y}_j;\hat{\boldsymbol{\Psi}})$ son iguales a 0 o 1, es decir, si la clasificación no tiene incertidumbre, en cuyo caso es igual a cero. Por tanto, este término mide la incertidumbre en la asignación de las observaciones a los clústeres: cuanto mayor es la incertidumbre, mayor es la entropía, y por tanto, mayor es la penalización. Como resultado, el ICL tiende a favorecer modelos que produzcan clústeres más claramente separados que el BIC, por lo que en la práctica el ICL suele elegir el mismo número o un número menor de clústeres que el BIC.

Además de los criterios de información mencionados, la elección del número de componentes en un modelo de mixtura para un modelo específico puede llevarse a cabo mediante un contraste de hipótesis de verosimilitud, llamado test de razón de verosimilitud (LRT, por sus siglas en inglés). Se recomienda consultar McLachlan y Rathnayake (2014) para una revisión más profunda en este tipo de contrastes.

Supongamos que se quiere probar la hipótesis nula $G = G_0$ frente a la alternativa $G = G_1$ para algún $G_1 > G_0$, de modo que

$$\begin{cases} H_0: & G = G_0 \\ H_1: & G = G_1 \end{cases}$$

Usualmente, $G_1 = G_0 + 1$, por lo que un procedimiento común es agregar componentes secuencialmente. Sea $\hat{\boldsymbol{\Psi}}_{G_j}$ el MLE de los parámetros del modelo $\boldsymbol{\Psi}$ calculado bajo $H_j : G = G_j$ (para $j = 0, 1$). El estadístico de contraste del test de razón de verosimilitud (LRTS, por sus siglas inglés) puede escribirse como

$$\text{LRTS} = -2\log\left(\frac{L(\hat{\boldsymbol{\Psi}}_{G_0})}{L(\hat{\boldsymbol{\Psi}}_{G_1})}\right) = 2\left(\ell(\hat{\boldsymbol{\Psi}}_{G_1}) - \ell(\hat{\boldsymbol{\Psi}}_{G_0})\right),$$

donde valores grandes del LRTS proporcionan evidencia en contra de la hipótesis nula, ya que sugieren que la diferencia en la verosimilitud entre los dos modelos es lo suficientemente grande como para pensar que el modelo con G_1 componentes ajusta mejor los datos. En ese caso, se rechaza la hipótesis nula H_0, lo que implica que se prefiere el modelo con más componentes. Por el contrario, si el valor del LRTS es pequeño, no hay suficiente evidencia para rechazar H_0, lo que significa que no se justifica aumentar el número de componentes del modelo.

Bajo condiciones de regularidad estándar, la distribución nula del estadístico LRTS sigue una distribución chi-cuadrada. Sin embargo, en los modelos de mixtura, estas condiciones no se cumplen (McLachlan y Peel, 2000, Capítulo 6). Como resultado, para evaluar la significancia del LRT, se recurre con frecuencia a enfoques de remuestreo para obtener un p-valor. McLachlan (1987) propuso usar el bootstrap para obtener la distribución nula del LRTS. El procedimiento bootstrap es el siguiente:

1. Se genera una muestra bootstrap x_b^* simulando desde el modelo ajustado bajo la hipótesis nula con G_0 componentes, es decir, desde la distribución del modelo de mixtura con el vector de parámetros desconocidos reemplazado por los MLE obtenidos de los datos originales bajo H_0;

2. Se calcula el estadístico de contraste LRTS_b^* para la muestra bootstrap x_b^* después de ajustar el modelo de mixtura en esta muestra con G_0 y G_1 componentes;

3. Los pasos 1 y 2 se replican B veces para aproximar la distribución nula bootstrap de LRTS^*.

Así, se puede calcular una aproximación basada en bootstrap del p-valor como

$$\text{p-valor} \approx \frac{1 + \sum_{b=1}^{B} I(\text{LRTS}_b^* \geq \text{LRTS}_{\text{obs}})}{B+1},$$

donde LRTS_{obs} es el estadístico de contraste calculado sobre la muestra original observada, y $I(\cdot)$ denota la función indicadora, que es igual a 1 si su argumento es verdadero y 0 en caso contrario.

Si se requiriera una estimación muy precisa del p-valor, entonces B podría tener que ser muy grande (Efron y Tibshirani, 1993). Sin embargo, por lo general, no hay interés en estimar un p-valor con alta precisión. En el paquete `mclust` (Scrucca *et al.*, 2016) de `R`, el número de remuestreos bootstrap es 999 por defecto. Incluso con un número limitado de réplicas B (por ejemplo, entre 100 y 300), la cantidad de cómputo involucrado sigue siendo considerable, especialmente para valores de G_0 y G_1 que no están próximos entre sí. Sin embargo, como señala Smyth (2000), el proceso puede implementarse de manera fácil y eficiente en hardware de computación paralela, por ejemplo, utilizando B procesadores paralelos (Smyth, 2000).

El objetivo principal del test de razón de verosimilitud (LRT) es evaluar si la adición de una componente mejora significativamente el ajuste del modelo, proporcionando una prueba estadística formal. Por lo tanto, se utiliza comúnmente para comparar modelos anidados y realizar inferencias sobre la necesidad de componentes adicionales. Sin embargo, debido a que requiere remuestreo, su coste computacional es mayor en comparación con el BIC y el ICL.

BIC e ICL son criterios de información ampliamente utilizados para determinar el modelo y el número óptimo de componentes en modelos de mixtura, penalizando su complejidad. En cambio, el LRT se emplea más específicamente para evaluar si la inclusión de componentes adicionales mejora significativamente el modelo ajustado. En ocasiones, el LRT se utiliza en combinación con el BIC y el ICL. Esto se debe a que, mientras que el BIC y el ICL ofrecen criterios de selección basados en el equilibrio entre el ajuste del modelo y su complejidad, el LRT permite contrastar, mediante un test formal de hipótesis, si la inclusión de componentes adicionales proporciona una mejora en el ajuste que sea estadísticamente significativa. Utilizar estos criterios en conjunto permite tomar decisiones más informadas sobre el número óptimo de componentes.

Aunque criterios como BIC e ICL permiten seleccionar un número adecuado de componentes en la mixtura, en el caso de mixturas gaussianas (Sección 1.1) puede ocurrir que varias componentes representen, en realidad, un único clúster no gaussiano. En la siguiente sección se aborda cómo la fusión de componentes puede ayudar a identificar y reagrupar estos casos para mejorar el resultado del clustering.

2.3. Fusión de clústeres no gaussianos en mixturas gaussianas

El clustering basado en modelos de mixturas gaussianas (Sección 1.1) se basa en la idea de una mixtura finita de distribuciones normales multivariantes o gaussianas, donde cada componente de la mixtura corresponde a un grupo o clúster. Sin embargo, a veces algunos de los grupos no tienen una distribución gaussiana. En ese caso, el BIC tenderá a elegir un modelo que represente un grupo no gaussiano mediante una combinación de varias componentes gaussianas. Esto da una buena estimación de la densidad, pero sobrestima el número de grupos. Por otro lado, el ICL tenderá a seleccionar un número de componentes que corresponde al número real de grupos. Pero al hacerlo, normalmente representa un grupo no gaussiano mediante una sola componente gaussiana, lo cual puede no ajustarse bien a los datos.

Una solución ampliamente utilizada para este dilema es mantener la solución del BIC como modelo para la densidad de los datos, pero combinar componentes de mixtura que estén cercanas antes de realizar el clustering. Esto permite lo mejor de ambos mundos: el número correcto de grupos, y un modelo flexible para cada grupo que se ajuste bien a los datos.

Según Hennig (2010), los métodos para fusionar componentes gaussianas suelen seguir los siguientes pasos:

1. Se parte de las componentes estimadas como grupos iniciales.
2. Se identifica el par de grupos con mayor potencial de ser fusionados.
3. Se aplica un criterio de parada para decidir si se deben combinar o si la partición actual debe considerarse definitiva.
4. Si se fusionan, se repite el procedimiento desde el paso 2.

Como base para la mixtura inicial, lo más común es utilizar el modelo seleccionado por el BIC. Los métodos de combinación se distinguen principalmente por la forma en que eligen los pares de componentes a fusionar y por el criterio que aplican para decidir cuándo detener la fusión.

Un enfoque ampliamente utilizado para seleccionar el par más prometedor es basarse en la entropía de la clasificación del modelo (Baudry *et al.*, 2010), definida previamente en (2.5) dentro del contexto de ICL. Sea $\tau_i(\mathbf{y}_j, \hat{\psi})$ la probabilidad a posteriori estimada de que la observación $\mathbf{y}_j$ provenga de la componente $i = 1, \ldots, g$, según una solución con g grupos (ver Ecuación (2.1)). La clasificación del modelo con g grupos tiene entropía:

$$\mathrm{E}(g) = -\sum_{i=1}^{g}\sum_{j=1}^{n} \tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}}) \log\left(\tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right) \geq 0$$

Si la clasificación no tiene incertidumbre (valores 0 o 1), la entropía es 0. Se maximiza cuando todas las probabilidades son $1/G$, es decir, cuando la clasificación no es informativa.

La idea principal es seleccionar en cada etapa el par de grupos cuya combinación minimice el aumento de la entropía. El cálculo es sencillo, ya que la probabilidad a posteriori de un punto en el grupo combinado es la suma de las probabilidades de los grupos fusionados. El proceso de combinación continúa mientras el incremento de entropía sea pequeño, y se detiene cuando este aumento se vuelve significativo. Para determinar el punto de parada, puede utilizarse un gráfico que representa la entropía en función del número de grupos, donde un cambio brusco en la pendiente, lo que comúnmente se denomina un "codo", sugiere un número adecuado de componentes fusionadas (véase Bouveyron *et al.*, 2019, Sección 3.3). Un ejemplo de este procedimiento puede verse en el Capítulo 3.

Si se combinan los grupos i' y i'' de la solución con g grupos, los valores de $\tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})$ permanecen iguales para todos los grupos i excepto para i' e i''. El nuevo grupo $i' \cup i''$ tiene entonces la siguiente probabilidad condicional:

$$\tau_{i'\cup i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}}) = \tau_{i'}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}}) + \tau_{i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}}).$$

Entonces, la nueva entropía es:

$$-\sum_{j=1}^{n}\left[\left(\tau_{i'}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}}) + \tau_{i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right)\log\left(\tau_{i'}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}}) + \tau_{i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right) + \sum_{i\neq i',i''}\tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\log\left(\tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right)\right].$$

Así, los dos grupos i' y i'' que se van a combinar son aquellos que maximizan el criterio:

$$\begin{aligned}&-\sum_{j=1}^{n}\left[\tau_{i'}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\log\left(\tau_{i'}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right) + \tau_{i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\log\left(\tau_{i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right)\right]\\ &+\sum_{j=1}^{n}\left[\left(\tau_{i'\cup i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right)\log\left(\tau_{i'\cup i''}(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})\right)\right]\end{aligned}$$

entre todos los pares de grupos (i', i'').

En el primer paso del procedimiento de combinación, g es el número de componentes seleccionado por el BIC, y $\tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})$ representa la probabilidad a posteriori estimada de que la observación $\mathbf{y}_j$ provenga de la componente i de la mixtura. Sin embargo, una vez que al menos dos componentes han sido fusionadas en un grupo i, de modo que ahora g es menor que el valor seleccionado por el BIC, $\tau_i(\mathbf{y}_j, \hat{\boldsymbol{\Psi}})$ pasa a representar la probabilidad a posteriori estimada de que la observación $\mathbf{y}_j$ pertenezca a una de las componentes fusionadas dentro del grupo i. El proceso de combinación continúa mientras el incremento de entropía sea pequeño, y se detiene cuando este aumento se vuelve significativo, como se mencionó anteriormente.

Hennig (2010) propuso varios métodos adicionales para seleccionar los pares de componentes a combinar, incluyendo la distancia de Bhattacharyya, probabilidades de clasificación errónea estimadas directamente (DEMP), fuerza predictiva, y versiones del método de ridgeline para evitar combinaciones de componentes con densidades no unimodales. También destacó el uso de la prueba de dip para unimodalidad.

ICL selecciona un modelo que puede no ser completamente satisfactorio debido a que representa cada grupo con una única distribución gaussiana, incluso cuando el grupo es no gaussiano. Sin embargo, tiende a ser efectivo al seleccionar el número adecuado de grupos, a diferencia del número adecuado de componentes de la mixtura. Así, en el caso en que BIC seleccione más componentes de la mixtura que ICL, un enfoque híbrido para el clustering sería el siguiente: usar ICL para determinar el número de grupos, usar BIC para seleccionar el modelo y el número de componentes de la mixtura, y finalmente, fusionar las componentes mediante un método de combinación, como el basado en entropía, hasta alcanzar el número de grupos seleccionado por ICL. Este enfoque híbrido combina lo mejor de ambos métodos, aprovechando la capacidad de ICL para elegir el número adecuado de grupos, mientras que BIC se encarga de la selección de las componentes de la mixtura y su ajuste adecuado, proporcionando un modelo de clustering más preciso y robusto.

Capítulo 3

Análisis ilustrativo

En este capítulo se ilustrarán de forma práctica los conceptos presentados a lo largo del trabajo, utilizando como herramienta principal el paquete `mclust` de R.

`mclust` (Scrucca *et al.*, 2016) es un paquete ampliamente utilizado en R (R Core Team, 2022) para realizar tareas de clustering, clasificación y estimación de densidad basadas en modelos de mixtura finita de distribuciones gaussianas (GMMs; véase la Sección 1.1). Proporciona un enfoque integrado para trabajar con estos modelos, incluyendo funciones que combinan inicialización mediante clustering jerárquico basado en modelos (véase la Sección 1.3.2), el algoritmo EM (Expectation-Maximization) para la estimación de los parámetros de la mixtura (véase la Sección 1.3.1), y diversas herramientas para la selección del modelo más adecuado. El paquete permite elegir entre diferentes estructuras de covarianza y aplicar restricciones entre componentes (véase la Sección 1.1.1). Además, incluye funciones para ejecutar pasos E y M por separado, simular datos a partir de los modelos disponibles y visualizar los modelos ajustados junto con los resultados obtenidos en clustering (véase el Capítulo 2), clasificación y estimación de densidad. Las versiones más recientes incorporan funcionalidades como reducción de dimensionalidad para visualización, inferencia mediante remuestreo, nuevos criterios de selección de modelos y opciones avanzadas para la inicialización del algoritmo EM. La página web del paquete `mclust`, junto con otros paquetes relacionados en R, está disponible en: `https://mclust-org.github.io`.

Una muestra de la popularidad de `mclust` puede observarse en los registros de descargas del espejo de CRAN (*The Comprehensive R Archive Network*) mantenido por RStudio (`http://cran-logs.rstudio.com`). Utilizando la base de datos proporcionada por el paquete `cranlogs` (Csárdi, 2019), puede comprobarse que es el paquete más descargado entre aquellos relacionados con modelos de mixtura gaussiana (Scrucca *et al.*, 2023, Sección 1.2). Este paquete se ha utilizado en una amplia variedad de contextos, incluyendo geoquímica (Ellefsen *et al.*, 2014), análisis de secuencias de ADN (Verbist *et al.*, 2015), hidrología (Kim *et al.*, 2014), energía

eólica (Kazor y Hering, 2015), epidemiología (Flynt y Daepp, 2015), psicología clínica (Suveg *et al.*, 2014) y ciencia política (Jang y Hitchcock, 2012).

La forma más sencilla de instalar la versión más reciente de `mclust` desde CRAN es usar el siguiente comando desde la consola de R:

```
install.packages("mclust")
```

Una vez instalado el paquete, se puede cargar en una sesión de R usando el siguiente comando:

```
library("mclust")
##                     __           __
##    ____ ___  _____/ /_  _______/ /_
##   / __ `__ \/ ___/ / / / / ___/ __/
##  / / / / / / /__/ / /_/ (__  ) /_
## /_/ /_/ /_/\___/_/\__,_/____/\__/   version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

La función principal del paquete `mclust` es `Mclust()`. Esta función ajusta, utilizando el algoritmo EM (descrito en las Secciones 1.3.1 y 1.3.3) inicializado por defecto con agrupamiento jerárquico basado en modelos (HMBC, ver Sección 1.3.2), los 14 modelos de covarianza definidos en la Tabla 1.1 para un número de componentes desde 1 hasta 9 (por defecto), y selecciona el modelo óptimo utilizando el criterio BIC (descrito en la Sección 2.2). Esta función cuenta con argumentos opcionales que permiten personalizar el ajuste, como `G`, para especificar el número de componentes, y `modelNames`, para definir las parametrizaciones del modelo de covarianza (véase la Tabla 1.1 y `help(mclustModelNames)` para una descripción de los modelos disponibles), siendo posible indicar uno o varios valores para cada uno.

La función `mclustBIC()` de este paquete permite obtener una matriz de valores BIC para cada uno de los 14 modelos de covarianza y para un número de componentes desde 1 hasta 9 (por defecto). De la misma forma, la función `mclustICL()` permite obtener una matriz de valores ICL (descrito en la Sección 2.2). Al igual que `Mclust()`, estas funciones admiten los argumentos `G` y `modelNames`. Tanto la función `Mclust()` como `mclustBIC()` aceptan un argumento opcional, `x`, que permite reutilizar los valores de BIC obtenidos en una llamada previa a `mclustBIC()` o a `Mclust()` (pasando el componente `BIC` del objeto resultante, accesible mediante `modelo$BIC`), evitando así recalcular el BIC para los mismos modelos.

Veamos esto con un ejemplo. Se va a considerar un conjunto de datos bidimensionales simulado `ej1`. El código utilizado para la generación de estos datos se puede consultar en el Anexo I, en la Sección I.2. Este conjunto de datos ha sido simulado de forma que está compuesto por cuatro clústeres gaussianos con distintas formas, volúmenes y orientaciones (modelo VVV). Además, dos de ellos están muy próximos entre sí, lo que genera un solapamiento considerable y dificulta su

separación clara. Esta configuración resulta especialmente útil para analizar el comportamiento de los criterios BIC e ICL en situaciones de solapamiento, como se comentó en la Sección 2.2. En la Figura 3.1 se muestra el modelo real de la mixtura gaussiana que genera estos datos simulados.

El siguiente código calcula los valores de BIC para todas las estructuras de covarianza desde 1 hasta 9 componentes:

```
BIC <- mclustBIC(ej1)
BIC
## Bayesian Information Criterion (BIC):
##          EII       VII       EEI       VEI       EVI
## 1 -4125.664 -4125.664 -4055.544 -4055.544 -4055.544
## 2 -3970.735 -3926.604 -3970.660 -3899.558 -3853.016
## 3 -3641.021 -3633.105 -3606.252 -3608.927 -3531.000
## 4 -3536.453 -3541.740 -3542.173 -3547.166 -3514.571
## 5 -3527.769 -3544.697 -3532.500 -3550.495 -3531.068
## 6 -3497.534 -3525.952 -3502.630 -3531.569 -3517.792
## 7 -3506.169 -3523.869 -3512.140 -3529.815 -3537.907
## 8 -3522.395 -3545.296 -3528.167 -3551.308 -3547.477
## 9 -3531.911 -3564.276 -3536.934 -3572.143 -3561.437
##          VVI       EEE       VEE       EVE       VVE
## 1 -4055.544 -4061.352 -4061.352 -4061.352 -4061.352
## 2 -3883.188 -3914.603 -3852.135 -3805.471 -3781.270
## 3 -3525.100 -3612.215 -3614.577 -3536.954 -3530.574
## 4 -3527.492 -3545.137 -3551.290 -3491.692 -3506.446
## 5 -3543.482 -3526.881 -3544.426 -3508.310 -3525.922
## 6 -3535.444 -3508.027 -3537.519 -3522.663 -3537.199
## 7 -3543.871 -3517.260 -3535.483 -3520.304 -3550.489
## 8 -3566.537 -3533.382 -3557.091 -3544.471 -3565.401
## 9 -3588.112 -3542.406 -3576.135 -3558.826 -3587.965
##          EEV       VEV       EVV       VVV
## 1 -4061.352 -4061.352 -4061.352 -4061.352
## 2 -3775.268 -3726.924 -3761.185 -3708.437
## 3 -3507.879 -3493.548 -3519.418 -3505.390
## 4 -3495.594 -3513.592 -3498.796 -3514.883
## 5 -3518.915 -3529.965 -3516.710 -3537.875
## 6 -3525.596 -3544.564 -3551.538 -3563.038
## 7 -3546.187 -3565.111 -3562.230 -3580.149
## 8 -3562.584 -3586.566 -3581.509 -3602.159
## 9 -3554.212 -3600.977 -3581.191 -3627.667
##
## Top 3 models based on the BIC criterion:
##     EVE,4     VEV,3     EEV,4
```

```
##  -3491.692  -3493.548  -3495.594
```

En este caso, el criterio BIC selecciona un modelo de mixtura gaussiana con cuatro componentes que presentan matrices de covarianza con igual volumen y orientación, pero diferente forma (modelo EVE). Cabe señalar que puede haber valores de BIC ausentes (denotados por `NA`) que corresponden a modelos que no pudieron ser estimados debido a problemas de singularidad en la estimación de la matriz de covarianza.

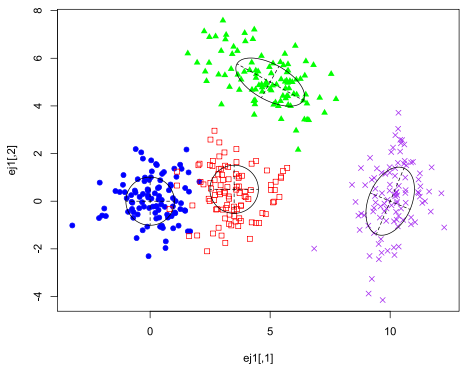

Figura 3.1: Modelo real de mixtura gaussiana del conjunto de datos simulados `ej1`, compuesto por cuatro clústeres con diferentes formas, volúmenes y orientaciones, incluyendo dos clústeres con solapamiento significativo.

El criterio ICL puede calcularse mediante la función `mclustICL()` como sigue:

```
ICL <- mclustICL(ej1)
ICL
## Integrated Complete-data Likelihood (ICL) criterion:
##          EII        VII        EEI        VEI        EVI
## 1 -4125.664 -4125.664 -4055.544 -4055.544 -4055.544
## 2 -3989.335 -3999.312 -3987.850 -3982.347 -3875.099
## 3 -3651.596 -3638.331 -3612.870 -3613.816 -3535.976
## 4 -3566.362 -3564.764 -3570.867 -3569.069 -3545.183
## 5 -3569.643 -3592.930 -3576.369 -3599.281 -3605.477
## 6 -3550.025 -3587.347 -3553.184 -3592.345 -3597.542
## 7 -3583.727 -3573.759 -3589.118 -3580.044 -3641.511
## 8 -3617.094 -3622.002 -3619.987 -3628.099 -3676.069
## 9 -3667.648 -3655.817 -3667.054 -3661.199 -3716.996
##          VVI        EEE        VEE        EVE        VVE
## 1 -4055.544 -4061.352 -4061.352 -4061.352 -4061.352
## 2 -3953.571 -3934.179 -3891.499 -3821.383 -3792.175
## 3 -3528.426 -3618.775 -3619.954 -3541.806 -3533.924
## 4 -3553.971 -3571.074 -3571.771 -3517.499 -3529.122
```

```
## 5 -3592.311 -3566.013 -3585.872 -3579.984 -3576.920
## 6 -3595.518 -3559.659 -3598.436 -3603.259 -3581.928
## 7 -3605.826 -3593.676 -3585.737 -3619.212 -3605.932
## 8 -3655.499 -3624.091 -3632.410 -3670.339 -3628.510
## 9 -3710.305 -3671.237 -3655.673 -3703.071 -3681.077
##         EEV       VEV       EVV       VVV
## 1 -4061.352 -4061.352 -4061.352 -4061.352
## 2 -3796.469 -3739.291 -3767.857 -3715.875
## 3 -3509.752 -3494.864 -3521.337 -3506.736
## 4 -3531.547 -3547.699 -3523.872 -3538.209
## 5 -3621.142 -3566.588 -3604.537 -3584.929
## 6 -3586.567 -3584.643 -3636.439 -3619.807
## 7 -3635.379 -3615.882 -3664.183 -3629.342
## 8 -3676.352 -3682.558 -3725.265 -3660.207
## 9 -3680.462 -3733.868 -3700.918 -3717.462
##
## Top 3 models based on the ICL criterion:
##     VEV,3     VVV,3     EEV,3
## -3494.864 -3506.736 -3509.752
```

En este caso, el criterio ICL selecciona el modelo VEV (matrices de covarianza de las componentes tienen la misma forma, pero diferente volumen y orientación) con tres componentes, una menos que las seleccionadas por el BIC. Esto es razonable, ya que, como se comentaba en la Sección 2.2, el ICL penaliza más fuertemente el solapamiento entre componentes, por lo que tiende a favorecer modelos que produzcan clústeres más claramente separados. Tal como se explicó en la Sección 2.2, esto puede interpretarse como que, en términos de clustering, el número óptimo de clústeres es tres (seleccionado por el ICL). Sin embargo, tres componentes gaussianas en la mixtura no son suficientes para capturar completamente la estructura de los datos, se necesitan cuatro (como elige el BIC).

En la Figura 3.2, se muestran los valores de BIC e ICL calculados para este conjunto de datos. Estas gráficas se obtienen con el siguiente código:

```
plot(BIC)
plot(ICL)
```

Como ya se han calculado los valores de BIC, se pueden pasar mediante el argumento opcional `x` a la función `Mclust()` para evitar recalcularlos, y esta devolverá el modelo óptimo ajustado:

```
mod_bic <- Mclust(ej1, x = BIC)
```

Para ajustar el modelo que ha seleccionado ICL, este se debe indicar a la función `Mclust()` mediante los argumentos opcionales `G` y `modelNames`:

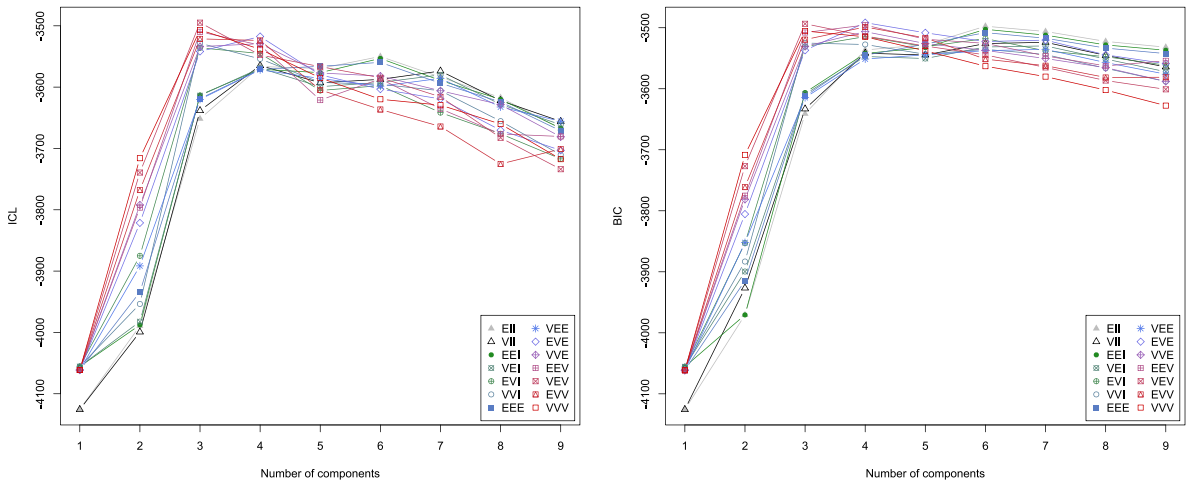

Figura 3.2: Valores de BIC (izquierda) e ICL (derecha) para los modelos de mixtura gaussiana (GMM) estimados a partir del conjunto de datos `ej1`. Cada color representa un tipo diferente de estructura de covarianza para el modelo (véase la Tabla 1.1). El eje horizontal muestra el número de componentes (*Number of components*) considerado, que varía entre 1 y 9, mientras que el eje vertical indica el valor correspondiente de BIC o ICL.

```
mod_icl <-  Mclust(ej1, x = BIC, G = 3, modelNames = "VEV")
```

El siguiente código representa el clustering obtenido según la clasificación MAP (descrita en la Sección 2.1) obtenida por el modelo óptimo estimado seleccionado respectivamente por los criterios de selección de modelos BIC e ICL (ver Figura 3.3):

```
plot(mod_bic, what = "classification")
plot(mod_icl, what = "classification")
```

Además, como se observa en la Figura 3.3, las características geométricas de los clústeres coinciden con lo esperado para cada modelo seleccionado, detallado en la Sección 1.1.1. En el caso del modelo EVE elegido por BIC (izquierda), los tres clústeres tienen igual volumen y orientación, pero diferente forma; mientras que en el caso del modelo VEV escogido por ICL (derecha), los clústeres presentan diferente volumen y orientación, pero comparten la misma forma.

El test de razón de verosimilitud (LRT) discutido en la Sección 2.2 para seleccionar el número de componentes de la mixtura para un modelo específico está implementado en la función `mclustBootstrapLRT()`, en la que se indica el conjunto de datos y el modelo a testear:

```
LRT <- mclustBootstrapLRT(ej1, modelName = "VVV")
LRT
## ---------------------------------------------------------
## Bootstrap sequential LRT for the number of mixture components
```

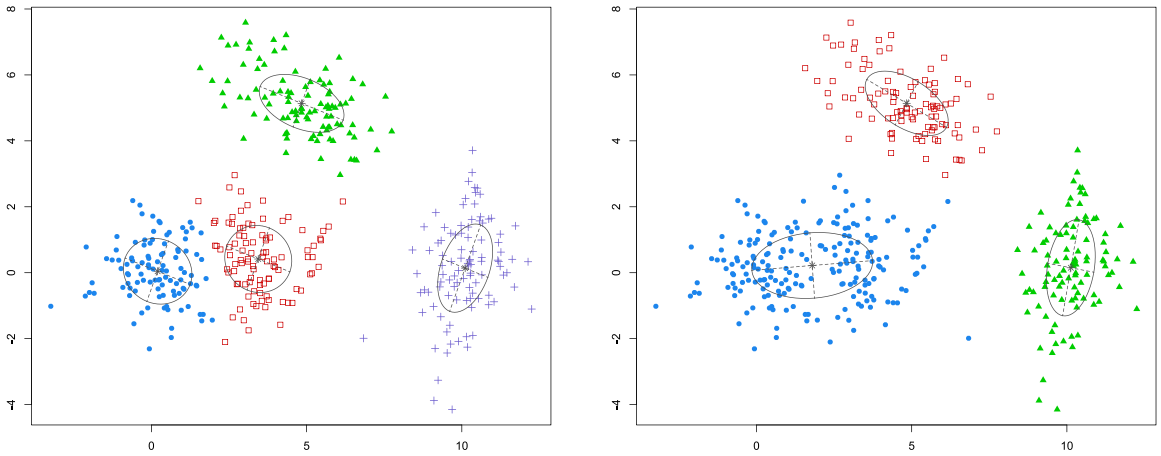

Figura 3.3: Clustering obtenido según los mejores modelos de mixtura gaussiana estimados, seleccionados por BIC (izquierda) e ICL (derecha) para el conjunto de datos `ej1`.

```
## ----------------------------------------------------------
## Model         = VVV
## Replications = 999
##                 LRTS bootstrap p-value
## 1 vs 2    388.86357             0.001
## 2 vs 3    238.99583             0.001
## 3 vs 4     26.45596             0.002
## 4 vs 5     12.95692             0.237
```

Esta función lleva a cabo el procedimiento bootstrap descrito en la Sección 2.2 para calcular una aproximación del p-valor del test de razón de verosimilitud (LRT). Este procedimiento se detiene cuando un test no resulta estadísticamente significativo, según el nivel de significación especificado por el argumento `level` (que por defecto está fijado en 0.05). También es posible limitar el número máximo de componentes de la mixtura a evaluar mediante el argumento `maxG`, y establecer el número de remuestreos bootstrap con el argumento opcional `nboot` (por defecto, `nboot = 999`).

En el ejemplo anterior, los p-valores obtenidos mediante bootstrap indican la presencia de cuatro clústeres. Así, en el primer test se rechaza la hipótesis nula $G = 1$ en favor de la alternativa $G = 2$; en el segundo y tercer test, también se rechaza la hipótesis nula $G = 2$ frente a $G = 3$ y $G = 3$ frente a $G = 4$, respectivamente; mientras que en el cuarto test no hay evidencias estadísticamente significativas para rechazar la hipótesis nula $G = 4$ frente a la alternativa $G = 5$, considerando el modelo VVV sin restricciones para las matrices de covarianza. En la función `mclustBootstrapLRT()` los modelos ajustados a los datos originales se estiman mediante el algo-

ritmo EM, inicializado con un agrupamiento jerárquico basado en modelos (por defecto). Luego, durante el procedimiento bootstrap, los modelos bajo las hipótesis nula y alternativa se ajustan a las muestras bootstrap usando nuevamente el algoritmo EM. Sin embargo, en este caso, el algoritmo comienza con el paso E inicializado con los parámetros estimados previamente a partir de los datos originales. Al usar los parámetros del ajuste original como punto de partida, el algoritmo EM puede converger más rápido y con más estabilidad, ayudando a que las comparaciones sean más fiables, ya que no dependen tanto de variaciones arbitrarias en la inicialización.

El paquete `mclust` utiliza el enfoque de máxima verosimilitud para la estimación de los parámetros (descrito en la Sección 1.3) de los modelos de mixtura gaussiana, empleando para ello el algoritmo EM (ver Secciones 1.3.1 y 1.3.3). La función `summary()` permite obtener información sobre el modelo ajustado y, en particular, sobre este proceso de estimación:

```
mod <- Mclust(ej1)
summary(mod)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EVE (ellipsoidal, equal volume and orientation) model with 4
   components:
##
##  log-likelihood   n df       BIC       ICL
##       -1694.918 400 17 -3491.692 -3517.499
##
## Clustering table:
##   1   2   3   4
## 104  97  99 100
```

Como se ha mencionado anteriormente, `Mclust()` selecciona el modelo óptimo según el criterio BIC, que en este caso corresponde al modelo EEE con tres componentes. Además de los valores de BIC e ICL para el modelo escogido, la salida de `summary()` incluye el valor final de la función de log-verosimilitud $\ell(\hat{\boldsymbol{\Psi}})$ (`log-likelihood`) de los datos observados o incompletos, que es la función objetivo maximizada durante el ajuste del modelo mediante el algoritmo EM (véase Sección 1.3), el número total de observaciones utilizadas en el ajuste del modelo (`n`) y el número de parámetros estimados (`df`), que depende tanto del modelo de covarianza seleccionado (ver Tabla 1.2) como del número de componentes, ya que se estima la media de cada una de ellas y las proporciones de mixtura de todas menos una (dado que deben sumar uno).

La función `Mclust()` emplea el algoritmo EM inicializado por defecto mediante agrupamiento jerárquico basado en modelos (HMBC, ver Sección 1.3.2). Como criterio de parada del proceso iterativo del algoritmo EM, utiliza una tolerancia que define el incremento relativo mínimo per-

mitido de la función de log-versosimilitud entre iteraciones sucesivas, de forma que, cuando dicho incremento es inferior al valor de tolerancia fijado, el algoritmo se detiene (criterio descrito en la Sección 1.3.1). Además, es posible especificar un número máximo de iteraciones para evitar ciclos infinitos en caso de no alcanzar convergencia, o simplemente para limitar el tiempo de cómputo en situaciones en las que el algoritmo tarda demasiado en cumplir el criterio de parada establecido. Por defecto, la tolerancia se establece en 10^{-5} y el número máximo de iteraciones en un valor entero muy grande dado por `.Machine$integer.max`, que equivale a 2 147 483 647. Al ser tan elevado, en la práctica no impone un límite efectivo al número de iteraciones. Estos valores pueden modificarse mediante el siguiente código:

```
emControl <- emControl(tol = 1e-6, itmax = 1150)
mod <- Mclust(ej1, control = emControl)
summary(mod)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EVE (ellipsoidal, equal volume and orientation) model with 4
   components:
##
##  log-likelihood   n df       BIC       ICL
##       -1694.909 400 17 -3491.674 -3517.213
##
## Clustering table:
##   1   2   3   4
## 104  97  99 100
```

En la salida anterior puede observarse que el valor de la función de log-verosimilitud (`log-likelihood`) ha aumentado con respecto al ajuste anterior, realizado con los valores por defecto. Esto se debe a que se han especificado criterios de parada más estrictos (una tolerancia más baja y un mayor número máximo de iteraciones), lo que ha permitido que el algoritmo EM continúe iterando hasta alcanzar una mejor aproximación al óptimo. Se puede comprobar que, si se modifica únicamente el parámetro de tolerancia (`emControl(tol = 1e-6)`), manteniendo el número máximo de iteraciones por defecto, es decir, sin imponer un límite efectivo, el valor final de la función de log-verosimilitud coincide con el obtenido al ajustar ambos parámetros (tolerancia y número máximo de iteraciones). Esto indica que, en este caso, el algoritmo EM converge antes de alcanzar el número máximo de iteraciones fijado.

Además, es posible modificar la tolerancia y el número máximo de iteraciones del procedimiento iterativo interno que se ejecuta para actualizar los parámetros en cada iteración del EM al ajustar los cinco modelos (VEI, VEE, VEV, EVE, VVE) cuyo paso M no tiene forma cerrada.

En este caso, el valor por defecto para el límite de iteraciones es el mismo que para las iteraciones externas del EM (un valor entero muy grande), mientras que el valor por defecto para la tolerancia es un número muy pequeño dado por `.Machine$double.eps`, que en R equivale aproximadamente a $2{,}22 \times 10^{-16}$. Para ello, se debe proporcionar un vector en lugar de un único valor en los parámetros de tolerancia y número máximo de iteraciones, donde el primer elemento corresponde a las iteraciones externas del EM y el segundo a las iteraciones del procedimiento interno. Por ejemplo: `emControl <- emControl(tol = c(1e-6, 1e-4), itmax = c(1150, 1000))`.

Una vez ajustado el modelo de mixtura gaussiana mediante `Mclust()`, el objeto resultante incluye, entre otros elementos, la matriz `mod$z`, que almacena las probabilidades a posteriori de pertenencia a cada componente del modelo para cada observación $\tau_i(\mathbf{y}_j; \hat{\mathbf{\Psi}})$. Tal como se explicó en la Sección 2.1, estas probabilidades se utilizan tanto para asignar cada observación al clúster con mayor probabilidad (clasificación MAP), como para cuantificar la incertidumbre de pertenencia u_j de una observación al clúster al que ha sido asignado. Esta incertidumbre puede visualizarse gráficamente. El siguiente código muestra la incertidumbre asociada a los modelos seleccionados según BIC e ICL (ajustados anteriormente) (ver Figura 3.4):

```
plot(mod_bic, what = "uncertainty")
plot(mod_icl, what = "uncertainty")
```

Estos gráficos, mostrados en la Figura 3.4, representan el espacio de observaciones, donde cada punto está coloreado según el clúster al que ha sido asignado y su tamaño refleja la incertidumbre de clasificación: a mayor incertidumbre, mayor tamaño del punto. Como se puede ver en la leyenda que se ha añadido, los tamaños se normalizan en función de la variación de la incertidumbre del modelo correspondiente. En el modelo seleccionado mediante BIC se identifican cuatro clústeres, con cierto solapamiento entre dos de ellos (en azul y rojo). Esto se manifiesta en la presencia de varios puntos grandes en la zona de intersección, lo que indica que estas observaciones presentan una alta incertidumbre en su asignación debido a la proximidad entre clústeres. En cambio, el modelo seleccionado por ICL propone una solución más simple, con solo tres clústeres bien separados. En este caso, la mayoría de las observaciones presentan una incertidumbre baja (puntos pequeños), lo que sugiere una clasificación más robusta. Aunque aún se observan algunos puntos con incertidumbre moderada (menor que en el modelo BIC) en las zonas de frontera entre clústeres, en general, el modelo ICL proporciona un clustering más claro al priorizar soluciones menos complejas y más robustas frente al solapamiento.

La estrategia de fusión de componentes no gaussianas en mixturas gaussianas de Baudry *et al.* (2010) basada en la entropía, descrita en la Sección 2.3, está implementada en la función `clustCombi` en el paquete `mclust`.

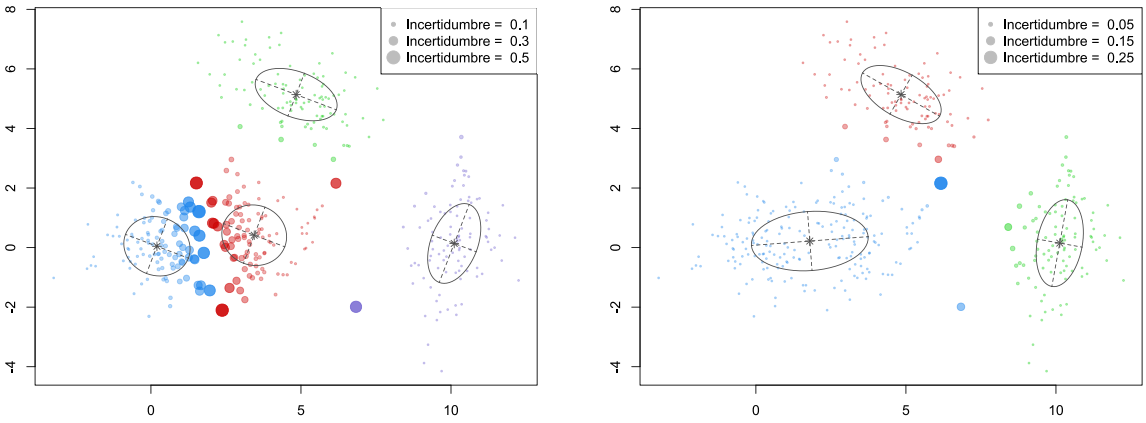

Figura 3.4: Representación de la incertidumbre de clasificación para los modelos seleccionados mediante BIC (izquierda) e ICL (derecha) para el conjunto de datos `ej1`. El color indica el clúster asignado a cada observación mediante clasificación MAP, mientras que el tamaño del punto refleja su incertidumbre de pertenencia: cuanto mayor es el punto, mayor es la incertidumbre asociada a esa asignación. Se ha añadido una leyenda que indica la relación entre el tamaño de los puntos y diferentes niveles de incertidumbre en cada modelo.

Para ilustrar el funcionamiento de esta estrategia, se utilizará otro conjunto de datos bidimensional simulado, `ej2`, que presenta varios clústeres, algunos con estructuras claramente no gaussianas, por lo que resulta ideal para visualizar cómo la fusión mejora el clustering inicial basado en mixturas gaussianas. El código utilizado para generar estos datos puede consultarse en el Anexo I, en la Sección I.3. El siguiente código representa estos datos gráficamente (Figura 3.5, a la izquierda):

```
plot(ej2)
```

Como se observa en la Figura 3.5, el conjunto de datos contiene cuatro clústeres bien separados. Sin embargo, los clústeres en la parte inferior izquierda y superior derecha del gráfico se alejan bastante de una distribución gaussiana, ya que presentan forma de cruz.

El modelo que maximiza el criterio BIC es el modelo EEV, que asume matrices de covarianza con igual volumen y forma, pero diferente orientación, con seis componentes:

```
mod_bic<-Mclust(ej2)
summary(mod_bic)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EEV (ellipsoidal, equal volume and shape) model with 6
```

```
   components:
##
##  log-likelihood   n df       BIC       ICL
##       -1959.085 600 25 -4078.093 -4292.061
##
## Clustering table:
##   1   2   3   4   5   6
##  76 124  91 100 109 100
```

Mientras que el modelo que maximiza el criterio ICL es el modelo VVV, que no impone restricciones sobre las matrices de covarianza de las componentes, con cuatro componentes:

```
mod_icl <- mclustICL(ej2)
icl_matrix <- as.matrix(mod_icl)
mejor_icl_pos <- which(icl_matrix == max(icl_matrix, na.rm = TRUE),
                       arr.ind = TRUE)
G_optimo_icl <- rownames(icl_matrix)[mejor_icl_pos[1]]
modelo_optimo_icl <- colnames(icl_matrix)[mejor_icl_pos[2]]
mod_icl_fit <- Mclust(ej2, G = G_optimo_icl,
                           modelNames = modelo_optimo_icl)
summary(mod_icl_fit)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EVV (ellipsoidal, equal volume) model with 4 components:
##
##  log-likelihood   n df       BIC       ICL
##       -2067.158 600 20 -4262.255 -4263.117
##
## Clustering table:
##   1   2   3   4
## 200 100 200 100
```

La solución basada en BIC, como se puede ver en la Figura 3.5, elige seis componentes gaussianas, dividiendo cada uno de los dos clústeres con forma de cruz en dos componentes, representando cada clúster no gaussiano como una mixtura de dos distribuciones gaussianas. La solución basada en ICL, como se puede ver en la Figura 3.5, selecciona cuatro componentes gaussianas, siendo el número correcto, pero representa cada uno de los dos clústeres con forma de cruz mediante una gaussiana casi esférica, lo que claramente no es un buen ajuste para los datos.

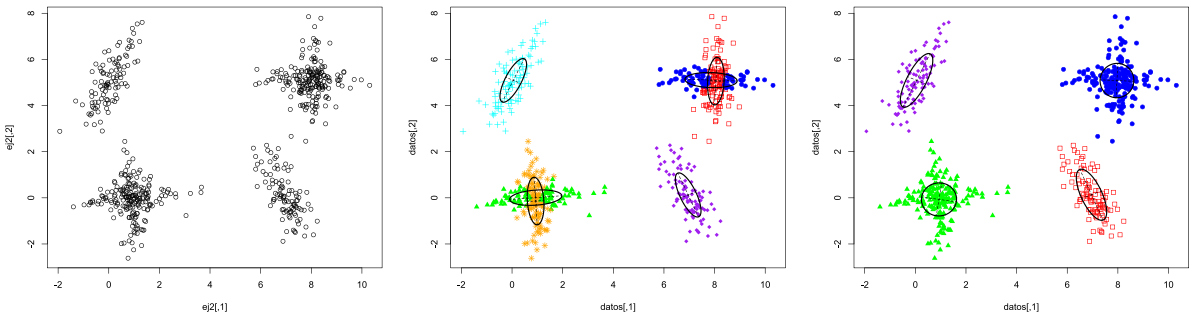

Figura 3.5: Conjunto de datos simulados `ej2` (izquierda), clustering según BIC con seis componentes de mixtura (centro) y clustering según ICL con cuatro componentes de mixtura (derecha).

La función `clustCombi()`, basada en la metodología descrita en Baudry *et al.* (2010), combina de forma jerárquica las componentes de la mixtura utilizando el criterio de entropía presentado en la Sección 2.3. El resultado es una secuencia de modelos con distintos números de clústeres, que va desde un único clúster hasta el número de componentes seleccionadas inicialmente por BIC. Esto se consigue de la siguiente manera:

```
mod_fusion <- clustCombi(mod_bic)
summary(mod_fusion)
## ------------------------------------------------------
## Combining Gaussian mixture components for clustering
## ------------------------------------------------------
##
## Mclust model name: EEV
## Number of components: 6
##
## Combining steps:
##
##  Step | Classes combined at this step | Class labels after this step
## ------|-------------------------------|-----------------------------
##    0  |               ---             | 1 2 3 4 5 6
##    1  |             1 & 2             | 1 3 4 5 6
##    2  |             2 & 5             | 1 3 4 6
##    3  |             1 & 4             | 1 3 6
##    4  |             2 & 6             | 1 3
##    5  |             1 & 3             | 1
```

El resumen anterior muestra el modelo de mixtura EEV con 6 componentes seleccionado por el criterio BIC, seguido de información acerca de los pasos que se han llevado a cabo en la fusión. Las combinaciones de clústeres que se han realizado en cada paso de la fusión pueden representarse gráficamente de la siguiente manera (Figura 3.7):

```
par(mfrow = c(3, 2), mar = c(4, 4, 3, 1))
plot(mod_fusion, ej2, what = "classification")
```

Los modelos de clustering obtenidos con `clustCombi()` pueden compararse mediante el análisis del llamado "gráfico de entropía", que muestra cómo varía esta a medida que se fusionan las componentes, es decir, refleja la evolución de la entropía a lo largo de los distintos modelos generados. Como se explicó en la Sección 2.3, el objetivo es combinar componentes mientras el aumento de entropía sea pequeño, y detenerse cuando dicho incremento se vuelve significativo. Por tanto, la presencia de un "codo" en el gráfico de entropía sirve como guía para decidir el número óptimo de clústeres. Para automatizar esta selección, la función `clustCombiOptim()` implementa un procedimiento que ajusta una regresión lineal por tramos al gráfico de entropía, más concretamente con dos tramos lineales. Esto se hace probando para cada posible punto de corte (es decir, para cada posible número de clústeres) cuál sería el punto que mejor divide el gráfico en dos segmentos con pendientes distintas. El modelo que minimiza el error total de ajuste en ambos tramos identifica ese "punto de cambio de pendiente", es decir, el "codo", sugiriéndolo como la solución más adecuada según este criterio de entropía:

```
optimClust <- clustCombiOptim(mod_fusion, plot = TRUE)
str(optimClust)
## List of 3
##  $ numClusters.combi: int 4
##  $ z.combi          : num [1:600, 1:4] 1 1 1 1 1 ...
##  $ cluster.combi    : num [1:600] 1 1 1 1 1 1 1 1 1 1 ...
```

Este procedimiento genera el gráfico de entropía de la Figura 3.6 y devuelve una lista que contiene el número de clústeres (`numClusters.combi`), las probabilidades (`z.combi`) obtenidas al sumar las probabilidades a posteri de las componentes fusionadas, tal como se describió en la Sección 2.3, y las etiquetas de clustering (`cluster.combi`) correspondientes al modelo sugerido como óptimo según el criterio de entropía. En la Figura 3.6, que muestra como varía la entropía con el número de clústeres, comenzando con un único clúster a la izquierda del gráfico (con entropía cero por definición) y llegando hasta el número de clústeres seleccionado por BIC en el extremo derecho, se puede ver que el punto de cambio de pendiente ("codo") ocurre claramente en 4 clústeres, que es también el número sugerido por la función `clustCombiOptim()` y el número de clústeres real del conjunto de datos.

En esta solución con cuatro clústeres, a diferencia de lo que pasaba con las soluciones obtenidas con BIC e ICL, estos ya no son todos gaussianos, sino que ahora dos de ellos están modelados como mezclas de dos componentes gaussianas cada uno. Cabe destacar que esta solución de cuatro clústeres no es la misma que la identificada por el criterio ICL (mostrada en la Figura 3.5), puesto que el ICL detecta cuatro componentes gaussianas, mientras que este mecanismo de fusión

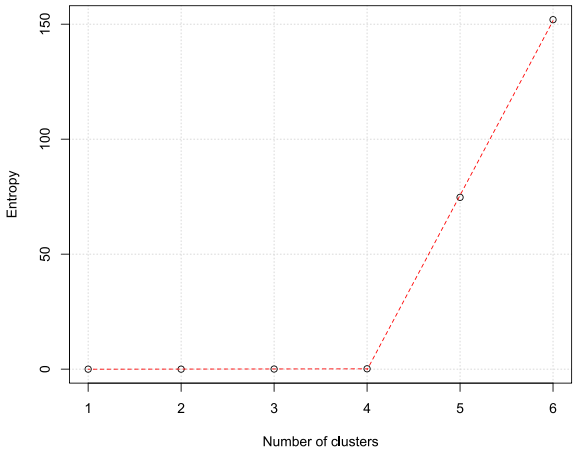

Figura 3.6: Gráfico de entropía para seleccionar el número de clústeres final fusionando las componentes de la mixtura para el conjunto de datos simulado `ej2`. Las dos rectas punteadas corresponden a los segmentos lineales ajustados mediante regresión lineal por tramos que permiten identificar el "codo" o punto de cambio de pendiente en la curva de entropía.

identifica cuatro clústeres, de los cuales dos no son gaussianos.

A continuación, se ilustrará mediante otro ejemplo cómo la fusión de componentes no gaussianos en modelos de mixtura gaussiana puede influir en los resultados del clustering. El código asociado a este ejemplo se encuentra en el Anexo I, en la Sección I.4. Como se discutió en la Sección 2.3, el criterio BIC tiende a seleccionar modelos que representan un grupo no gaussiano mediante varias componentes gaussianas, lo que puede llevar a una sobreestimación del número de grupos. Por otro lado, el criterio ICL suele elegir un número de componentes que coincide con el número real de grupos, representando cada grupo no gaussiano con una única componente gaussiana, lo que puede resultar en un ajuste menos preciso a los datos. En el ejemplo que se presenta, uno de los clústeres (a la izquierda) tiene una forma claramente no gaussiana, en forma de cruz, mientras que el otro (a la derecha) es aproximadamente elíptico, por tanto con forma gaussiana. Esta diferencia se observa en la Figura 3.8, donde BIC selecciona tres clústeres, mientras que ICL opta por dos.

A pesar de que ambos criterios pueden coincidir en el número de clústeres tras aplicar una estrategia de fusión a las componentes seleccionadas por BIC, los resultados del clustering pueden diferir. En este ejemplo, como se puede ver en la Figura 3.8, al fusionar las componentes del modelo seleccionado por BIC para obtener dos clústeres, uno de los clústeres resultantes está compuesto por dos componentes gaussianas, mientras que el otro corresponde a una sola componente. Esto contrasta con la solución proporcionada por ICL, donde cada clúster está representado por una única componente gaussiana. Aunque el número total de clústeres coincide,

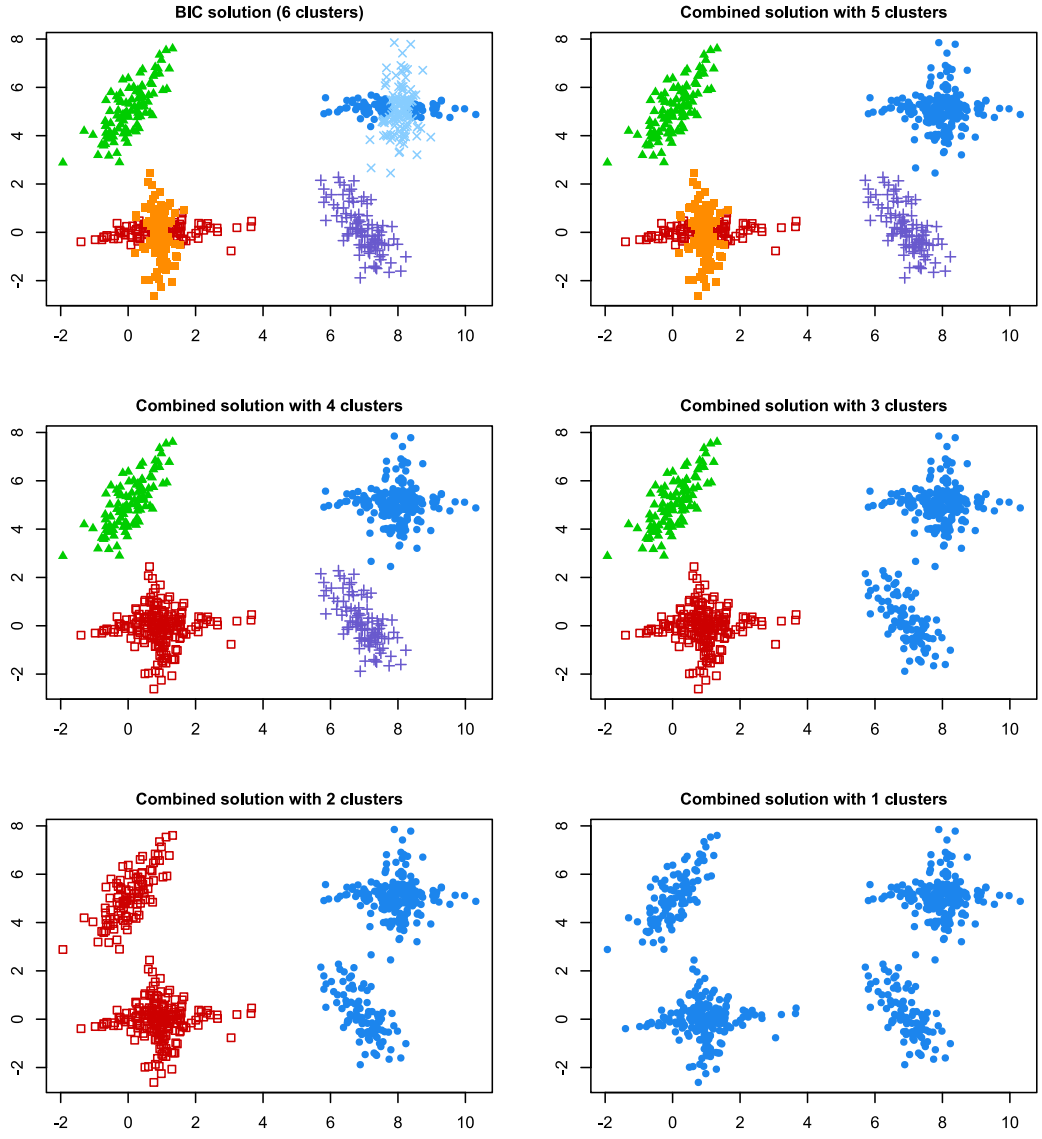

Figura 3.7: Jerarquía de combinaciones de componentes obtenida en la fusión del modelo de mixtura para el conjunto de datos simulado `ej2`. Cada subfigura (de arriba a abajo y de izquierda a derecha) muestra una clasificación resultante de combinar componentes del modelo de mixtura gaussiana ajustado, utilizando el criterio de entropía (Sección 2.3). La jerarquía parte del número de componentes seleccionado por BIC, que desciende hasta una única componente, visualizando cómo se agrupan las observaciones a medida que se reduce el número de clústeres.

la estructura interna del modelo es distinta.

Esta diferencia en la representación afecta tanto a la interpretación del modelo como a la asignación de los datos. Como se observa en la Figura 3.8, el punto resaltado, ubicado entre ambos clústeres, es asignado al clúster izquierdo por el modelo ajustado con ICL. Esto se debe a que ICL representa toda la estructura en forma de cruz como una única componente gaussiana, lo que da lugar a una mayor densidad en la región intermedia. En cambio, el modelo basado en BIC con fusión de componentes asigna ese mismo punto al clúster derecho. En este caso, el clúster izquierdo está representado por dos componentes gaussianas separadas (una elíptica vertical y otra horizontal), lo que genera una zona de menor densidad entre ellas. Como resultado, la densidad posterior en esa región es menor que la del clúster derecho, que está representado por una única componente gaussiana más compacta, lo que lleva a la asignación del punto intermedio a dicho clúster según la regla de clasificación basada en la máxima verosimilitud posterior. Este ejemplo ilustra cómo las decisiones sobre la estructura del modelo, incluso cuando el número total de clústeres coincide, pueden influir notablemente en la asignación de observaciones y en la comprensión de la organización interna de los datos.

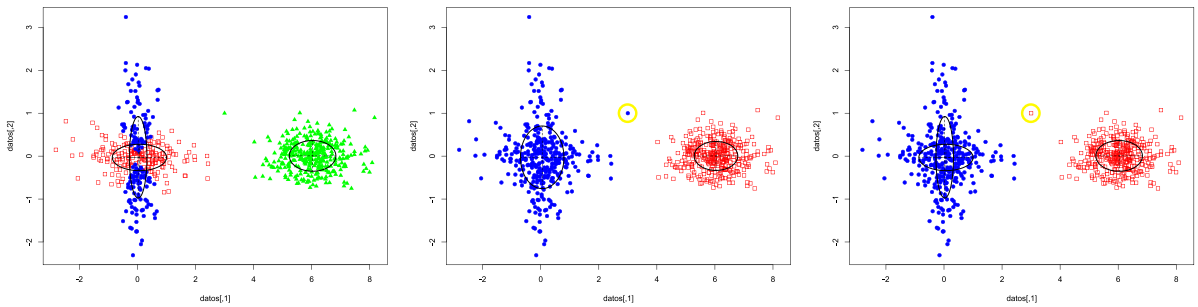

Figura 3.8: Clustering según tres modelos: BIC (izquierda), ICL (centro) y BIC con fusión de componentes (derecha). El punto resaltado en amarillo muestra una diferencia en la asignación entre los modelos ICL y BIC con fusión de componentes, a pesar de tener igual número de clústeres, debido a su distinta representación de la estructura del clúster izquierdo.

Como último ejemplo, se va a utilizar un conjunto de datos simulado `ej4`, para comprobar la capacidad de un modelo de mixtura gaussiana de identificar los clústeres. Este conjunto de datos contiene tres clústeres estructuralmente distintos: un clúster generado como la unión de dos distribuciones normales con medias cercanas; un clúster en forma de cruz, construido como la unión de dos distribuciones normales, una alargada en dirección horizontal y otra en dirección vertical; y un clúster generado a partir de una distribución t de Student con pocos grados de libertad, lo que produce colas pesadas y mayor dispersión en comparación con una distribución normal. El objetivo es analizar cómo distintos criterios de selección de modelos basados en mixturas gaussianas (en concreto, el BIC, el ICL y el método de fusión) identifican estos clústeres.

En el Anexo I, en la Sección I.5, se puede consultar el código utilizado tanto para simular los datos como para ajustar y representar gráficamente los modelos de mixtura gaussiana según cada uno de los criterios mencionados. En la Figura 3.9 se muestra el resultado del clustering obtenido por cada uno de los modelos considerados. El modelo ajustado mediante BIC selecciona seis componentes, ya que son necesarias para representar adecuadamente la distribución de los datos mediante una mixtura de gaussianas. Este comportamiento es esperado, ya que el BIC se centra en ajustar con precisión la distribución observada. En consecuencia, reconoce las dos normales subyacentes en los clústeres formados por combinaciones de distribuciones normales, y utiliza dos componentes gaussianas para modelar el clúster generado a partir de la distribución t de Student, debido a su mayor dispersión. Como se comentó anteriormente, el ICL tiende a priorizar soluciones más simples que las seleccionadas por el BIC, favoreciendo configuraciones con menor número de clústeres. Esto se refleja también en la Figura 3.9, donde ICL selecciona cuatro clústeres, agrupando tanto el conjunto de dos normales elípticas como el clúster con distribución t de Student en una única componente gaussiana en cada caso, aunque es un ajuste menos preciso para los datos. Sin embargo, ICL no logra identificar el clúster en forma de cruz como un solo grupo, dividiéndolo en dos componentes. Esto ocurre porque, aunque ICL es un criterio más conservador que penaliza la incertidumbre en la asignación de observaciones a los clústeres, favoreciendo agrupamientos con asignaciones claras y bien separadas, sigue basándose en la suposición de que cada clúster tiene una distribución normal. En el caso del clúster con forma de cruz, cuya distribución global se aleja claramente de una única forma gaussiana, la asignación de observaciones resulta ambigua si se intenta modelar con una sola componente normal. Por ello, ICL opta por dividirlo en dos componentes gaussianas separadas para mantener clústeres con baja incertidumbre en la clasificación. Por tanto, el único modelo que logra reconocer correctamente los tres clústeres reales es el obtenido mediante el método de fusión de componentes gaussianas a partir del modelo inicial ajustado con BIC. Este enfoque permite combinar las componentes detectadas por BIC en agrupaciones más coherentes con la estructura verdadera de los datos, logrando un equilibrio entre ajuste estadístico y simplicidad interpretativa.

Para evaluar la calidad de la clasificación obtenida por cada modelo, se utiliza el índice de Rand ajustado (*Adjusted Rand Index*, ARI) propuesto por Hubert y Arabie (1985), una métrica ampliamente utilizada que compara el clustering estimado con las etiquetas verdaderas, permitiendo que el número de clústeres reales y estimados difiera, y corrige la medida teniendo en cuenta la posibilidad de agrupamientos aleatorios. El ARI toma valores entre 0 (agrupamiento aleatorio) y 1 (coincidencia perfecta). En este ejemplo, el modelo ajustado mediante BIC obtiene un ARI de 0.63, lo que refleja una coincidencia moderada con la estructura real de los datos. Este valor relativamente bajo se explica porque BIC selecciona un número de componentes superior al número real de clústeres, fragmentando los grupos verdaderos. En contraste, el modelo basado en ICL mejora considerablemente esta correspondencia, con un ARI de 0.86, al seleccionar menos

clústeres y favorecer agrupamientos con asignaciones más claras, aunque aún no logra capturar adecuadamente el clúster con forma de cruz como un único grupo. Finalmente, el modelo seleccionado mediante la fusión de componentes gaussianas a partir del ajuste inicial de BIC alcanza el ARI más alto, 0.97, reflejando una clasificación que se aproxima casi por completo a la estructura real del conjunto de datos. Esto confirma que la fusión es un procedimiento eficaz para combinar componentes fragmentadas y obtener clústeres que representan mejor la verdadera organización de los datos.

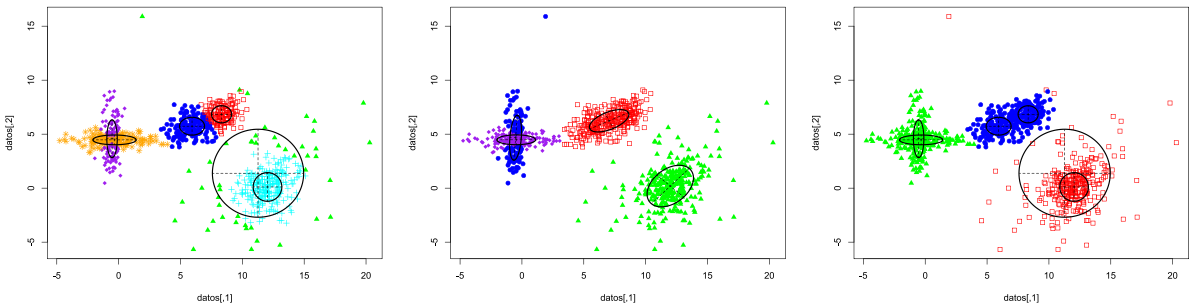

Figura 3.9: Clustering de los datos simulados `ej4` obtenido por tres modelos: BIC (izquierda), ICL (centro) y BIC con fusión de componentes (derecha).

Por último, como se ha podido observar, todos los ejemplos anteriores se han realizado sobre conjuntos de datos bivariantes, ya que estos pueden representarse gráficamente de forma directa. Sin embargo, en muchos casos el número de variables o características utilizadas para el clustering puede ser mayor que dos. En estos casos, con el objetivo de visualizar la estructura de clustering y las características geométricas inducidas por un modelo de mixtura gaussiana (GMM), Scrucca (2010, 2014) propuso una metodología para proyectar los datos en subespacios de dimensión reducida. Estos subespacios están generados por un conjunto de combinaciones lineales de las variables originales, denominadas direcciones GMMDR ("DR" por Dimension Reduction). Estas direcciones se obtienen mediante un procedimiento que busca el subespacio más pequeño que capture la información de clustering contenida en los datos. El objetivo es identificar aquellas direcciones en las que las medias de los clústeres μ_i y/o las covarianzas Σ_i varíen lo máximo posible, siempre que cada dirección sea ortogonal a las demás en un espacio transformado. El número máximo de direcciones útiles es $min(d, G-1)$, siendo d el número de variables y G el número de clústeres. Este enfoque resulta especialmente útil porque permite representar gráficamente (en dos o tres dimensiones) la estructura de los datos y la separación entre grupos, incluso cuando el clustering se ha realizado en espacios de alta dimensión. Para ello, las observaciones se proyectan sobre las primeras direcciones calculadas, que son las que capturan mayor variación entre los grupos y, por tanto, resultan más informativas.

Este método está implementado en la función `MclustDR()` del paquete `mclust`. Para mostrarlo

con un ejemplo, se va a considerar el conjunto de datos `wine`, que contiene medidas fisicoquímicas de 13 variables (como alcohol, ácido málico, alcalinidad de la ceniza, magnesio, fenoles totales, flavonoides, etc.) obtenidas a partir de muestras de tres tipos de vino cultivados en la región italiana de Piamonte, disponible en el paquete `gclus` (Hurley, 2019). Aunque las observaciones están etiquetadas según el tipo de vino, en este caso se emplean únicamente las variables numéricas para realizar un análisis de clustering no supervisado, con el objetivo de identificar grupos naturales en los datos y analizar si la estructura descubierta por el modelo se corresponde con las variedades conocidas.

Primero, se obtiene el modelo estimado con `Mclust()` para este conjunto de datos y después se aplica `MclustDR()` para obtener el subespacio de proyección:

```
data("wine", package = "gclus")
X <- data.matrix(wine[,-1])
mod <- Mclust(X)
drmod <- MclustDR(mod)
summary(drmod)
## -----------------------------------------------------------------
## Dimension reduction for model-based clustering and classification
## -----------------------------------------------------------------
##
## Mixture model type: Mclust (VVE, 3)
##
## Clusters  n
##        1 59
##        2 69
##        3 50
##
## Estimated basis vectors:
##                         Dir1        Dir2
## Alcohol           0.13399009  0.19209123
## Malic            -0.03723778  0.06424412
## Ash              -0.01313103  0.62738796
## Alcalinity       -0.04299147 -0.03715437
## Magnesium        -0.00053971  0.00051772
## Phenols          -0.13507235 -0.04687991
## Flavanoids        0.51323644 -0.13391186
## Nonflavanoid      0.68462875 -0.61863302
## Proanthocyanins -0.07506153 -0.04652587
## Intensity        -0.08855450  0.04877118
## Hue               0.28941727 -0.39564601
## OD280             0.36197696 -0.00779361
## Proline           0.00070724  0.00075867
```

```
##
##                  Dir1    Dir2
## Eigenvalues   1.6189   1.292
## Cum. %       55.6156 100.000
```

En este ejemplo, hay $d = 13$ variables y $G = 3$ clústeres, por lo que el subespacio reducido es bidimensional. El modelo ajustado con `Mclust()` opera en el espacio original de los datos, que tiene dimensión arbitraria. Por ello, sus opciones de `plot` generan gráficos por pares de variables originales. En cambio, `MclustDR()` permite visualizar los datos en un subespacio de dimensión reducida e incorpora opciones adicionales de visualización, como `what = "contour"`, que muestra las curvas de nivel proyectadas de la densidad, y `what = "boundaries"`, que representa las fronteras de incertidumbre de la clasificación MAP (véase Sección 2.1), ambas diseñadas específicamente para representar la proyección bidimensional de los datos. Para ello, aunque el subespacio de reducción de dimensión pueda tener más de dos dimensiones, los datos se proyectan sobre las dos primeras direcciones estimadas, que, como se comentó anteriormente, son las que mejor capturan la variabilidad entre los grupos y, por tanto, resultan más informativas para visualizar la estructura de clustering. Los datos proyectados para este ejemplo, utilizando ambas opciones de `MclustDR()`, se muestran en la Figura 3.10, y se obtienen con el siguiente código:

```
plot(drmod, what = "contour")
plot(drmod, what = "boundaries")
```

Es importante señalar que, como se observa en la Figura 3.10, puede ocurrir que algunas observaciones aparezcan dentro de la frontera de decisión de un clúster distinto al asignado por el modelo. Esto se debe a que la representación corresponde a una proyección bidimensional del espacio original, lo que puede hacer que puntos que estaban claramente separados en el espacio original aparezcan más cercanos o mezclados en la proyección. La opción `what = "boundaries"` muestra las regiones de decisión de la clasificación MAP en el plano proyectado, pero, como el punto se clasifica según la mayor probabilidad a posteriori en el espacio completo, al representarlo en el plano reducido puede aparecer dentro de otra región de decisión, ya que la forma de las regiones puede cambiar al proyectar.

Por último, cabe destacar que el modelo ajustado mediante el criterio BIC selecciona correctamente el número de clústeres y alcanza un índice ARI (*Adjusted Rand Index*, donde 1 equivale a una coincidencia perfecta) Hubert y Arabie (1985) de 0.96, lo que indica una clasificación casi perfecta de las observaciones según los tipos de vino. Esto pone de manifiesto la capacidad del modelo de mixtura gaussiana para detectar agrupaciones en los datos, mostrando su utilidad y buen rendimiento en un contexto real.

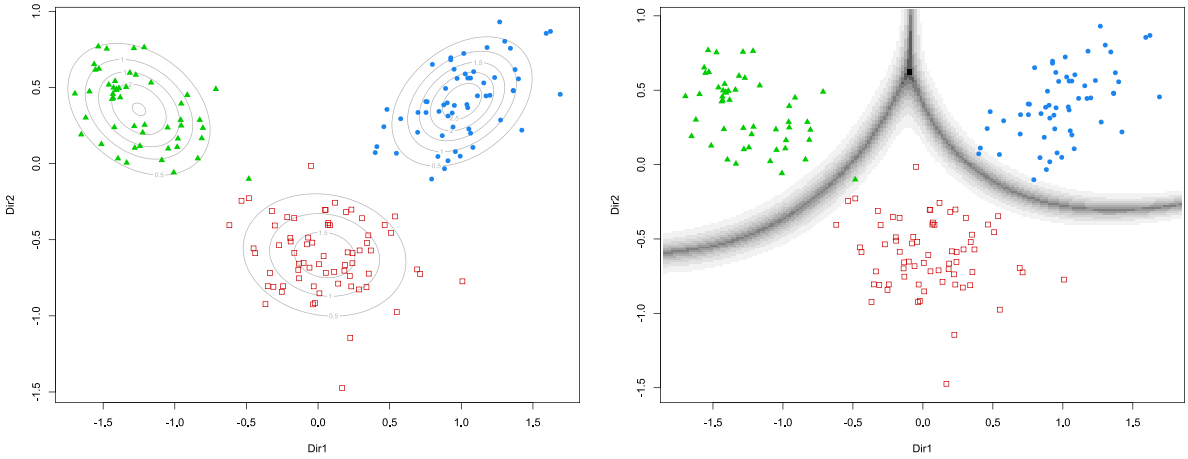

Figura 3.10: Gráfico de superficies de nivel de las densidades estimadas de la mixtura (izquierda) y gráfico de las fronteras de incertidumbre (derecha), proyectados en el subespacio estimado con `MclustDR()` para el conjunto de datos `wine`.

Capítulo 4

Conclusiones

En este trabajo se ha presentado un estudio detallado del clustering basado en modelos de mixtura finita, poniendo especial énfasis en las mixturas gaussianas, que constituyen el caso más común y flexible para datos continuos. Se ha revisado la fundamentación estadística de este enfoque, destacando sus ventajas sobre métodos heurísticos tradicionales, en particular la posibilidad de realizar inferencias rigurosas y evaluar la incertidumbre en la asignación de observaciones a clústeres.

Se ha analizado en profundidad la parametrización de las matrices de covarianza y cómo estas parametrizaciones permiten controlar la complejidad del modelo y facilitar su interpretación, adaptándose a diferentes contextos y tipos de datos. Asimismo, se ha explicado el proceso de estimación mediante máxima verosimilitud y el algoritmo EM, destacando la importancia de una correcta inicialización para obtener resultados fiables.

También se ha abordado el problema de la elección del número de clústeres y del modelo de clustering más adecuado, mediante criterios de selección como BIC e ICL, así como mediante métodos de contraste basados en la log-verosimilitud. Además, se ha mostrado cómo flexibilizar la relación uno a uno entre componentes de la mixtura y clústeres mediante técnicas de fusión, lo que amplía la aplicabilidad práctica de los modelos de mixtura finita.

Finalmente, se ha ilustrado la aplicación práctica de estos conceptos mediante un análisis en R con el paquete `mclust`, mostrando cómo implementar y analizar modelos de mixtura gaussiana con datos simulados y reales.

En resumen, el clustering basado en modelos ofrece un marco sólido, flexible e interpretable para la agrupación de datos continuos, siendo especialmente útil en situaciones donde la incertidumbre y la elección del número de clústeres son aspectos clave. Sin embargo, también presenta ciertas limitaciones, como la necesidad de asumir una forma funcional para las distribuciones y

el alto coste computacional en espacios de gran dimensión, aspectos que deben tenerse en cuenta en aplicaciones prácticas.

En definitiva, el enfoque presentado proporciona una base estadística robusta para el clustering, y su comprensión y correcta aplicación pueden contribuir significativamente a resolver problemas en diversas áreas de la ciencia y la ingeniería donde la agrupación de datos es una tarea fundamental.

A pesar de que este trabajo se ha centrado en los modelos de mixtura gaussiana, se ha presentado un marco general desde el cual es posible construir esquemas de clustering basados en otros modelos. Por ejemplo, la construcción de modelos de mixtura t, también ampliamente utilizados, aunque en menor medida que los gaussianos, se realiza de forma análoga utilizando distribuciones t de Student (McLachlan y Peel, 2000, Capítulo 7). Asimismo, pueden emplearse otras distribuciones dentro de la mixtura para adaptarse a otros tipos de datos, como datos discretos, mediante la elección adecuada del modelo discreto (McLachlan y Peel, 2000, Sección 1.3).

Además, es importante destacar que existen otros enfoques dentro del clustering estadístico. Uno de ellos es el clustering modal (Menardi, 2016), que en lugar de asumir un modelo paramétrico para la distribución de los datos (como hacen los modelos de mixturas), identifica las modas o máximos locales de la función de densidad subyacente, asignando cada observación al clúster correspondiente a la moda a la que converge al seguir el camino ascendente de la densidad, es decir, siguiendo su gradiente. Este enfoque resulta especialmente útil cuando se busca una interpretación más geométrica de las agrupaciones, sin necesidad de asumir una estructura paramétrica específica para los datos.

En conjunto, este trabajo proporciona una base teórica y práctica sólida para el clustering basado en modelos, destacando particularmente el caso de las mixturas gaussianas, poniendo de manifiesto la utilidad y flexibilidad de este enfoque.

Anexo I

Código R

I.1. Función para visualizar resultados de clustering

```
# Colores y formas para los clusters
colores <- c("blue", "red", "green", "purple", "orange", "cyan", "pink",
    "yellow")
formas <- c(19, 0, 17, 18, 8, 3, 4, 0)
# Función para visualizar el clustering
graficar <- function(mod, datos, is_comb = FALSE, level = 0.4) {
  # Comprobación básica
  if (ncol(datos) != 2) {
    stop("Esta función está diseñada para datos bidimensionales.")
  }
  # Cargar el paquete ellipse si no está cargado
  if (!requireNamespace("ellipse", quietly = TRUE)) {
    stop("El paquete 'ellipse' debe estar instalado.")
  }
  # Dibujar los datos
  if (is_comb == TRUE) {
    n <- clustCombiOptim(mod, reg = 2)$numClusters.combi
    clases <- mod$classification[[n]]
    mod <- mod$MclustOutput # Modelo BIC interno, representación real de
        los datos
  } else {
    n <- mod$G
    clases <- mod$classification
  }
  plot(datos, col = colores[clases], pch = formas[clases])
```

```
  # Dibujar elipses de las distribuciones gaussianas
  for (g in 1:mod$G) {
    centro <- mod$parameters$mean[, g]
    sigma <- mod$parameters$variance$sigma[, , g]
    elipse_coords <- ellipse::ellipse(sigma, centre = centro, level =
       level)
    lines(elipse_coords, col = "black", lwd = 2)
  }
  # Dibujar orientaciones
  for (k in 1:mod$G) {
    centro <- mod$parameters$mean[, k]
    Sigma_k <- mod$parameters$variance$sigma[, , k]
    eig <- eigen(Sigma_k)
    v1 <- eig$vectors[, 1] * sqrt(eig$values[1])
    v2 <- eig$vectors[, 2] * sqrt(eig$values[2])
    segments(centro[1], centro[2], centro[1] + v1[1], centro[2] + v1[2],
        col = "black", lwd = 1, lty = 2)
    segments(centro[1], centro[2], centro[1] - v1[1], centro[2] - v1[2],
        col = "black", lwd = 1, lty = 2)
    segments(centro[1], centro[2], centro[1] + v2[1], centro[2] + v2[2],
        col = "black", lwd = 1, lty = 2)
    segments(centro[1], centro[2], centro[1] - v2[1], centro[2] - v2[2],
        col = "black", lwd = 1, lty = 2)
  }
}
```

I.2. Generación de los datos simulados del ejemplo 1

```
library(MASS)

set.seed(123)
n_per_comp <- 100

medias <- list(
  c(0, 0),       # Clúster 1
  c(5, 5),       # Clúster 2
  c(10, 0),      # Clúster 3
  c(3.5, 0.5)      # Clúster 4 (muy pegado a Clúster 1)
)

covs <- list(
```

```
  matrix(c(1, 0, 0, 1), nrow=2),    # Clúster 1, elíptico
  matrix(c(2, -0.8, -0.8, 1), nrow=2), # Clúster 2, otro elíptico
  matrix(c(1, 0.5, 0.5, 2), nrow=2),   # Clúster 3
  matrix(c(1, 0, 0, 1), nrow=2)    # Clúster 4, pequeño y cercano a Clú
     ster 1
)

# Simular datos para cada cluster
ej1 <- do.call(rbind, lapply(1:4, function(i) {
  mvrnorm(n = n_per_comp, mu = medias[[i]], Sigma = covs[[i]])
}))
```

I.3. Generación de los datos simulados del ejemplo 2

```
library(MASS)

set.seed(123)
G_componentes <- 6 # Número de componentes
n_per_comp <- 100 # Número de muestras por componente

# Medias
medias <- list(
  c(8, 5),
  c(7, 0),
  c(1, 0),
  c(1, 0),
  c(0, 5),
  c(8, 5)
)

# Matrices de covarianza
covs <- list(
  matrix(c(1, -0.04, -0.04, 0.1), nrow=2, byrow=TRUE),
  matrix(c(0.3, -0.4, -0.4, 1), nrow=2, byrow=TRUE),
  matrix(c(1, 0, 0, 0.1), nrow=2, byrow=TRUE),
  matrix(c(0.1, -0.04, -0.04, 1), nrow=2, byrow=TRUE),
  matrix(c(0.3, 0.4, 0.4, 1), nrow=2, byrow=TRUE),
  matrix(c(0.1, 0.06, 0.06, 1), nrow=2, byrow=TRUE)
)

# Simular los datos por componente y unirlos
```

```
ej2 <- do.call(rbind, lapply(1:G_componentes, function(i) {
  mvrnorm(n = n_per_comp, mu = medias[[i]], Sigma = covs[[i]])
}))
```

I.4. Ejemplo 3: efecto de la fusión en la asignación de clusters

```
library(mclust)
library(mvtnorm)

# 1. Generación de los datos simulados para el ejemplo
set.seed(123)
n_puntos <- 350

media_cruz <- c(0, 0)
media_circulo <- c(6,0)
cov_matrix_v <- matrix(c(
  0.1, 0,   # Var(X) pequeña (estrecha en eje X)
  0,   1    # Var(Y) grande (alargada en eje Y)
), nrow = 2)

cov_matrix_h <- matrix(c(
  1,   0,   # Var(X) grande (alargada en eje X)
  0,   0.1  # Var(Y) pequeña (estrecha en eje Y)
), nrow = 2)

cov_matrix_circulo <- matrix(c(
  0.6,   0,
  0,   0.1
), nrow = 2)

rama_vertical <- rmvnorm(n_puntos/2, mean = media, sigma = cov_matrix_v)
rama_horizontal <- rmvnorm(n_puntos/2, mean = media, sigma = cov_matrix_
   h)
circulo <- rmvnorm(n_puntos, mean = media_circulo, sigma = cov_matrix_
   circulo)
ej3 <- rbind(rama_horizontal, rama_vertical, circulo)
plot(ej3, main = "Datos simulados")

# Punto entre los dos clústeres que cambiará de asignación
punto_diagonal <- c(3, 1)
points(punto_diagonal[1], punto_diagonal[2], col = "red", pch = 19)
```

```
# Se añade el punto al dataset
ej3 <- rbind(ej3, punto_diagonal)

# 2. Ajuste de los modelos
modelo_bic <- Mclust(ej3) # Modelo con BIC
modelo_fusion <- clustCombi(modelo_bic) # Modelo con fusión de clusters
modelo_icl <- mclustICL(ej3)
icl_matrix <- as.matrix(modelo_icl)
mejor_icl_pos <- which(icl_matrix == max(icl_matrix, na.rm = TRUE), arr.
   ind = TRUE)
G_optimo_icl <- rownames(icl_matrix)[mejor_icl_pos[1]]
modelo_optimo_icl <- colnames(icl_matrix)[mejor_icl_pos[2]]
modelo_icl_fit <- Mclust(ej3, G = G_optimo_icl, modelNames = modelo_
   optimo_icl)

# 3. Visualización del clustering de cada modelo
graficar(modelo_bic, ej3)
graficar(modelo_icl_fit, ej3)
graficar(modelo_fusion, ej3, is_comb = TRUE)
```

I.5. Ejemplo 4: capacidad de un modelo de realizar clustering

```
library(MASS)
library(mvtnorm)
library(mclust)

# 1. Generación de los datos simulados para el ejemplo
set.seed(123)
# Cantidad por subgrupo
n <- 150

# --- Clúster 1: 2 normales separadas horizontalmente ---
mu1a <- c(6, 6)
sigma1a <- matrix(c(1.2, 0.7, 0.7, 1.2), nrow = 2)
cluster1a <- mvrnorm(n = n, mu = mu1a, Sigma = sigma1a)

mu1b <- c(8.3, 6.5)
sigma1b <- matrix(c(0.9, 0.5, 0.5, 1), nrow = 2)
cluster1b <- mvrnorm(n = n, mu = mu1b, Sigma = sigma1b)
```

```
cluster1 <- rbind(cluster1a, cluster1b)

# --- Clúster 2: cruz ---
mu2a <- c(-0.5, 4.5)
sigma2a <- matrix(c(2.5, 0, 0, 0.2), nrow = 2)
cluster2a <- mvrnorm(n = n, mu = mu2a, Sigma = sigma2a)

mu2b <- c(-0.5, 4.5)
sigma2b <- matrix(c(0.2, 0, 0, 2.5), nrow = 2)
cluster2b <- mvrnorm(n = n, mu = mu2b, Sigma = sigma2b)

cluster2 <- rbind(cluster2a, cluster2b)

# --- Clúster 3: t de Student ---
mu3 <- c(12, 0)
sigma3 <- matrix(c(1.5, 0.4, 0.4, 1.5), nrow = 2)
df_t <- 3
cluster3 <- rmvt(n = 2 * n, sigma = sigma3, df = df_t, delta = mu3)

# --- Unir todos los datos ---
ej4 <- rbind(cluster1, cluster2, cluster3)
true_labels <- factor(c(rep("Normal", 2 * n),
                        rep("Cruz", 2 * n),
                        rep("T-Student", 2 * n)))

# --- DataFrame final ---
sim_df <- data.frame(ej4)
colnames(sim_df) <- c("X1", "X2")
sim_df$verdadero <- true_labels

# --- Visualización ---
plot(sim_df$X1, sim_df$X2, col = sim_df$verdadero, pch = 19,
     main = "Conjunto de datos simulados ej4", asp = 1)
legend("topright", legend = levels(sim_df$verdadero), col = 1:3, pch =
   19)


# 2. Ajuste de los modelos
# Clustering según BIC
mod_bic <- Mclust(ej4)
summary(mod_bic)

# Clustering según ICL
```

```
mod_icl <- mclustICL(ej4)
icl_matrix <- as.matrix(mod_icl)
mejor_icl_pos <- which(icl_matrix == max(icl_matrix, na.rm = TRUE), arr.
    ind = TRUE)
G_optimo_icl <- rownames(icl_matrix)[mejor_icl_pos[1]]
modelo_optimo_icl <- colnames(icl_matrix)[mejor_icl_pos[2]]
mod_icl_fit <- Mclust(ej4, G = G_optimo_icl, modelNames = modelo_optimo_
    icl)
summary(mod_icl_fit)

# Clustering según modelo de fusión
mod_fusion <- clustCombi(mod_bic)
summary(mod_fusion)

# 3. Visualización del clustering de cada modelo
graficar(mod_bic, ej4)
graficar(mod_icl_fit, ej4)
graficar(mod_fusion, ej4, is_comb = TRUE)

# 4. Cálculo del ARI
# Predicciones BIC
pred_bic <- mod_bic$classification
# Predicciones ICL
pred_icl <- mod_icl_fit$classification
# Predicciones fusión
optimClust <- clustCombiOptim(mod_fusion)
G <- optimClust$numClusters.combi
pred_fusion <- mod_fusion$classification[[G]]

# Calcular ARI
ari_bic <- adjustedRandIndex(true_labels, pred_bic)
ari_icl <- adjustedRandIndex(true_labels, pred_icl)
ari_fusion <- adjustedRandIndex(true_labels, pred_fusion)

cat("Modelo BIC:\n")
cat("  ARI =", ari_bic, "\n")

cat("Modelo ICL:\n")
cat("  ARI =", ari_icl, "\n")

cat("Modelo Fusión:\n")
cat("  ARI =", ari_fusion, "\n")
```

Bibliografía

Banfield, J. D. y Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821.

Baudry, J. P., Raftery, A. E., Celeux, G., Lo, K., y Gottardo, R. (2010). Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353.

Biernacki, C., Celeux, G., y Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725.

Biernacki, C., Celeux, G., y Govaert, G. (2003). Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics Data Analysis*, 41(3):561–575.

Blashfeld, R. K. y Aldenderfer, M. S. (1988). The methods and problems of cluster analysis. En Nesselroade, J. R. y Cattell, R. B., editores, *Handbook of Multivariate Experimental Psychology*, capítulo 14, pp. 447–474. Plenum Press, New York.

Bouveyron, C., Celeux, G., Murphy, T. B., y Raftery, A. E. (2019). *Model-based clustering and classification for data science: with applications in R*. Cambridge University Press, Reino Unido.

Campbell, J. G., Fraley, C., Murtagh, F., y Raftery, A. E. (1997). Linear flaw detection in woven textiles using model-based clustering. *Pattern Recognition Letters*, 18:1539–1548.

Celeux, G. y Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.

Csárdi, G. (2019). *cranlogs: Download Logs from the 'RStudio' 'CRAN' Mirror*. R package version 2.1.1.

Dasgupta, A. y Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93:294–302.

Dempster, A. P., Laird, N. M., y Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–38.

Efron, B. y Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, London.

Ellefsen, K. J., Smith, D. B., y Horton, J. D. (2014). A modified procedure for mixture-model clustering of regional geochemical data. *Applied Geochemistry*, 51:315–326.

Flynt, A. y Daepp, M. I. (2015). Diet-related chronic disease in the northeastern united states: a model-based clustering approach. *International journal of health geographics*, 14:1–14.

Fraley, C. y Raftery, A. E. (1998). How many clusters? which clustering method? - answers via model-based cluster analysis. *Computer Journal*, 41:578–588.

Fraley, C. y Raftery, A. E. (2003). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 20(2):263–286.

Hennig, C. (2010). Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification*, 4(1):3–34.

Hubert, L. y Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2:193–218.

Hurley, C. (2019). *gclus: Clustering Graphics*. R package version 1.3.2.

Jang, J. y Hitchcock, D. B. (2012). Model-based cluster analysis of democracies. *Journal of Data Science*, 10.

Jeffreys, H. (1961). *Theory of Probability*. Clarendon Press, Oxford, 3 edición.

Kass, R. E. y Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kazor, K. y Hering, A. S. (2015). Assessing the performance of model-based clustering methods in multivariate time series with application to identifying regional wind regimes. *Journal of Agricultural, Biological, and Environmental Statistics*, 20:192–217.

Keribin, C. (1998). Consistent estimate of the order of mixture models. *Comptes Rendus de l'Académie des Sciences. Série I — Mathématiques*, 326:243–248.

Kim, K.-H., Yun, S.-T., Park, S.-S., Joo, Y., y Kim, T.-S. (2014). Model-based clustering of hydrochemical data to demarcate natural versus human impacts on bedrock groundwater quality in rural areas, south korea. *Journal of Hydrology*, 519:626–636.

Leroux, B. G. (1992). Consistent estimation of a mixing distribution. *The Annals of Statistics*, 20(3):1350–1360.

McLachlan, G. J. (1987). On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics*, 36:318–324.

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, Nueva York.

McLachlan, G. J. y Krishnan, T. (2008). *The EM Algorithm and Extensions*. John Wiley & Sons, Hoboken, Nueva Jersey, 2ª edición.

McLachlan, G. J. y Peel, D. (2000). *Finite Mixture Models*. John Wiley & Sons, Nueva York.

McLachlan, G. J. y Rathnayake, S. (2014). On the number of components in a gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5):341–355.

Menardi, G. (2016). A review on modal clustering. *International Statistical Review*, 84(3):413–433.

Neath, A. A. y Cavanaugh, J. E. (2012). The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Roeder, K. y Wasserman, L. (1997). Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439):894–902.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rémi Lebret, Serge Iovleff, Florent Langrognet, Christophe Biernacki, Gilles Celeux, y Gérard Govaert (2015). Rmixmod: The r package of the model-based unsupervised, supervised, and semi-supervised classification mixmod library. *Journal of Statistical Software*, 67(6):1–29.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., y Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267:664–681.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461 – 464.

Scrucca, L. (2010). Dimension reduction for model-based clustering. *Statistics and Computing*, 20(4):471–484.

Scrucca, L. (2014). Graphical tools for model-based mixture discriminant analysis. *Advances in Data Analysis and Classification*, 8(2):147–165.

Scrucca, L., Fop, M., Murphy, T. B., y Raftery, A. E. (2016). mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. *The R Journal*, 8(1):205–233.

Scrucca, L., Fraley, C., Murphy, T. B., y Raftery, A. E. (2023). *Model-based clustering, classification, and density estimation using mclust in R*. Chapman and Hall/CRC, Boca Ratón, Florida.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall/CRC.

Smyth, P. (2000). Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10:63–72.

Stahl, D. y Sallis, H. (2012). Model-based cluster analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(4):341–358.

Stanford, D. C. y Raftery, A. E. (2000). Principal curve clustering with noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:601–609.

Suveg, C., Jacob, M. L., Whitehead, M., Jones, A., y Kingery, J. N. (2014). A model-based cluster analysis of social experiences in clinically anxious youth: links to emotional functioning. *Anxiety, Stress, & Coping*, 27(5):494–508.

Verbist, Bie and Clement, Lieven and Reumers, Joke and Thys, Kim and Vapirev, Alexander and Talloen, Willem and Wetzels, Yves and Meys, Joris and Aerssens, Jeroen and Bijnens, Luc and others (2015). Vivambc: estimating viral sequence variation in complex populations from illumina deep-sequencing data using model-based clustering. *BMC bioinformatics*, 16:1–11.

Wu, C. J. (1983). On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103.