Universidad de Oviedo

Mixturas finitas:
Un modelo estadístico con numerosas aplicaciones

Juan Carballo Fernández

Supervisado por: Raúl Pérez Fernández

UNIVERSIDAD DE OVIEDO

Facultad de Ciencias

Grado en Matemáticas

Julio de 2024

Índice general

Capítulo 1

Introducción

1.1. Motivación

El concepto de mixtura finita surge por primera vez en la literatura a través de la figura de Karl Pearson para modelar una población de cangrejos de la bahía de Nápoles ([15]). La idea de Pearson consistía en dividir la población de cangrejos en dos grupos que representasen cada uno una subespecie distinta y con la formalización de esta idea comienza la historia del que será nuestro objeto de estudio durante el trabajo.

Uno de los principales inconvenientes que tuvieron las mixturas finitas hasta el último cuarto del siglo XX era la dificultad técnica que entrañaba calcular los parámetros para ajustar un modelo de mixturas. Esto se debe a que el método que se utilizaba mayoritariamente para obtener los parámetros era el método de los momentos, el cual requería la búsqueda de raíces de polinomios. Para que nos hagamos una idea, Pearson necesitó resolver un polinomio de grado nueve durante su investigación en una época en la que no existían las facilidades actuales para la resolución numérica de ecuaciones y en la que tuvo que hacer todos los cálculos necesarios a mano.

Fue la publicación de Dempster, Laird y Rubin de 1997 sobre el algoritmo EM ([6]) la que solucionó en parte el problema anterior. Gracias al marco teórico del algoritmo EM se puede abordar la estimación de los parámetros del modelo a través del método de máxima verosimilitud lo cual permite reducir los tiempos de cálculo y le otorga al modelo la posibilidad de clasificar las observaciones de la muestra. Estas ventajas hacen que sea éste el método preferido en la actualidad.

Continuando con la idea de utilizar las mixturas finitas como una herramienta para

clasificar individuos surgen los criterios para elegir el número más adecuado de grupos dentro de una población a través de suponer que cada grupo es una componente de la mixtura. Existen muchas estrategias que se pueden seguir, sin embargo, no hay, de momento, un procedimiento estándar por lo que sigue siendo un punto de debate dentro de la literatura y una de las mayores dificultades dentro de este ámbito.

En definitiva, las mixturas finitas son un modelo muy interesante desde el punto de vista tanto teórico, pues abordaremos temas como el algoritmo EM o la estadística bayesiana, como práctico pues aunque su principal aplicación es el modelado de fenómenos, se encuentran aplicadas en diversos campos científicos.

1.2. Objetivos

El objetivo de este trabajo es desarrollar un marco teórico adecuado que justifique el uso y proporcione herramientas para la aplicación de las mixturas a datos reales. En particular, buscaremos formas de obtener estimadores de los parámetros de una mixtura y estrategias para determinar el número de componentes de las mismas.

1.3. Estructura del trabajo

El trabajo se estructura en capítulos de la siguiente forma: En el capítulo 2 se define el concepto de mixtura de manera formal y se ilustrará, mediante ejemplos, la flexibilidad del modelo. En el capítulo 3 se trata la estimación máximo verosímil de los parámetros utilizando el algoritmo EM aportando ejemplos de uso y funcionamiento. En el capítulo 4 se afronta la elección del número de componentes, se comentan, justifican y se dan ejemplos de uso de diversos criterios. Finalmente, en el último capítulo se citan varias aplicaciones de las mixturas y se trata en especial el análisis clúster desde el punto de vista de las mixturas finitas.

Capítulo 2

Primeras definiciones y ejemplos

Comenzaremos definiendo formalmente el concepto de mixtura finita, posteriormente daremos una serie de ejemplos para ilustrar las diferentes casuísticas que se pueden describir mediante un modelo de mixturas finitas.

2.1. Definición formal de mixtura finita

Atendiendo a la propia definición de mixtura: mezcla, juntura o incorporación de varias cosas [16]; surge de manera natural que una mixtura en nuestro contexto no sea otra cosa que una combinación de vectores aleatorios. Sin embargo, para que dicha combinación sea consistente desde el punto de vista estadístico tendremos que imponerle ciertas condiciones, por ejemplo, que sea un vector aleatorio.

Definición 2.1. *Sea* $\{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_k\}$ *una colección de vectores aleatorios p-dimensionales donde* $\{f_i\}_{i=1}^k$ *son las funciones de densidad asociadas y sea* $\{\pi_1, \ldots, \pi_k\}$ *una colección de números reales positivos verificando* $\sum_{i=1}^k \pi_i = 1$. *Sea* $\vec{\mathbf{y}}$ *el vector aleatorio determinado por la función de densidad*

$$f_{\vec{\mathbf{y}}} : \vec{x} \in \mathbb{R}^p \to f_{\vec{\mathbf{y}}}(\vec{x}) = \sum_{i=1}^{k} \pi_i f_i(\vec{x}) \in \mathbb{R}, \tag{2.1}$$

entonces se dice que $\vec{\mathbf{y}}$ *es una mixtura finita de* $\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_k$ *con pesos* $\pi_1, \ldots, \pi_k$.

Observación 1. *Durante este trabajo nos referiremos a los vectores aleatorios* $\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_k$ *como componentes de la mixtura* $\vec{\mathbf{y}}$ *o, simplemente, como mixturas de* $\vec{\mathbf{y}}$ *cuando no exista ambigüedad. Cuando nos refiramos a mixturas de distribuciones concretas estaremos haciendo referencia a las distribuciones de los vectores aleatorios* $\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_k$.

Lo primero que hemos de comprobar es que la definición está bien dada, es decir, que la expresión de la Ecuación (2.1) es una función de densidad.

Proposición 2.1. *La expresión dada en 2.1 es una función de densidad.*

Demostración.
Comenzaremos probando que $f_{\vec{\mathbf{y}}} \geq 0$. En efecto, basta notar que $f_i \geq 0$ para cualquier índice $i \in \{1, \ldots, k\}$ por ser cada una de ellas función de densidad. Como además los pesos son números no negativos se verifica que $f_{\vec{\mathbf{y}}} = \sum_{i=1}^{k} \pi_i f_i \geq 0$.
Probaremos ahora que $\int_{\mathbb{R}^p} f_{\vec{\mathbf{y}}} = 1$. Basta observar que

$$\int_{\mathbb{R}^p} f_{\vec{\mathbf{y}}}(\vec{x})\, d\vec{x} = \int_{\mathbb{R}^p} \left(\sum_{i=1}^{k} \pi_i f_i(\vec{x}) \right) d\vec{x} = \sum_{i=1}^{k} \left(\pi_i \underbrace{\int_{\mathbb{R}^p} f_i(\vec{x})\, d\vec{x}}_{1} \right) = \sum_{i=1}^{k} \pi_i = 1.$$

□

Por tanto la definición está bien dada y efectivamente una mixtura finita es un vector aleatorio. Veamos entonces algunos ejemplos de este nuevo concepto.

2.2. Ejemplos de mixturas

1. Experimento de los cangrejos de Pearson.

   Como se ha comentado en la introducción este ejemplo es de vital importancia en el desarrollo de la teoría de las mixturas finitas. Si observamos la muestra (Figura 2.1) tomada por el Profesor W.F.R. Weldon y recogida en [15] vemos una distribución asimétrica que imposibilita el ajuste mediante un modelo normal, lo cual impedía utilizar la inferencia paramétrica y obligaba a buscar otros ajustes. El propio Weldon fue el que tuvo la idea de que esta asimetría se debía a la existencia de dos especies distintas observadas dentro de la población. Esto fue lo que le comunicó a Pearson con el objetivo de que fuese este el que buscase el mejor ajuste suponiendo que existían dos grupos suficientemente diferenciados dentro de la población.

   Gracias a este nuevo enfoque comenzó el estudio de las mixturas finitas como una herramienta que permitía una mayor flexibilidad para describir conjuntos de datos en los que falla la estadística paramétrica.

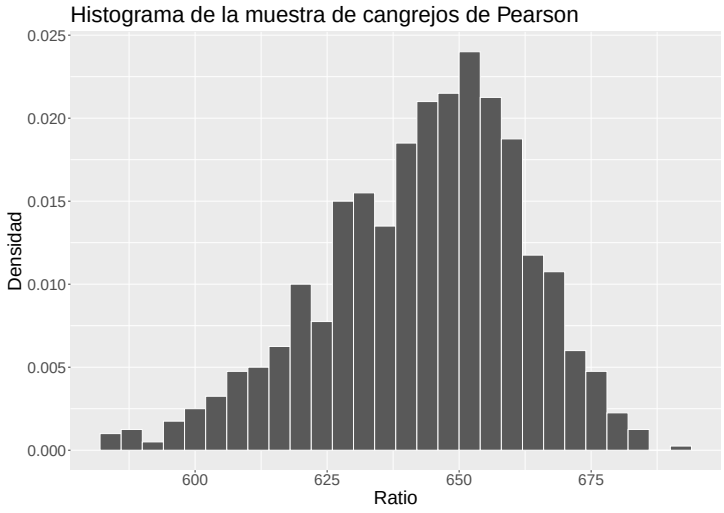

Figura 2.1: Histograma de la muestra obtenida por Weldon.

2. Mixtura de una distribución normal estándar y una distribución de Cauchy estándar.

   Supongamos que tenemos una variable aleatoria $\mathbf{y}$ que sigue una distribución dada por la siguiente función de densidad:

$$f_{\mathbf{y}}(y) = (1-\varepsilon) f_{\mathcal{N}(0,1)}(y) + \frac{\varepsilon}{\pi(1+y^2)}$$

   donde $\varepsilon \in (0,1)$. Es fácil ver que la esperanza de esta distribución no existe pues

$$\begin{aligned}\mathbb{E}(\mathbf{y}) &= \int_{\mathbb{R}} y f_{\mathbf{y}}(y)\, dy = (1-\varepsilon)\int_{\mathbb{R}} y\, f_{\mathcal{N}(0,1)}(y)\, dy + \varepsilon \int_{\mathbb{R}} \frac{y}{\pi(1+y^2)}\, dy \\ &= \varepsilon \int_{\mathbb{R}} \frac{y}{\pi(1+y^2)}\, dy,\end{aligned}$$

   pero la última integral es la esperanza de una distribución Cauchy que no existe, por tanto, la esperanza de la mixtura tampoco puede existir.

   Con este ejemplo queremos resaltar la importancia que tienen las componentes en el comportamiento de la mixtura, pues aunque el peso de una componente sea pequeño puede afectar en gran medida a la mixtura; como en este caso en el que la componente que sigue una densidad Cauchy hará que no exista la esperanza de la mixtura, siempre y cuando su peso sea no nulo. El principal interés de este tipo de modelos es la simulación de valores atípicos de la distribución distinta a la Cauchy (en este caso la normal) pues al mezclarla con la Cauchy ya hemos visto que la esperanza deja de estar definida.

3. Mixturas de normales unidimensionales.

   En este epígrafe queremos poner de manifiesto la gran variedad de formas que puede adoptar la densidad de una mixtura y, en particular, de una mixtura de distribuciones normales univariantes, que le permiten ajustarse a casi cualquier densidad. En la Figura 2.2 tenemos representadas densidades asimétricas, bimodales, trimodales y claw[1] procedentes de mixturas finitas de distribuciones normales, presentando sus correspondientes distribuciones en la Tabla 2.1.

   Estos ejemplos se han elegido por su capacidad de modelar distintos conjuntos de datos. En particular, observamos que la representación de picos de densidad se consigue fácilmente añadiendo componentes a la mixtura y, por otro lado, las asimetrías son fáciles de conseguir modificando los pesos de las componentes. Además, con los ejemplos de las densidades claw ofrecemos densidades modeladas a partir de mixturas finitas que son de especial relevancia en el estudio de la estimación núcleo de la función de densidad.

   En resumen, la clase de las mixturas finitas de distribuciones normales unidimensionales es una clase muy amplia lo que le permite tener una gran flexibilidad a la hora de ajustarse a cualquier tipo de datos o experimento.

Densidad	**Distribución de la mixtura**	**Nº. de componentes**
Bimodal	$\frac{1}{2}\mathcal{N}(-1,\frac{3}{2})+\frac{1}{2}\mathcal{N}(1,\frac{3}{2})$	2
Bimodal asimétrica	$\frac{3}{4}\mathcal{N}(0,1)+\frac{1}{4}\mathcal{N}(\frac{3}{2},\frac{1}{3})$	2
Trimodal	$\frac{9}{20}\mathcal{N}(-1,\frac{1}{3})+\frac{9}{20}\mathcal{N}(1,\frac{1}{3})+\frac{1}{10}\mathcal{N}(0,\frac{1}{5})$	3
Asimétrica	$\sum_{i=0}^{7}\mathcal{N}\left(3\left(\left(\frac{2}{3}\right)^i-1\right)-1,\left(\frac{2}{3}\right)^i\right)$	8
Claw	$\frac{1}{2}\mathcal{N}(0,1)+\sum_{i=-2}^{2}\left(\frac{2^{1-i}}{31}\mathcal{N}\left(i+\frac{1}{2},\frac{2^{-i}}{10}\right)\right)$	4
Claw asimétrica	$\frac{1}{2}\mathcal{N}(0,1)+\sum_{i=0}^{4}\left(\frac{1}{10}\mathcal{N}\left(\frac{i}{2}-1,\frac{1}{10}\right)\right)$	4

Tabla 2.1: Distribuciones de las mixturas representadas en la Figura 2.2.

[1]Introducida en [10].

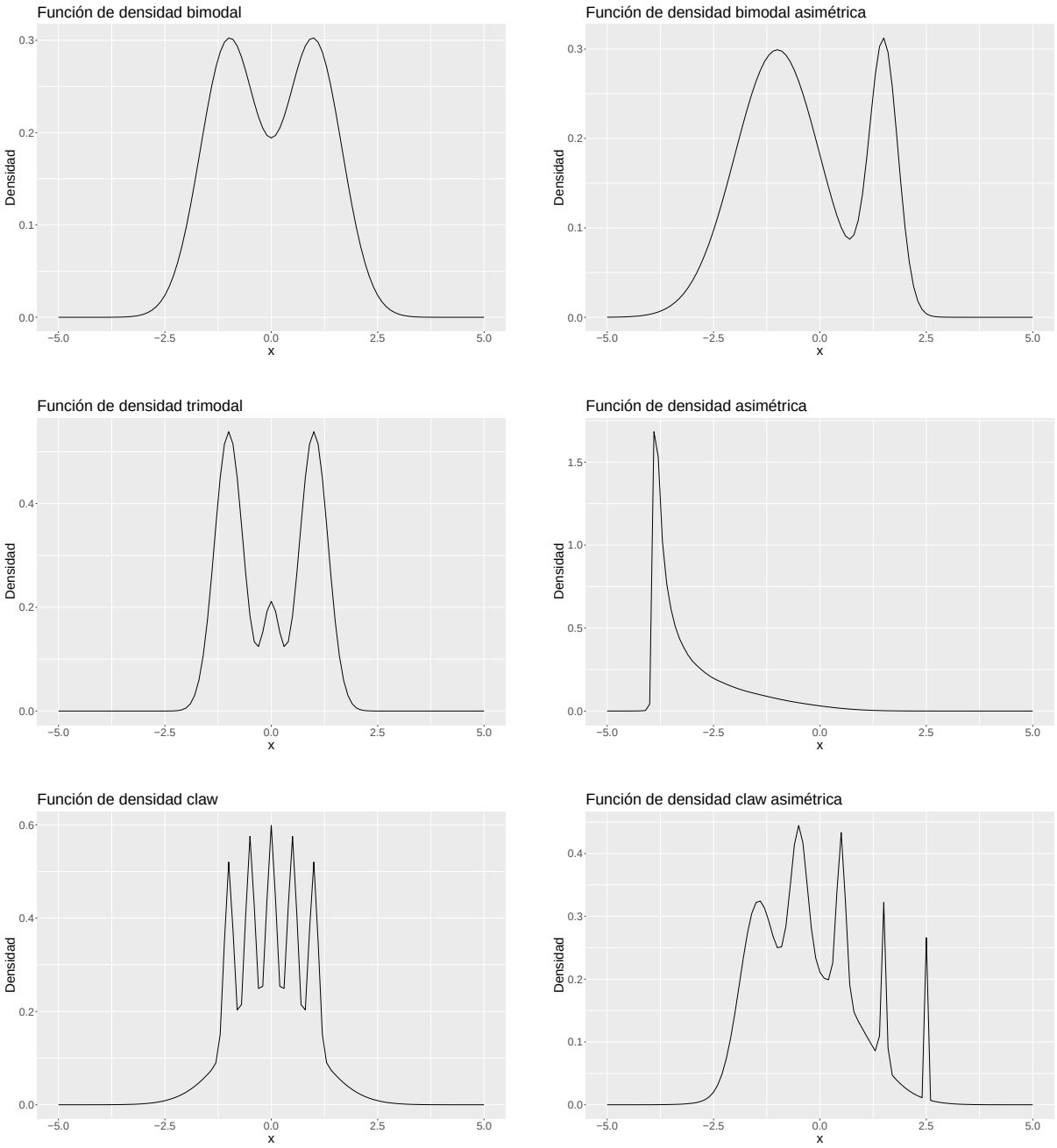

Figura 2.2: Diferentes densidades de mixturas de distribuciones normales univariantes.

Capítulo 3

Inferencia estadística en mixturas

En esta sección analizaremos la manera de conseguir buenos estimadores para los parámetros que determinan completamente a una mixtura. A saber, buscaremos estimadores para los pesos y para los parámetros de cada una de las familias involucradas. Sin embargo, tendremos que imponer la restricción de que el número de mixturas sea fijo para poder hacer cualquier estudio de tipo inferencial.

3.1. Estimación máximo verosímil

Comenzaremos discutiendo los estimadores máximo verosímiles para un modelo de mixturas general donde el número de mixturas es fijo. Posteriormente analizaremos en detalle el caso de las mixturas donde todas sus componentes siguen una distribución normal.

Nos planteamos por tanto la situación más general en la que no conocemos los pesos ni las elementos que caracterizan las distribuciones de la mixtura.

Consideremos entonces un vector aleatorio $\vec{\mathbf{y}}$ p-dimensional mixtura de los vectores $\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_k$ con funciones de densidad $\{f_{i;\theta_i}\}_{i=1}^k$ con pesos $\pi_1, \ldots \pi_k$ desconocidos. Sea $Y \in \mathcal{M}_{n\times p}(\mathbb{R})$ una matriz de datos proveniente de $\vec{\mathbf{y}}$.

La función de verosimilitud surge de forma natural,

$$L(Y, \Psi) = \prod_{i=1}^{n} f_{\vec{\mathbf{y}}}(\vec{y}_i; \Psi) = \prod_{i=1}^{n} \sum_{j=1}^{k} \pi_j f_j(\vec{y}_i; \theta_j)\,,$$

donde $\Psi = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_k\}$. Como de costumbre, resulta más sencillo maximizar

sumas que productos, por lo que trabajaremos con la función de log-verosimilitud

$$l(Y,\Psi) = \log\left(L(Y,\Psi)\right) = \log\prod_{i=1}^{n}\sum_{j=1}^{k}\pi_j f_j(\vec{y}_i;\theta_j) = \sum_{i=1}^{n}\log\left(\sum_{j=1}^{k}\pi_j f_j(\vec{y}_i;\theta_j)\right),$$

que conserva los puntos críticos por la inyectividad del logaritmo.

Formulación como un problema de información incompleta

Aunque se puede tratar de buscar los puntos críticos a partir de la expresión anterior, las ecuaciones que se obtienen analíticamente no resultan útiles (ver [12]). Sin embargo, si suponemos que conocemos la procedencia de cada individuo, es decir, si conocemos de qué componente de la mixtura proviene cada observación $\vec{y}_i'$ con $i \in \{1,\ldots,n\}$ y tenemos esa información almacenada en un vector $\vec{z}$ de manera que

$$\vec{z} = (z_1,\ldots,z_n)',$$

donde z_i toma un valor en $\{1,\ldots,k\}$ dependiendo de qué componente provenga el individuo i-ésimo para cada $i \in \{1,\ldots,n\}$. Notar que bajo la hipótesis de independencia podríamos suponer que $\vec{z}$ proviene de un vector aleatorio con distribución multinomial

$$\vec{\mathbf{z}} \equiv \mathcal{M}\left(k, \begin{pmatrix}\pi_1\\ \vdots \\ \pi_{k-1}\end{pmatrix}\right),$$

ya que esto es equivalente a suponer que, bajo independencia, la probabilidad de que una observación provenga de una componente concreta es el peso de dicha componente.

Consideremos entonces la muestra $Y_c = (Y,\vec{z})$ que llamaremos muestra completa, la verosimilitud de dicha muestra viene dada en este caso por

$$L(Y_c,\Psi) = \prod_{i=1}^{n}\prod_{j=1}^{k}\left(\pi_j f_j(\vec{y}_i;\theta_j)\right)^{\mathbb{1}_{z_i=j}}$$

y su función de log-verosimilitud por

$$l(Y_c,\Psi) = \sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{1}_{z_i=j}\log\left(\pi_j f_j(\vec{y}_i;\theta_j)\right).$$

Naturalmente, en general, no conoceremos la información almacenada en $\vec{z}$ por lo que necesitaremos alguna técnica para conseguir dicha información aunque sea de manera parcial. En particular utilizaremos el *algoritmo EM* desarrollado en [6].

3.2. Algoritmo EM

Siguiendo en el mismo contexto que en lo expuesto anteriormente presentaremos el algoritmo EM. Se trata de un método iterativo en el que partiremos de un valor inicial del vector de parámetros Ψ_0 y tendremos que dar dos pasos comúnmente llamados *expectación* y *maximización* de los cuales toma nombre el algoritmo. Supongamos que estamos en la iteración número t del algoritmo, entonces el algoritmo prosigue de la siguiente forma:

1. Expectación ó paso E.

   Se ha de hallar

   $$Q(\Psi, \Psi_t) = \mathbb{E}_{\Psi_t}\left[l(Y_c, \Psi)|Y\right].$$

2. Maximización ó paso M.

   Tomar Ψ_{t+1} de manera que

   $$Q(\Psi, \Psi_{t+1}) \geq Q(\Psi, \Psi_t).$$

Intuitivamente, lo que haremos será sustituir la verosimilitud original por su valor esperado cuando $\Psi = \Psi_t$ y maximizarlo. Veamos que este procedimiento proporciona una solución adecuada.

Teorema 3.1. *El algoritmo EM aumenta la verosimilitud con cada iteración, es decir, dada Y una matriz de datos cualquiera*

$$L(Y, \Psi_t) \geq L(Y, \Psi_{t-1})$$

para cualquier $t \in \mathbb{N}$.

Demostración.

Sea $t \in \mathbb{N}$, definamos la distribución del vector $\vec{\mathbf{x}} = (\vec{\mathbf{y}}, \vec{\mathbf{z}})'$ condicionado por $\vec{\mathbf{y}} = Y$

$$k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi) = \frac{L(Y_c, \Psi)}{L(Y, \Psi)},$$

de donde podemos deducir

$$l(Y, \Psi) = l(Y_c, \Psi) - k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi).$$

Tomando esperanzas a ambos lados con respecto a Y y Ψ_t notamos que el término $l(Y, \Psi)$ es constante bajo dicha esperanza de donde

$$\begin{aligned} l(Y, \Psi) &= \mathbb{E}_{\Psi_t}\left[l(Y_c, \Psi)\right] - \mathbb{E}_{\Psi_t}\left[k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi)\right] \\ &= Q(\Psi, \Psi_t) - \mathbb{E}_{\Psi_t}\left[\log\left(k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi)\right)\right]. \end{aligned}$$

Por comodidad definimos

$$H(\Psi, \Psi_t) := \mathbb{E}_{\Psi_t}\left[\log\left(k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi)\right)\right]$$

y observamos que basta probar que $H(\Psi, \Psi_{t+1}) - H(\Psi, \Psi_{t+1}) \leq 0$, pues en ese caso

$$l(Y, \Psi_{t+1}) - l(Y, \Psi_t) = Q(\Psi_{t+1}, \Psi_t) - Q(\Psi_{t+1}, \Psi_t) - (H(\Psi_{t+1}, \Psi_t) - H(\Psi_t, \Psi_t)) \geq 0,$$

ya que Ψ_{t+1} se ha escogido de tal manera que $Q(\Psi, \Psi_{t+1}) - Q(\Psi, \Psi_{t+1}) \geq 0$.
Veamos que, en efecto, $H(\Psi, \Psi_{t+1}) - H(\Psi, \Psi_{t+1}) \leq 0$. Sea $\Psi \in \Omega$ cualquiera

$$\begin{aligned}
H(\Psi_{t+1}, \Psi_t) - H(\Psi_t, \Psi_t) &= \mathbb{E}_{\Psi_t}\left[\log\left(k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y)\,; \Psi\right)\right] - \mathbb{E}_{\Psi_t}\left[\log\left(k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y)\,; \Psi_t\right)\right] \\
&= \mathbb{E}_{\Psi_t}\left[\frac{\log\left(k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y); \Psi\right)}{\log\left(k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y)\,; \Psi_t\right)}\right] \\
&= \mathbb{E}_{\Psi_t}\left[\log\left(\frac{k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi)}{k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi_t)}\right)\right] \\
&\leq \log\left(\mathbb{E}_{\Psi_t}\left[\frac{k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi)}{k(\vec{\mathbf{x}}|\vec{\mathbf{y}} = Y; \Psi_t)}\right]\right) \\
&= 0,
\end{aligned}$$

donde hemos aplicado la desigualdad de Jensen. □

Gracias a este resultado tenemos garantías de que el algoritmo proporciona una sucesión de estimaciones que hacen que la verosimilitud sea no decreciente. Sin embargo, esto no asegura que el algoritmo vaya a converger a un máximo global, es más, ni siquiera garantiza que la convergencia sea hacia un máximo local pues existen ejemplos en los que el algoritmo se queda atrapado en puntos de silla. A pesar de todo, se puede demostrar (ver [12]) que en caso de que el algoritmo se encuentre con un punto de silla, una ligera perturbación de la estimación permite al algoritmo alejarse de dicho punto y podremos llegar, por tanto, a un máximo local.

Finalmente, es importante señalar que en [12] también se prueba que en los casos en los que la verosimilitud tenga más de un máximo relativo, la capacidad del algoritmo para converger a uno u otro dependerá de la elección del valor de inicio, por tanto, la importancia de elegir correctamente dicha raíz es significativa.

Elección de la raíz

Tras lo expuesto anteriormente parece natural que la estrategia que sigamos para elegir nuestra raíz tenga en cuenta más de un posible inicio. Es decir, parece recomendable ver

qué ocurre con la verosimilitud para distintas raíces. Con ello en mente, la estrategia que seguiremos en caso de no tener ninguna información relativa al verdadero valor de los parámetros es la siguiente:

Se elige un número cualquiera de puntos iniciales que se consideran también de manera arbitraria; se realiza un número suficiente de iteraciones del algoritmo y se escoge aquella raíz que presente la mayor verosimilitud.

Hemos elegido esta estrategia por su simplicidad y por la libertad que aporta al usuario.

3.3. El algoritmo EM aplicado a las mixturas finitas

En esta sección particularizaremos lo visto anteriormente al caso que nos ocupa: las mixturas finitas. Comenzaremos hallando de manera general la expresión que se necesita en el primer paso. Observamos que en este caso

$$Q(\Psi, \Psi_t) = \mathbb{E}_{\Psi_t}\left[l(Y_c, \Psi)|Y\right] = \mathbb{E}_{\Psi_t}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} \mathbb{1}_{z_i=j} \log\left(\pi_j f_j(\vec{y}_i;\theta_j)\right)\big|Y\right]$$

y todos los términos excepto la función indicadora son constantes al conocer Y. Añadiendo la linealidad de la esperanza al argumento anterior deducimos que

$$Q(\Psi, \Psi_t) = \sum_{i=1}^{n}\sum_{j=1}^{k} \mathbb{E}_{\Psi_t}\left[\mathbb{1}_{z_i=j}|\vec{\mathbf{y}} = \vec{y}_i\right] \log\left(\pi_j f_j(\vec{y}_i;\theta_j)\right),$$

pero

$$\mathbb{E}_{\Psi_t}\left[\mathbb{1}_{z_i=j}|\vec{\mathbf{y}} = \vec{y}_i\right] = P(\vec{\mathbf{z}} = \vec{e}_j|\vec{\mathbf{y}} = \vec{y}_i; \Psi_t)$$

para cada $i \in \{1, \ldots, n\}$ y $j \in \{1, \ldots, k\}$ y, por tanto,

$$Q(\Psi, \Psi_t) = \sum_{i=1}^{n}\sum_{j=1}^{k} P(\vec{\mathbf{z}} = \vec{e}_j|\vec{\mathbf{y}} = \vec{y}_i; \Psi_t) \log\left(\pi_j f_j(\vec{y}_i;\theta_j)\right).$$

Observación 2. *Se adoptará la notación f^t para referirnos a la función de densidad f con los parámetros estimados en la iteración t. De igual manera π^t es el peso estimado en la iteración t.*

Lema 3.2. *Se verifica que $P(\vec{\mathbf{z}} = \vec{e}_j|\vec{\mathbf{y}} = \vec{y}_i; \Psi_t) = \frac{f_j^t(\vec{y}_i)\pi_j^t}{\sum_{l=1}^{k}\pi_l^t f_l^t(\vec{y}_i)}$ para cualquier $\Psi_t \in \Omega$ siempre que $\vec{\mathbf{x}}_j$ tenga como función de densidad f_j para cada $j \in \{1, \ldots, k\}$.*

Demostración.

Sea f la función de densidad de la mixtura formada por los vectores aleatorios $\vec{\mathbf{x}}_j$, entonces a partir de la extensión del teorema de Bayes tenemos que

$$P(\vec{\mathbf{z}} = \vec{e}_j | \vec{\mathbf{y}} = \vec{y}_i; \Psi_t) = \frac{f(\vec{y}_i; \Psi_t) P(\vec{\mathbf{z}} = \vec{e}_j; \Psi_t)}{\sum_{l=1}^{k} f(\vec{y}_i | \mathbf{z}_i = \vec{e}_l; \Psi_t) P(\mathbf{z}_i = \vec{e}_l; \Psi_t)}. \tag{3.1}$$

Habíamos visto que bajo la hipótesis de independencia, para cualquier $j \in \{1, \ldots, k\}$

$$P(\vec{\mathbf{z}} = \vec{e}_j; \Psi_t) = \pi_j^t,$$

pues se distribuye siguiendo un modelo multinomial. Además debido a que el vector $\vec{\mathbf{z}}$ almacena la información relativa a la procedencia de cada individuo surge de manera natural que

$$f(\vec{y}_i | \mathbf{z}_i = \vec{e}_l; \Psi_t) = f_l(\vec{y}_i; \Psi_t) = f_l^t(\vec{y}_i)\,,$$

para cualquier $l \in \{1, \ldots, k\}$. Sustituyendo en (3.1)

$$P(\vec{\mathbf{z}} = \vec{e}_j | \vec{\mathbf{y}} = \vec{y}_i; \Psi_t) = \frac{f_j^t(\vec{y}_i)\pi_j^t}{\sum_{l=1}^{k} \pi_l^t f_l^t(\vec{y}_i)}\,,$$

para cada $j \in \{1, \ldots, k\}$. □

Observación 3. *Por comodidad se define* $\gamma_{ij}^t := P(\vec{\mathbf{z}} = \vec{e}_j | \vec{\mathbf{y}} = \vec{y}_i); \Psi_t)$ *para cualesquiera* $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, k\}$.

Ya estamos en posición de calcular el estimador máximo verosímil de los pesos que como veremos es independiente a la naturaleza de los vectores que intervengan en la mixtura.

Proposición 3.3. *Dado* $j \in \{1, \ldots, k\}$ *fijo, la función* $Q(\Psi, \Psi_t)$ *alcanza un máximo relativo fijando el resto de argumentos en*

$$\pi_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^t}{n}.$$

Demostración.

Comencemos desarrollando la función

$$Q(\Psi, \Psi_t) = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^t \log\left(\pi_j f_j(\vec{y}_i)\right) = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^t \left(\log(\pi_j) + \log\left(f_j(\vec{y}_i)\right)\right).$$

Como los pesos están sujetos a la condición de ligadura

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

utilizaremos el método de los multiplicadores de Lagrange para obtener su estimador máximo verosímil. Así pues la función de Lagrange es

$$L_{\Psi_t}(\Psi) = Q(\Psi, \Psi_t) - \lambda \left(\sum_{j=1}^{k} \pi_j - 1 \right)$$

y para cada $j \in \{1, \ldots, k\}$ su máximo en función de π_j se alcanza en

$$\frac{\partial L_{\Psi_t}}{\partial \pi_j} = \frac{\sum_{i=1}^{n} \gamma_{ij}^t}{\pi_j} - \lambda = 0 \iff \pi_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^t}{\lambda}.$$

Para obtener el valor de λ utilizaremos la condición de ligadura de los pesos, que nos permite escribir

$$1 = \sum_{j=1}^{k} \pi_j = \sum_{j=1}^{k} \left(\frac{\sum_{i=1}^{n} \gamma_{ij}^t}{\lambda} \right) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^t}{\lambda}$$

pero

$$\sum_{j=1}^{k} \gamma_{ij}^t = \sum_{j=1}^{k} P(\mathbf{z}_i = j | \vec{\mathbf{y}} = \vec{y}_i; \Psi_t) = 1$$

por ser una suma sobre todos los posibles resultados de la variable aleatoria $\mathbf{z}_i$. Por tanto,

$$1 = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^t}{\lambda} = \frac{n}{\lambda}$$

de donde $\lambda = n$. Concluimos entonces que

$$\pi_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^t}{n}$$

es un punto crítico de la función. Veamos que es un máximo

$$\frac{\partial^2 L_{\Psi_t}}{\partial \pi_j^2} = -\frac{2 \sum_{i=1}^{n} \gamma_{ij}^t}{\pi_j^2}$$

de donde

$$\left(\frac{\partial^2 L_{\Psi_t}}{\partial \pi_j^2} \right)_{\frac{\sum_{i=1}^{n} \gamma_{ij}^t}{n}} = -\frac{2n^2 \sum_{i=1}^{n} \gamma_{ij}^t}{\left(\sum_{i=1}^{n} \gamma_{ij}^t \right)^2} = -\frac{2n^2}{\sum_{i=1}^{n} \gamma_{ij}^t} < 0$$

pues $\sum_{i=1}^{n} \gamma_{ij}^t = \sum_{i=1}^{n} P(\mathbf{z}_i = j | \vec{\mathbf{y}} = \vec{y}_i; \Psi_t) > 0$. □

3.3.1. El caso de una mixtura de distribuciones normales unidimensionales

Éste es quizás el caso más tratado en la literatura por tratarse de una de las distribuciones centrales de la Estadística. Además, es esta clase la que motiva los primeros estudios de las mixturas finitas con el ya comentado ejemplo de los cangrejos de Pearson.

Al final de esta sección seremos capaces de estimar de manera rigurosa los parámetros de las dos poblaciones observadas en dicho ejemplo.

Supongamos entonces que $\mathbf{x}_i \equiv \mathcal{N}(\mu_i, \sigma_i)$ para cada $i \in \{1, \ldots, k\}$. Entonces,

$$\gamma_{ij}^t = \frac{f_j^t(y_i)\pi_j^t}{\sum_{l=1}^k \pi_l^t f_l^t(y_i)}$$

donde f_j^t denota la función de densidad de una $\mathcal{N}(\mu_j^t, \sigma_j^t)$ y μ_j^t, σ_j^t es el valor de la estimación de μ_j, σ_j en la iteración t. En este caso se puede escribir

$$\begin{aligned}
Q(\Psi, \Psi_t) &= \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \log\left(\pi_j f_j(y_i)\right) = \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \left(\log(\pi_j) + \log\left(f_j(y_i)\right)\right) \\
&= \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \left(\log(\pi_j) + \log\left(\frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(\frac{-(y_i-\mu_j)^2}{2\sigma_j^2}\right)\right)\right) \\
&= \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \left(\log(\pi_j) + \log\left(\frac{1}{\sigma_j\sqrt{2\pi}}\right) + \log\left(\exp\left(\frac{-(y_i-\mu_j)^2}{2\sigma_j^2}\right)\right)\right) \\
&= \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \left(\log(\pi_j) + \log\left(\frac{1}{\sigma_j\sqrt{2\pi}}\right) + \left(\frac{-(y_i-\mu_j)^2}{2\sigma_j^2}\right)\right).
\end{aligned}$$

Es importante observar que γ_{ij}^t no depende del valor de ninguno de los elementos de Ψ pues se ha calculado previamente en función de los valores de la iteración anterior. Con esto en cuenta procedemos al cálculo de los puntos críticos.

En lo que sigue consideraremos un $j \in \{1, \ldots, k\}$ cualquiera pero fijo. Comenzaremos calculando el estimador máximo verosímil para la media:

$$\begin{aligned}
\frac{\partial Q(\Psi, \Psi_t)}{\partial \mu_j} = \sum_{i=1}^n \gamma_{ij}^t \left(\frac{1}{\sigma_j^2}(y_i - \mu_j)\right) = 0 \\
\iff \quad \sum_{i=1}^n \left(\gamma_{ij}^t y_i - \gamma_{ij}^t \mu_j\right) = 0 \\
\iff \quad \sum_{i=1}^n \gamma_{ij}^t y_i - \sum_{i=1}^n \gamma_{ij}^t \mu_j = 0 \\
\iff \quad \sum_{i=1}^n \gamma_{ij}^t y_i = \sum_{i=1}^n \gamma_{ij}^t \mu_j \\
\iff \quad \mu_j \sum_{i=1}^n \gamma_{ij}^t = \sum_{i=1}^n \gamma_{ij}^t y_i \\
\iff \quad \mu_j = \frac{\sum_{i=1}^n \gamma_{ij}^t y_i}{\sum_{i=1}^n \gamma_{ij}^t}.
\end{aligned}$$

Si se calcula la segunda derivada parcial se observa que

$$\frac{\partial^2 Q(\Psi, \Psi_t)}{\partial \mu_j^2} = -\sum_{i=1}^n \frac{\gamma_{ij}^t}{\sigma_j^2} = \frac{-1}{\sigma_j^2} \sum_{i=1}^n \gamma_{ij^t} < 0$$

en todo punto, pues de igual forma que en el caso de los pesos $\sum_{i=1}^{n} \gamma_{ij}^t > 0$.

Notar que el estimador que resulta no es más que la media de las observaciones ponderada por la probabilidad de que pertenezcan a cada una de las componentes de la mixtura. De manera similar podemos obtener el máximo para cada σ_j:

$$\begin{aligned}
\frac{\partial Q(\Psi, \Psi_t)}{\partial \sigma_j} &= \sum_{i=1}^{n} \gamma_{ij}^t \left(\frac{-1}{\sigma_j} + \frac{(y_i - \mu_j)^2}{\sigma_j^3} \right) = 0 \\
\iff \quad & \frac{1}{\sigma_j} \sum_{i=1}^{n} \gamma_{ij}^t = \frac{1}{\sigma_j^3} \sum_{i=1}^{n} \gamma_{ij}^t (y_i - \mu_j)^2 \\
\iff \quad & \sigma_j^2 = \frac{\sum_{i=1}^{n} \gamma_{ij}^t (y_i - \mu_j)^2}{\sum_{i=1}^{n} \gamma_{ij}^t}.
\end{aligned}$$

Derivando una segunda vez

$$\frac{\partial^2 Q(\Psi, \Psi_t)}{\partial \sigma_j^2} = \sum_{i=1}^{n} \gamma_{ij}^t \left(\frac{1}{\sigma_j^2} - \frac{3(y_i - \mu_j^2)}{\sigma_j^4} \right) = \frac{1}{\sigma_{j^2}} \sum_{i=1}^{n} \gamma_{ij}^t \left(1 - \frac{3(y_i - \mu_j)^2}{\sigma_j^2} \right)$$

si denotamos $\hat{\sigma}_j^2 = \frac{\sum_{i=1}^{n} \gamma_{ij}^t (y_i - \mu_j)^2}{\sum_{i=1}^{n} \gamma_{ij}^t}$ y evaluamos la segunda derivada en ese punto

$$\begin{aligned}
\left(\frac{\partial^2 Q(\Psi, \Psi_t)}{\partial \sigma_j^2} \right)_{\hat{\sigma}_j^2} &= \frac{1}{\hat{\sigma}_j^2} \left(\sum_{i=1}^{n} \gamma_{ij}^t - 3 \frac{\sum_{i=1}^{n} (y_i - \mu_j)^2}{\hat{\sigma}_j^2} \right) \\
&= \frac{1}{\hat{\sigma}_j^2} \left(\sum_{i=1}^{n} \gamma_{ij}^t - 3 \sum_{i=1}^{n} \gamma_{ij}^t \right) \\
&= -\frac{2}{\hat{\sigma}_j^2} \sum_{i=1}^{n} \gamma_{ij}^t < 0 \,,
\end{aligned}$$

de donde deducimos que $\hat{\sigma}_j^2$ es un mínimo de la función.

A continuación analizaremos el comportamiento del algoritmo con diferentes datos simulados. Comenzaremos analizando una de las situaciones más favorables al algoritmo, en el cual las medias están suficientemente separadas y los pesos son iguales para ambas componentes. En particular los datos simulados provendrán de una muestra aleatoria simple generada a partir de una mixtura con su densidad dada por

$$f_{\mathbf{x}_1} = \frac{1}{2} f_{\mathcal{N}(-3,1)} + \frac{1}{2} f_{\mathcal{N}(3,1)}.$$

Supondremos, para la aplicación del algoritmo, que la muestra proviene de una mixtura de dos normales. Además generaremos tres conjuntos de datos variando el tamaño muestral para observar cómo varía el desempeño del algoritmo con cada uno de ellos. Simularemos tres conjuntos de datos de tamaño $n_1 = 50$, $n_2 = 100$ y $n_3 = 200$ respectivamente. Comenzamos resaltando el hecho de que no hay apenas solapamiento entre las componentes como podemos observar en la Figura 3.1. Los resultados del algoritmo se encuentran en las Tablas 3.1, 3.2 y 3.3 respectivamente.

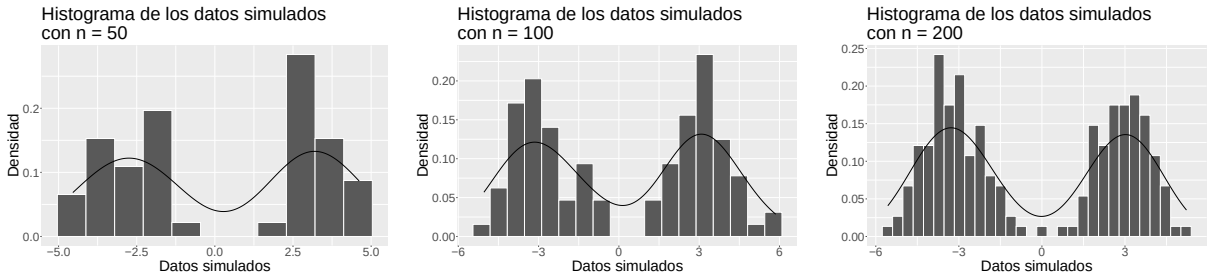

Figura 3.1: Histograma con la función de densidad estimada superpuesta de las muestras simuladas a partir de $\mathbf{x}_1$ con tamaños 50, 100 y 200, de izquierda a derecha.

Parámetro	**Componente 1**	**Componente 2**
Peso	0.50	0.50
Media	-2.78	3.24
Desviación Típica	0.94	0.73

Tabla 3.1: Valores de pesos, medias y desviaciones típicas obtenidos a través del algoritmo EM para el primer conjunto de datos ($n = 50$).

Parámetro	**Componente 1**	**Componente 2**
Peso	0.50	0.50
Media	-2.95	3.19
Desviación Típica	1.13	1.00

Tabla 3.2: Valores de pesos, medias y desviaciones típicas obtenidos a través del algoritmo EM para el segundo conjunto de datos ($n = 100$).

Los valores dados por el algoritmo son muy consistentes para los tres tamaños muestrales y proporcionan una buena estimación de los valores teóricos. Es decir, con independencia del tamaño muestral el algoritmo EM consigue buenas estimaciones de los parámetros cuando las componentes de las mixturas están lo suficientemente separadas como para que no haya solapamiento.

Veamos qué resultados obtenemos para mixturas en las que las componentes estén superpuestas. Para este ejemplo consideraremos muestras aleatorias simples simuladas a partir de una mixtura con función de densidad

$$f_{\mathbf{x}_2} = \frac{3}{10} f_{\mathcal{N}(-1.5,\sqrt{2})} + \frac{3}{10} f_{\mathcal{N}(1.5,\sqrt{2})} + \frac{2}{5} f_{\mathcal{N}\left(0,\frac{1}{2}\right)}.$$

Parámetro	**Componente 1**	**Componente 2**
Peso	0.53	0.47
Media	-3.21	2.97
Desviación Típica	1.06	0.90

Tabla 3.3: Valores de pesos, medias y desviaciones típicas obtenidos a través del algoritmo EM para el tercer conjunto de datos ($n = 200$).

Nuevamente simularemos tres muestras de tamaños $n_1 = 50$, $n_2 = 100$ y $n_3 = 200$ y asumiremos que tenemos tres componentes en nuestra mixtura a la hora de utilizar el algoritmo. En este caso observamos como los histogramas (Figura 3.2) ya no permiten ver a simple vista las tres componentes que conforman las mixturas por lo que la correcta determinación de éstas será, sin duda, más complicada.

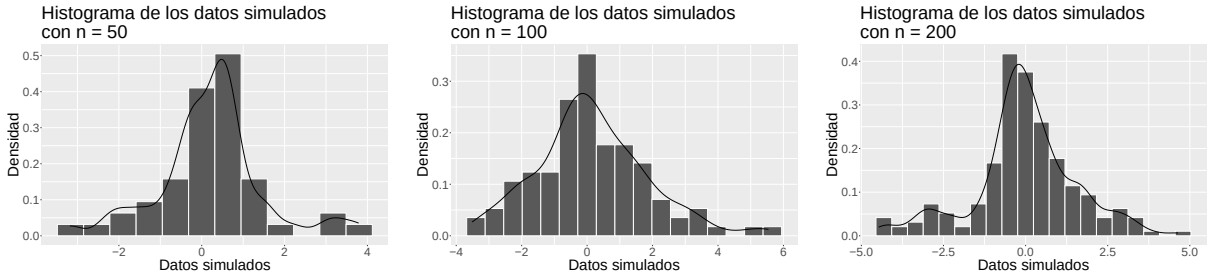

Figura 3.2: Histograma con la función de densidad estimada superpuesta de las muestras simuladas a partir de $\mathbf{x}_2$ con tamaños 50, 100 y 200, de izquierda a derecha.

Como veníamos anticipando, en este escenario los resultados del algoritmo (Tablas 3.4,3.5 y 3.6) no son tan buenos. En particular, observamos que en la muestra de tamaño 50 tanto las estimaciones de la media como de las desviaciones típicas se quedan muy lejos de los valores teóricos. Por otro lado, en este caso sí que se aprecia una mejora al aumentar el tamaño muestral, aún así, el algoritmo no consigue determinar de manera correcta ni los pesos, en ambos casos, ni las desviaciones típicas, en el caso con 100 observaciones, ni las medias, en la muestra con 200 observaciones.

En resumen, cuando las componentes de la mixtura se encuentran superpuestas el algoritmo tiene problemas para identificar correctamente dichas componentes y, por tanto, sus resultados deben de interpretarse con cautela.

Parámetro	**Componente 1**	**Componente 2**	**Componente 3**
Peso	0.40	0.33	0.27
Media	0.26	0.55	-0.25
Desviación Típica	1.94	0.18	0.28

Tabla 3.4: Valores de pesos, medias y desviaciones típicas obtenidos a través del algoritmo EM para los datos simulados a partir de $\mathbf{x}_2$ con tamaño muestral $n = 50$.

Parámetro	**Componente 1**	**Componente 2**	**Componente 3**
Peso	0.33	0.34	0.33
Media	-1.25	1.55	0.12
Desviación Típica	1.18	1.66	0.72

Tabla 3.5: Valores de pesos, medias y desviaciones típicas obtenidos a través del algoritmo EM para los datos simulados a partir de $\mathbf{x}_2$ con tamaño muestral $n = 100$.

3.3.2. El caso de una mixtura de distribuciones exponenciales

Supongamos $\mathbf{x}_i \equiv \mathcal{E}(\lambda_i)$ para cada $i \in \{1, \ldots, n\}$. Entonces, de manera idéntica al caso anterior

$$\gamma_{ij}^t = \frac{f_j^t(y_i)\pi_j^t}{\sum_{l=1}^k \pi_l^t f_l^t(y_i)}$$

donde f_j^t denota la función de densidad de una $\mathcal{E}(\lambda_j^t)$ y λ_j^t es el valor de la estimación de λ_j en la iteración t. Ahora la función auxiliar se escribe

$$\begin{aligned} Q(\Psi, \Psi_t) &= \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \log\left(\pi_j f_j(y_i)\right) = \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \left(\log(\pi_j) + \log\left(f_j(y_i)\right)\right) \\ &= \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \left(\log(\pi_j) + \log\left(\lambda_j e^{-\lambda_j y_i}\right)\right) \\ &= \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^t \left(\log(\pi_j) + \log\left(\lambda_j\right) - \lambda_j y_i\right) \end{aligned}$$

y por tanto podemos calcular, para un $j \in \{1, \ldots, k\}$ fijo, el estimador máximo verosímil de λ_j como sigue

$$\begin{aligned} \frac{\partial Q(\Psi, \Psi_t)}{\partial \lambda_j} &= \sum_{i=1}^n \left(\gamma_{ij}^t \left(\frac{1}{\lambda_j} - y_i\right)\right) = 0 \\ \iff \quad \lambda_j &= \frac{\sum_{i=1}^n \gamma_{ij}^t}{\sum_{i=1}^n \gamma_{ij}^t y_i}. \end{aligned}$$

Parámetro	**Componente 1**	**Componente 2**	**Componente 3**
Peso	0.35	0.33	0.32
Media	-0.21	1.11	-0.97
Desviación Típica	0.41	1.26	1.88

Tabla 3.6: Valores de pesos, medias y desviaciones típicas obtenidos a través del algoritmo EM para los datos simulados a partir de $\mathbf{x}_2$ con tamaño muestral $n = 200$.

Al derivarla una segunda vez obtenemos

$$\frac{\partial^2 Q(\Psi, \Psi_t)}{\partial \lambda_j^2} = \frac{-1}{\lambda_j^2} \sum_{i=1}^{n} \gamma_{ij}^t < 0$$

para cualquier elección de $\lambda_j > 0$. Por tanto el punto crítico encontrado anteriormente es un máximo de la función.

La principal diferencia entre este contexto y el contexto de las mixturas de distribuciones normales es la dependencia entre la media y la varianza pues el parámetro λ caracteriza ambos momentos. Esto hace que la distancia entre los parámetros tenga que ser mayor para poder estimarlos correctamente Veamos un ejemplo de este fenómeno. Consideremos una colección de mixturas $\mathbf{y}_i$ con funciones de densidad

$$\mathbf{y}_i = \frac{3}{4} f_{\mathcal{E}(1)} + \frac{1}{4} f_{\mathcal{E}(\lambda_i)},$$

donde $i \in \{1, \ldots, 6\}$ y $\lambda_i = \frac{1}{2^i}$. Generaremos una muestra aleatoria simple de cada una de las mixturas de tamaño $n = 500$.

Los resultados obtenidos en la Tabla 3.7 concuerdan con lo expuesto, pues vemos que necesitamos distancias muy grandes entre las medias de la distribución para alcanzar buenos resultados.

Variable	Pesos	$\boldsymbol{\lambda}$
$\mathbf{y}_1$	0.50, 0.50	0.7131, 1.0763
$\mathbf{y}_2$	0.48, 0.52	0.3666, 1.5441
$\mathbf{y}_3$	0.46, 0.54	0.1852, 1.2756
$\mathbf{y}_4$	0.39, 0.61	0.0992, 1.3338
$\mathbf{y}_5$	0.33, 0.67	0.0391, 1.2704
$\mathbf{y}_6$	0.31, 0.69	0.0212, 1.1082
Valores Teóricos	0.25, 0.75	$\frac{1}{2^i}$, 1

Tabla 3.7: Valores de pesos y lambda obtenidos mediante el algoritmo EM para las diferentes mixturas de distribuciones exponenciales.

3.3.3. El caso de una mixtura de distribuciones normales multivariantes

Vamos a generalizar el caso normal a p dimensiones, para ello consideremos que $\vec{\mathbf{x}}_i \equiv \mathcal{N}_p(\vec{\mu}_i, \Sigma_i)$ para cada $i \in \{1, \ldots, k\}$ en este caso

$$\begin{aligned} Q(\Psi, \Psi_t) &= \sum_{i=1}^{n}\sum_{j=1}^{k} \gamma_{ij}^t \left(\log(\pi_j) + \log(f_j(\vec{y}_i))\right)) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{k} \gamma_{ij}^t \left(\log(\pi_j) + \log\Big(\frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\vec{y}_i - \vec{\mu}_j)'\Sigma_j^{-1}(\vec{y}_i - \vec{\mu}_j)\right)\Big)\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{k} \gamma_{ij}^t \left(\log(\pi_j) - \frac{p}{2}\log(2\pi) - \frac{1}{2}\log\left(|\Sigma_j|\right) - \left(\frac{1}{2}(\vec{y}_i - \vec{\mu}_j)'\Sigma_j^{-1}(\vec{y}_i - \vec{\mu}_j)\right)\right). \end{aligned}$$

Para optimizar la expresión anterior necesitaremos un lema previo.

Lema 3.4. *Dado un vector $\vec{x} \in \mathbb{R}^n$ y una matriz simétrica $A \in \mathcal{M}_{n\times n}(\mathbb{R}^n)$ se definen*

$$f(\vec{x}) = \vec{x}\,' A\vec{x},$$
$$g(\vec{x}) = A\vec{x}.$$

Entonces

$$\frac{\partial f}{\partial \vec{x}} = 2A\vec{x},$$
$$\frac{\partial g}{\partial \vec{x}} = A.$$

Por tanto dado un $j \in \{1, \ldots, k\}$ fijo

$$\frac{\partial Q(\Psi, \Psi_t)}{\partial \vec{\mu}_j} = \frac{\partial(\vec{y}_i - \vec{\mu}_j)}{\partial \vec{\mu}_j} \frac{\partial Q(\Psi, \Psi_t)}{\partial(\vec{y}_i - \vec{\mu}_j)} = \sum_{i=1}^{n} \gamma_{ij}^t (\Sigma_j^{-1}(\vec{y}_i - \vec{\mu}_j)),$$

igualando a cero

$$\frac{\partial Q(\Psi, \Psi_t)}{\partial \vec{\mu}_j} = 0 \iff \sum_{i=1}^{n} \gamma_{ij}^t (\Sigma_j^{-1}(\vec{y}_i - \vec{\mu}_j)) = 0 \iff \vec{\mu}_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^t \vec{y}_i}{\sum_{i=1}^{n} \gamma_{ij}^t}.$$

Volviendo a derivar y aplicando el Lema 3.4

$$\frac{\partial^2 Q(\Psi, \Psi_t)}{\partial \vec{\mu}_j^2} = -\sum_{i=1}^{n} \gamma_{ij}^t \Sigma_j^{-1},$$

de donde

$$\frac{\partial^2 Q(\Psi, \Psi_t)}{\partial \vec{\mu}_j^2} = -\sum_{i=1}^{n} \gamma_{ij}^t \Sigma_j^{-1},$$

que es una matriz definida negativa pues Σ_j^{-1} es definida positiva (por serlo Σ_j).

Para la obtención del estimador de Σ_j se necesitan conceptos del cálculo matricial que quedan fuera del alcance de este trabajo, se puede consultar [12] para más detalles. Utilizaremos el estimador proporcionado en la cita:

$$\Sigma_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^t (\vec{y}_i - \vec{\mu}_j)(\vec{y}_i - \vec{\mu}_j)'}{\sum_{i=1}^{n} \gamma_{ij}^t}.$$

Finalmente, realizaremos un análisis similar del comportamiento del algoritmo al que hicimos en el caso de las mixturas de normales univariantes. Para ello consideraremos dos mixturas $\vec{\mathbf{z}}_1$ y $\vec{\mathbf{z}}_2$, dadas por las funciones de densidad

$$f_{\vec{\mathbf{z}}_1} = \frac{1}{4} f_{\mathcal{N}_2(\vec{\mu}_1, I_2)} + \frac{1}{4} f_{\mathcal{N}_2(\vec{\mu}_2, I_2)} + \frac{1}{4} f_{\mathcal{N}_2(\vec{\mu}_3, I_2)} + \frac{1}{4} f_{\mathcal{N}_2(\vec{\mu}_4, I_2)}$$

y

$$f_{\vec{\mathbf{z}}_2} = \frac{1}{5} f_{\mathcal{N}_2(\vec{\mu}_1, I_2)} + \frac{1}{5} f_{\mathcal{N}_2(\vec{\mu}_2, I_2)} + \frac{1}{5} f_{\mathcal{N}_2(\vec{\mu}_3, I_2)} + \frac{1}{5} f_{\mathcal{N}_2(\vec{\mu}_4, I_2)} + \frac{1}{5} f_{\mathcal{N}_2(\vec{\mu}_5, I_2)},$$

donde

$$\vec{\mu}_1 = (-2, 2)', \quad \vec{\mu}_2 = (2, 2)', \quad \vec{\mu}_3 = (-2, 2)', \quad \vec{\mu}_4 = (2, -2)', \quad \vec{\mu}_5 = (0, 0)'$$

y la matriz I_2 es la matriz identidad de dimensión 2×2.

La idea detrás de estas mixturas es poder ver las diferencias en el proceder del algoritmo cuando las componentes están separadas y cuando no lo están. Comenzaremos

considerando una matriz de datos de proveniente de cada una de las mixturas. Comenzaremos analizando los resultados para $\vec{\mathbf{z}}_1$ recogidos en la Tabla 3.8 en la que observamos que el método consigue estimar con bastante precisión los parámetros de la mixtura. Sin embargo, para la muestra obtenida a partir de $\vec{\mathbf{z}}_2$ los resultados que se presentan en la Tabla 3.9 no son los deseables pues vemos que el algoritmo no es capaz de estimar correctamente la componente central. A pesar de ello, si nos fijamos en la asignación que hace el algoritmo de los puntos observamos, a partir de las probabilidades γ_{ij}, como las formas están bastante bien reconstruidas coincidiendo en gran medida con las formas que reconstruye en primer caso, en el que sí estima todas las componentes de manera correcta (Figuras 3.3, 3.4).

En definitiva, en el caso multivariante el algoritmo sigue teniendo los mismos problemas con la estimación en caso de que haya superposición entre las componentes pero consigue establecer agrupaciones coherentes para las observaciones.

Componente	Pesos	Media	Matriz varianzas-covarianzas
1	0.24	(2.14, -1.75)	$\begin{pmatrix} 0.96 & 0.04 \\ 0.04 & 0.88 \end{pmatrix}$
2	0.28	(-1.94, -2.06)	$\begin{pmatrix} 0.75 & -0.27 \\ -0.27 & 1.45 \end{pmatrix}$
3	0.22	(1.98, 1.83)	$\begin{pmatrix} 0.65 & 0.25 \\ 0.25 & 0.74 \end{pmatrix}$
4	0.26	(-1.93, 1.89)	$\begin{pmatrix} 1.08 & -0.07 \\ -0.07 & 1.06 \end{pmatrix}$

Tabla 3.8: Valores obtenidos mediante el algoritmo EM para la muestra obtenida a partir de $\vec{\mathbf{z}}_1$.

Componente	Pesos	Media	Matriz de varianzas-covarianzas
1	0.21	(1.59, -1.59)	$\begin{pmatrix} 1.67 & -0.61 \\ -0.61 & 1.22 \end{pmatrix}$
2	0.17	(-2.39, -1.79)	$\begin{pmatrix} 0.64 & -0.08 \\ -0.08 & 0.45 \end{pmatrix}$
3	0.21	(2.24, 2.42)	$\begin{pmatrix} 0.75 & -0.20 \\ -0.20 & 0.8 \end{pmatrix}$
4	0.21	(-0.15, 0.85)	$\begin{pmatrix} 0.80 & 0.44 \\ 0.44 & 1.50 \end{pmatrix}$
5	0.20	(-2.11, 2.03)	$\begin{pmatrix} 0.42 & -0.07 \\ -0.07 & 1.20 \end{pmatrix}$

Tabla 3.9: Valores obtenidos mediante el algoritmo EM para la muestra obtenida a partir de $\vec{\mathbf{z}}_2$.

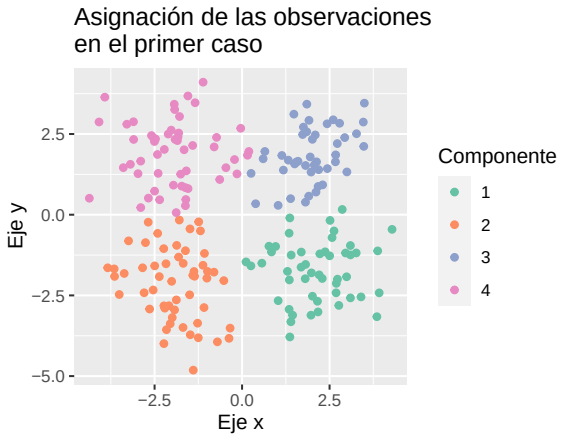

Figura 3.3: Clasificación de la muestra proveniente de $\vec{\mathbf{z}}_1$.

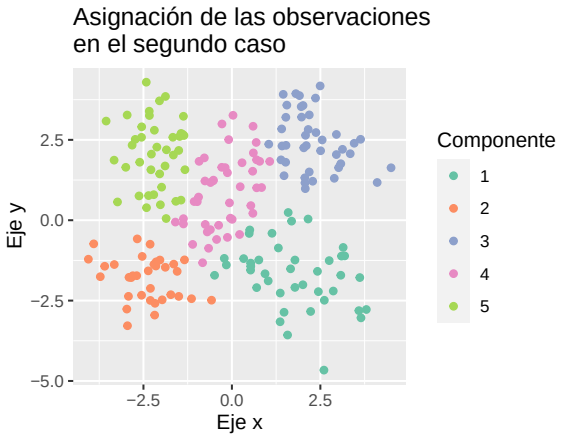

Figura 3.4: Clasificación de la muestra proveniente de $\vec{\mathbf{z}}_2$.

Capítulo 4

Análisis del número de mixturas

Hasta ahora siempre se ha supuesto que teníamos un número fijo de mixturas, pero, ¿cuál es ese número? Éste es un problema complejo del cual no se ha podido obtener aún una solución completa.

En esta sección analizaremos el orden de la mixtura teniendo como objetivo determinar el número de grupos diferenciados que existen en la población. Para ello trataremos de proporcionar criterios suficientes para determinar el número de mixturas que mejor se ajuste a la información disponible de dicha población. En particular, discutiremos dos alternativas diferentes: tomar como criterio una verosimilitud penalizada por el número de mixturas, es decir, con un término que disminuya conforme aumenta dicho número; o utilizar como criterio el test de la razón de verosimilitudes.

4.1. Test de la razón de verosimilitudes

4.1.1. Formalización

Fijemos una muestra $\vec{y}$ proveniente de una mixtura $\vec{\mathbf{y}}$ con un número de componentes desconocido. Con el fin de determinar dicho número plantearemos el siguiente contraste de hipótesis

$$\begin{aligned} H_0 &\equiv k = k_0 \\ H_1 &\equiv k = k_1 \end{aligned} \tag{4.1}$$

donde k es el número de componentes de la mixtura y $k_0, k_1 \in \mathbb{N}$ verificando que $k_1 > k_0$. Trataremos de resolverlo utilizando el estadístico de la razón de verosimilitudes λ. Sean

Ψ_0, Ψ_1 los estimadores máximo verosímiles de Ψ bajo las hipótesis nula y alternativa respectivamente, entonces la transformación Del estadístico del test de la razón de verosimilitudes se escribe

$$\Lambda(\vec{y}) = \frac{L(\vec{y}, \Psi_0)}{L(\vec{y}, \Psi_1)}\,.$$

Si consiguiésemos resolver dicho contraste podríamos ir tomando de manera iterativa k_1 de manera que $k_1 = k_0 + 1$ hasta que no se rechace la hipótesis nula. El primer problema que encontramos es que, como ya hemos visto, no somos capaces de hallar los máximos de la función de verosimilitud sin recurrir a procedimientos iterativos y por tanto no podremos determinar tampoco la expresión analítica del estadístico de la razón de verosimilitudes.

El segundo problema surge al tratar de utilizar la convergencia asintótica de $-2\log\Lambda$ bajo la hipótesis nula a la distribución χ^2 para resolver el contraste pues ésta no se da en condiciones generales, para un ejemplo de esto ver [13].

Por tanto, no nos queda otra opción que no sea simular la distribución del estadístico.

4.1.2. Simulación del estadístico de la razón de verosimilitudes

Para simular la distribución de Λ de manera general se puede seguir el siguiente procedimiento de manera reiterada:

Partiremos de una muestra $\vec{y}$ procedente de una mixtura con función de densidad f_Ψ y calcularemos, utilizando el algoritmo EM, el estimador máximo verosímil de Ψ. Denotémoslo por $\hat{\Psi}$. A continuación, generaremos de manera aleatoria una muestra $\widehat{\vec{y}}$ de igual tamaño que la original procedente de la mixtura con función de densidad $f_{\hat{\Psi}}$. Finalmente, evaluaremos $\Lambda(\widehat{\vec{y}})$.

Repitiendo el procedimiento anterior un número suficiente de veces obtendremos una buena aproximación de la distribución del estadístico bajo la hipótesis nula. En base a estas simulaciones se han ido formulando hipótesis acerca de la distribución de dicho estaditico, en especial, para el caso en el que las mixturas sean distribuciones normales. La primera fue dada para mixturas normales cuyas componentes comparten la misma matriz de varianzas-covarianzas por Wolfe en [18], donde propone que para el contraste de la forma de la Ecuación (4.1) para el que el estadístico de la razón de verosimilitudes se puede aproximar mediante

$$-2C\log\Lambda \approx_{H_0} \chi^2_d\,,$$

donde $C = N - 1 - m - \frac{k_1}{2}$ y $d = 2m(k_1 - k_0)$, siendo N el tamaño muestral y m el número de variables a estimar bajo la hipótesis nula. Wolfe llegó a esta conjetura a partir de una serie de simulaciones para el caso $k_0 = 1$ y $k_1 = 2$.

Posteriormente, McLachlan [11] analizó si la propuesta de Wolfe era admisible si se eliminaba la homocedasticidad. Sin embargo, a través de sus simulaciones observó que cuando no se restringían las varianzas la distribución del estadístico parecía asemejarse más a una χ^2_6 que a una χ^2_4 que sería la extensión natural de la hipótesis de Wolfe. A pesar de los numerosos estudios que existen sobre el tema no existe una aproximación que sea válida en general, por lo que lo usual es utilizar una estimación del p-valor del contraste. Esto implica un coste computacional muy grande que lo hace poco viable cuando los tamaños muestrales son grandes.

4.2. Criterios basados en la información de Kullback-Leibler

4.2.1. Información de Kullback-Leibler

La información de Kullback-Leibler es una medida de la diferencia entre la función de densidad real f_Ψ y la función de densidad estimada $f_{\hat{\Psi}}$ que podemos definir como sigue.

Definición 4.1. *Dado $\vec{\mathbf{y}}$ un vector aleatorio p-dimensional con función de densidad f_Ψ dependiente de un conjunto de parámetros Ψ se define la información de Kullback-Leibler con respecto a un estimador $\hat{\Psi}$ como*

$$I(f_\Psi, f_{\hat{\Psi}}) = \int f(\vec{y}; \Psi) \log f(\vec{y}; \Psi)\, d\vec{y} - \int f(\vec{y}; \Psi) \log f(\vec{y}; \hat{\Psi})\, d\vec{y}. \tag{4.2}$$

Para conseguir nuestro objetivo de estimar correctamente el número de componentes que mejor se ajuste a una muestra dada buscaremos minimizar la cantidad de la Ecuación (4.2). Observamos que el primer término no depende de la muestra y, por tanto, trataremos de minimizar

$$\eta(\vec{y}, \Psi) = \int f(\vec{\omega}; \Psi) \log f(\vec{y}; \hat{\Psi})\, d\vec{\omega}.$$

Aprovechando que Ψ caracteriza la distribución de f_Ψ podemos reescribir la expresión

anterior como

$$\begin{aligned}\eta(\vec{y}, F) &= \int f(\vec{\omega}; \Psi) \log f(\vec{\omega}; \hat{\Psi})\, d\vec{\omega} \\ &= \int \log f(\vec{\omega}; \hat{\Psi})\, dF(\vec{\omega}),\end{aligned}$$

donde F denota la verdadera función de distribución de f_Ψ.

Uno de los estimadores más sencillos de η es el dado al sustituir la función de distribución real por la función de distribución empírica pues nos permite aprovechar su definición a trozos para escribir

$$\hat{\eta} := \eta(\vec{y}, \hat{F}_n) = \frac{1}{n} \sum_{j=1}^{n} \log f(\vec{y}_j; \hat{\Psi}) = \frac{1}{n} \log L(\vec{y}; \hat{\Psi}),$$

donde L denota la función de verosimilitud asociada a la muestra $\vec{y}$. Como el sesgo del estimador $\hat{\eta}$ es

$$b(F) = \mathbb{E}_F \left[\eta(\vec{y}, \hat{F}_n) - \eta(\vec{y}, F) \right], \tag{4.3}$$

lo que haremos es tomar el estimador insesgado de $\eta(\vec{y}, F)$ dado por $\hat{\eta} - b(F)$. Entonces, para minimizar la información de Kullback-Leibler buscaremos maximizar $\hat{\eta} - b(F)$. Como no es posible obtener una expresión analítica manejable de $b(T)$ hemos de estimarlo, dependiendo del estimador usado obtendremos criterios diferentes. En general, los criterios basados en la información se suelen definir en la literatura a partir de dos veces la diferencia de $\hat{\eta} - b(F)$. En este trabajo seguiremos la misma convención.

Definición 4.2. *Llamaremos criterio basado en la información a cualquier criterio que consista en elegir el modelo que minimice la diferencia*

$$-\frac{2}{n} \log L(\hat{\Psi}) + 2C.$$

Observamos que se sustituye el sesgo por una constante C que representa la penalización por complejidad y que se obtiene a partir de algún estimador de $b(F)$. Veamos entonces los diferentes criterios que han surgido bajo esta perspectiva.

4.2.2. Criterio de Akaike (AIC)

El criterio de Akaike se basa en los resultados obtenidos en [1] y [2] en los que prueba que $b(F)$ converge asintóticamente a d/n donde d es el número de parámetros independientes del modelo al que estemos ajustando y n el tamaño muestral. En el caso de las

mixturas finitas de distribuciones normales heterocedásticas $d = 3k - 1$ donde k es el número de componentes de la mixtura, de igual forma en el caso de mixturas de distribuciones exponenciales $d = 2k - 1$. A partir de dicho resultado el criterio de Akaike (AIC) consiste en elegir el modelo que minimice

$$\text{AIC} = -2\log\Big(L(\hat{\Psi})\Big) + 2d.$$

Sin embargo, para poder asegurar la convergencia es necesario que se satisfagan ciertas condiciones teóricas que dependen a su vez de las condiciones de regularidad clásicas del estadístico del test de la razón de verosimilitud. Como ya hemos visto en el contraste del número de componentes de una mixtura finita dichas condiciones no se satisfacen. A pesar de ello, el criterio AIC se sigue utilizando en este contexto aunque también tiende a sobrestimar el número de componentes como veremos más adelante.

Para analizar la eficacia del método utilizaremos las muestras que hemos utilizado en el capítulo anterior. Comenzaremos analizando las mixturas de distribuciones normales, así las puntuaciones que asigna este método a cada uno de los grupos quedan recogidas en las Tablas 4.1 y 4.2 donde la negrita indica el número de grupos teóricos y el número de grupos elegido por el método, esta convención se seguirá en el resto de tablas de esta sección.

Tamaño Muestral	Número de Componentes				
	1	**2**	3	4	5
50	259.81	202.15	**201.93**	214.15	220.15
100	523.37	**444.01**	446.60	450.33	455.15
200	1041.42	**844.82**	849.64	855.44	858.67
500	2581.76	**2153.85**	2160.07	2165.98	2170.41
1000	5111.37	**4218.55**	4226.53	4231.16	4451.54
2000	10304.30	**8419.35**	8425.62	8429.74	8438.00

Tabla 4.1: Valores de AIC para las muestras generadas a partir de $\mathbf{x}_1$.

Observamos que en el caso de las mixturas de distribuciones normales univariantes el criterio funciona de manera adecuada para todos los tamaños muestrales excepto el más pequeño, en el que sobrestima, en el caso de que las mixturas estén separadas (caso $\mathbf{x}_1$).

Tamaño Muestral	Número de Componentes				
	1	2	**3**	4	5
50	**192.07**	196.96	200.48	206.86	213.04
100	399.22	**395.32**	395.79	400.39	407.46
200	736.42	**713.18**	718.67	720.94	726.66
500	1936.29	1900.79	**1892.30**	1897.75	1905.88
1000	3799.62	3692.86	**3680.60**	3689.65	3695.36
2000	7733.00	7526.01	**7475.44**	7488.30	7505.06

Tabla 4.2: Valores de AIC para las muestras generadas a partir de $\mathbf{x}_2$.

Sin embargo, cuando las componentes están muy superpuestas, segundo caso, necesitamos tamaños muestrales grandes para que el método no infraestime el número de componentes.

4.2.3. Criterio extendido (EIC)

Este criterio surge como una extensión del criterio AIC que no depende de las condiciones teóricas. Desarrollado por Ishiguro, Sakamoto, y Kitagawa en [8] utiliza la técnica bootstrap desarrollada por Efron para estimar el sesgo de la siguiente manera:

Se generan B muestras bootstrap $\vec{y}_b^*$ a partir de la muestra original. Después, para cada una de ellas se calculan las estimaciones máximo verosímiles de sus parámetros $\hat{\Psi}_b^*$. Finalmente se aproxima el sesgo mediante el estimador

$$\hat{b}(F) = \frac{1}{nB}\sum_{b=1}^{B}\left(\log L(\vec{y}_b^*;\hat{\Psi}) - \log L(\vec{y}_b^*;\hat{\Psi}_b^*)\right).$$

En consecuencia el criterio EIC consiste en tomar el modelo que minimice la expresión

$$\text{EIC} = -2\log L(\hat{\Psi}) + 2n\hat{b}(F).$$

El principal inconveniente de este criterio es su alto coste computacional y para tratar de paliarlo proponen como solución reducir la varianza asociada a la simulación. Para ello descomponen la expresión de la Ecuación (4.3) como:

$$\begin{aligned} b(F) &= \mathbb{E}_F\left[\int \log f(\vec{\omega};\hat{\Psi})\, dF(\vec{\omega}) - \frac{1}{n}\log L(\vec{y};\hat{\Psi})\right] \\ &= \mathbb{E}_F\left[D_1(\vec{y};F) + D_2(\vec{y};F) + D_3(\vec{y};F)\right] \end{aligned}$$

definiendo

$$D_1(\vec{y};F) = \log L(\vec{y};\hat{\Psi}) - \log L\left(\vec{y};T(F)\right)$$
$$D_2(\vec{y};F) = \log L(\vec{y};T(F)) - \int \log f(\vec{\omega};T(F)\, dF(\vec{\omega})$$
$$D_3(\vec{y};F) = \int \log f(\vec{\omega};T(F)\, dF(\vec{\omega}) - \int \log f(\vec{\omega};\hat{\Psi})\, dF(\vec{\omega}),$$

siendo T el operador que verifica la ecuación $T(\hat{F}_n) = \hat{\Psi}$. Utilizando argumentos que quedan fuera del alcance de este trabajo se puede probar ([9]) que

$$\mathbb{E}_F(D_2(\vec{y};F)) = 0.$$

Por tanto

$$b(F) = \mathbb{E}_F\left[D_1(\vec{y};F) + D_3(\vec{y};F)\right].$$

Además también se prueba que

$$Var\left[D_1(\vec{y};F) + D_2(\vec{y};F) + D_3(\vec{y};F)\right] = O(n^{-1})$$

mientras que

$$Var\left[D_1(\vec{y};F) + D_3(\vec{y};F)\right] = O(n^{-2})$$

por lo que al eliminar el término $D_2(\vec{y};F)$ reducimos en un orden la varianza del sesgo sin comprometer su aproximación.

En definitiva, podemos estimar de manera más precisa, en el sentido de que necesitaremos menos muestreos bootstrap para obtener el mismo grado de error, el sesgo utilizando como estimador

$$\hat{b}_E(F) = \frac{1}{nB}\sum_{b=1}^{B}\left(\log L(\vec{y}_b^*;\hat{\Psi}) - \log L(\vec{y}_b^*;\hat{\Psi}_b^*) + \log L(\vec{y}_b;\hat{\Psi}) - \log L(\vec{y}_b;\hat{\Psi}_b^*)\right).$$

Gracias a esto, los resultados obtenidos mediante bootstrap con este nuevo estimador necesitarán un menor número de iteraciones para obtener la misma precisión que los obtenidos con el estimador inicial. El criterio que utiliza este estimador lo denotaremos por EIC_E donde el subíndice hace referencia a su mayor eficiencia en comparación al criterio EIC. Se define entonces el criterio EIC_E como aquel que elige el modelo que hace mínima la expresión

$$\mathrm{EIC}_E = -2\log L(\hat{\Psi}) + 2n\hat{b}_E(F).$$

4.3. Criterio bayesiano (BIC)

Ya hemos analizado los criterios que surgen a partir de la información de Kullback-Leibler, ahora analizaremos lo que sucede si se adopta un enfoque bayesiano. Para ello asumiremos que existe $p(\Psi)$ distribución a priori de Ψ y compararemos los modelos a partir de la verosimilitud integrada o evidencia.

Definición 4.3. *Dado un vector aleatorio $\vec{\mathbf{x}}$ con función de densidad f_θ dependiente de un parámetro θ que sigue una distribución a priori $p(\theta)$ y una muestra proveniente de $\vec{\mathbf{x}}$, $\vec{x}$ se define la verosimilitud integrada, también conocida como verosimilitud marginal o evidencia, $p(\vec{x})$ de la siguiente manera*

$$p(\vec{x}) = \int p(\theta, \vec{x})\, d\theta, \tag{4.4}$$

donde $p(\theta, \vec{x}) := p(\theta)L(\vec{x}; \theta)$.

La evidencia no deja de ser una medida de cómo de verosímil es una muestra bajo cierto modelo, por tanto, buscaremos maximizar dicha medida. Como no podemos resolver la anterior integral pues ni siquiera conocemos la distribución a priori la estrategia más usual es aproximarla utilizando el método de Lagrange:

Supongamos que $\dim(\Psi) = d$, comenzaremos reescribiendo y adaptando a nuestro problema la integral (4.4)

$$\begin{aligned} p(\vec{x}) &= \int p(\Psi, \vec{x})\, d\Psi = \\ &= \int \exp\{\log p(\Psi, \vec{x})\, d\Psi\}, \end{aligned}$$

con el objetivo de encontrar una buena aproximación. Sea $\bar{\Psi}$ la moda a posteriori, por definición sabemos que $\bar{\Psi}$ satisface que

$$\left(\frac{\partial p(\Psi, \vec{x})}{\partial \Psi}\right)_{\bar{\Psi}} = \vec{0},$$

por tanto gracias a la inyectividad del logaritmo

$$\left(\frac{\partial \log p(\Psi, \vec{x})}{\partial \Psi}\right)_{\bar{\Psi}} = \vec{0}.$$

En virtud del Teorema de Taylor podemos aprovechar que la primera derivada parcial se anula para aproximar

$$\log p(\Psi, \vec{x}) \approx \log p(\bar{\Psi}, \vec{x}) - \frac{1}{2}(\Psi - \bar{\Psi})' H(\bar{\Psi})(\Psi - \bar{\Psi}),$$

siendo $H(\bar{\Psi})$ la menos matriz hessiana de $\log p(\Psi, \vec{x})$ evaluada en $\bar{\Psi}$. Sustituyendo en la integral

$$\begin{aligned} p(\vec{x}) &\approx \int \exp\left\{\log p(\bar{\Psi}, \vec{x}) - \frac{1}{2}(\Psi - \bar{\Psi})' H(\bar{\Psi})(\Psi - \bar{\Psi})\right\} d\Psi \\ &= p(\bar{\Psi}, \vec{x}) \int \exp\left\{-\frac{1}{2}(\Psi - \bar{\Psi})' H(\bar{\Psi})(\Psi - \bar{\Psi})\right\} d\Psi. \end{aligned}$$

Ahora observamos que $H(\bar{\Psi})$ es una matriz de varianzas-covarianzas, es decir, que es semidefinida positiva, pues es simétrica por definición, lo cual se cumple pues $\bar{\Psi}$ es un máximo local, por tanto, la matriz hessiana evaluada en dicho punto es semidefinida negativa y como estamos considerando la matriz hessiana con signo negativo deducimos que $H(\bar{\Psi})$ es semidefinida positiva. Entonces la integral se convierte en la integral de la función de densidad de una normal multidimensional con vector de medias $\bar{\Psi}$ y matriz de varianzas-covarianzas $H^{-1}(\bar{\Psi})$ salvo constantes multiplicativas. En efecto,

$$\begin{aligned} &\int \exp\left\{-\frac{1}{2}(\Psi - \bar{\Psi})' H(\bar{\Psi})(\Psi - \bar{\Psi})\right\} d\Psi \\ &= (2\pi)^{d/2} |H(\bar{\Psi})|^{-1/2} \int \frac{1}{(2\pi)^{d/2} |H(\bar{\Psi})|^{-1/2}} \exp\left\{-\frac{1}{2}(\Psi - \bar{\Psi})' H(\bar{\Psi})(\Psi - \bar{\Psi})\right\} d\Psi \\ &= (2\pi)^{d/2} |H(\bar{\Psi})|^{-1/2}. \end{aligned}$$

Por tanto, podemos aproximar la logverosimilitud integrada como

$$\begin{aligned} \log p(\vec{x}) &\approx \log p(\bar{\Psi}, \vec{x}) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|H(\bar{\Psi})| \\ &= \log L(\bar{\Psi}) + \log p(\bar{\Psi}) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|H(\bar{\Psi})|. \end{aligned}$$

A pesar de todo, esta expresión sigue sin ser manejable, para remediarlo sustituiremos la moda a posteriori por el estimador máximo verosímil $\hat{\Psi}$ y en consecuencia sustituiremos $H(\bar{\Psi})$ por la matriz de cantidad de información de Fisher, $I(\vec{x}, \Psi)$. Para justificar este cambio es necesario imponer que la distribución a priori $p(\Psi)$ sea uniforme, es decir, que no sea informativa para que la primera derivada parcial se siga anulando al cambiar el punto de evaluación de $\bar{\Psi}$ a $\hat{\Psi}$. Seguidamente, se utiliza que bajo las condiciones clásicas de regularidad

$$|I(\vec{x}, \Psi)| = O(n^d)\,,$$

donde n es el tamaño muestral. Finalmente se eliminan los términos que no dependen de n de la expresión

$$\log L(\hat{\Psi}) + \log p(\hat{\Psi}) + \frac{d}{2}\log(2\pi) - \frac{d}{2}\log(n) - \frac{1}{2}\log(c)$$

donde $c \in \mathbb{R}$ es una constante, lo que resulta en

$$\log L(\hat{\Psi}) - \frac{d}{2}\log(n).$$

Para construir el criterio basta recordar que buscábamos maximizar la evidencia para la muestra dada, por lo que el criterio BIC consiste en elegir el modelo que minimice

$$\text{BIC} = -2\log L(\hat{\Psi}) + d\log(n).$$

Sin embargo, durante el desarrollo teórico del criterio hemos utilizado que se cumplían las condiciones clásicas de regularidad, a pesar de que no se verifican en nuestro contexto, y que podíamos asumir que la distribución a priori no era informativa (para más información de por qué está asunción no es baladí ver [17] sección 2.6). Es decir, el criterio pierde parte de su justificación teórica; a pesar de ello, numerosos estudios ([7],[4], [5]) avalan el uso de este método en el contexto de las mixturas finitas.

A continuación analizaremos la bondad del método haciendo uso nuevamente de los datos del capítulo anterior que ya han sido usados para el criterio de Akaike. Los resultados se encuentran en las Tablas 4.3 y 4.4, en las cuales podemos observar un comportamiento muy similar al de Akaike en el caso en el que las componentes están separadas eliminando, sin embargo, el sobreajuste con el tamaño muestral más pequeño. Por otra parte, las respuestas en el otro caso son también muy similares a las del criterio de Akaike pues nos encontramos con una subestimación en el caso de que el tamaño muestral sea pequeño y con una buena estimación en cuanto el tamaño muestral es grande.

Tamaño Muestral	Número de Componentes				
	1	**2**	3	4	5
50	263.63	**211.71**	217.13	235.18	240.48
100	528.58	**457.04**	477.69	478.98	492.50
200	1048.01	**861.31**	876.02	891.73	907.74
500	2590.19	**2174.93**	2193.29	2212.26	2333.35
1000	5121.18	**4243.09**	4262.85	4282.77	4520.25
2000	10315.50	**8447.36**	8468.85	8499.02	8514.23

Tabla 4.3: Valores de BIC para las muestras generadas a partir de $\mathbf{x}_1$.

Tamaño Muestral	Número de Componentes				
	1	2	**3**	4	5
50	**195.90**	206.52	215.86	227.48	228.62
100	**404.43**	408.37	416.73	429.90	428.74
200	743.01	**729.68**	745.07	757.21	772.81
500	1944.72	1921.87	**1920.10**	1941.45	1964.89
1000	3809.43	**3717.40**	3719.86	3743.86	3770.90
2000	7744.20	7554.01	**7526.54**	7550.25	7585.47

Tabla 4.4: Valores de BIC para las muestras generadas a partir de $\mathbf{x}_2$.

4.4. Criterio basados en la verosimilitud completa (CLC)

Hasta ahora no hemos utilizado en ningún criterio la función de verosimilitud completa en la que nos apoyamos para desarrollar el algortimo EM. Este criterio aprovechará su relación con la función de verosimilitud para determinar qué modelo se ajusta mejor a la muestra dada.

Recuperando la notación utilizada en el capítulo anterior es fácil ver que

$$\log L(\Psi) = \log L_c(\Psi) - \log k(\Psi),$$

donde

$$\log k(\Psi) = \sum_{i=1}^{n}\sum_{j=1}^{k} \mathbb{1}_{z_i=j} \log(\gamma_{ij}).$$

Como ya habíamos visto,

$$\mathbb{E}\left[1_{z_i=j}|\vec{\mathbf{y}} = \vec{y_i}\right] = P(\vec{\mathbf{z}} = \vec{e}_j|\vec{\mathbf{y}} = \vec{y_i}; \Psi_t) = \gamma_{ij}$$

para cada $i \in \{1, \ldots, n\}$ y cada $j \in \{1, \ldots, k\}$, por lo que deducimos que

$$\mathbb{E}\left[\log k(\Psi)|Y\right] = \sum_{i=1}^{n}\sum_{j=1}^{k} \gamma_{ij} \log(\gamma_{ij}).$$

Denotaremos

$$H(\Gamma) = \sum_{i=1}^{n}\sum_{j=1}^{k} \gamma_{ij} \log(\gamma_{ij}),$$

donde Γ es la matriz dada por $\Gamma = (\gamma_{ij})_{i=1,\ldots,n}^{j=1,\ldots,k}$. Recordemos además que el valor de γ_{ij} depende de Ψ, por tanto es lógico tomar como estimador $H(\hat{\Gamma})$ donde estamos denotando por $\hat{\Gamma}$ la matriz formada por los γ_{ij} obtenidos a partir del estimador máximo verosímil $\hat{\Psi}$. En definitiva, el criterio CLC busca maximizar la función de verosimilitud completa, es decir el criterio CLC consiste en elegir el modelo que minimice

$$\text{CLC} = -2 \log L(\hat{\Psi}) + 2H(\hat{\Gamma}).$$

Se podría suponer que al utilizar la función de verosimilitud completa en vez de la verosimilitud tradicional estamos ante un criterio que funciona mejor que los anteriores. Sin embargo, a la vista de los resultados recogidos en la Tabla 4.5 tiene un peor desempeño que los anteriores cuando las componentes están superpuestas infraestimando el número de componentes con cualquier tamaño muestral. La explicación de este fenómeno la encontramos en la forma de penalización que tiene el criterio pues al penalizar utilizando los γ_{ij} lo que ocurre cuando las componentes no están separadas es que la asignación de una observación a una componente no está clara, es decir, la penalización no va a ser grande lo que permite compensar la pérdida de verosimilitud al simplificar el modelo.

En definitiva, en base a los resultados, es válido para analizar mixturas cuyas componentes estén lo suficientemente separadas como para no superponerse pero no es útil para analizar mixturas con componentes superpuestas

Tamaño Muestral	Número de Componentes				
	1	**2**	3	4	5
50	255.81	**192.15**	224.42	243.92	261.46
100	519.37	**434.28**	473.38	526.66	663.55
200	1037.42	**836.10**	953.52	1004.46	1132.45
500	2577.76	**2147.50**	2466.95	2751.84	2961.70
1000	5107.37	**4218.14**	4859.42	5562.64	5969.18
2000	10300.30	**8420.57**	9772.27	10514.96	11945.46

Tabla 4.5: Valores de CLC para las muestras generadas a partir de $\mathbf{x}_1$.

Tamaño Muestral	Número de Componentes				
	1	2	**3**	4	5
50	**188.07**	226.30	247.44	282.48	306.70
100	**395.22**	478.75	534.29	603.18	584.16
200	**732.42**	903.05	1048.52	1093.80	1192.88
500	**1932.29**	2317.66	2738.76	3062.64	3314.08
10000	**3795.62**	4574.39	5411.41	6042.68	6542.25
2000	**7729.00**	9194.88	10893.18	12161.04	13145.78

Tabla 4.6: Valores de CLC para las muestras generadas a partir de $\mathbf{x}_2$.

4.5. Resumen

A la vista de lo expuesto anteriormente nos damos cuenta de lo difícil que es desarrollar criterios para conseguir determinar de manera consistente el número de componentes de una mixtura. Hemos necesitado unos desarrollos muy laboriosos y aun así no somos capaces de determinar de manera eficaz el número de componentes cuando están muy superpuestas.

Capítulo 5

Aplicaciones: Análisis clúster

En este capítulo exploraremos algunas de las muchas aplicaciones que pueden tener las mixturas finitas las cuales son de utilidad en diversos campos científicos. Prestaremos especial atención al análisis clúster utilizando un modelo de mixturas finitas.

5.1. Análisis clúster

El análisis clúster es la idea que da origen a las mixturas cuando Pearson se planteó utilizar dicho modelo para modelar una población de cangrejos ([15]). En el segundo capítulo hemos visto como a partir de la estimación máximo verosímil y el algoritmo EM conseguíamos estimadores para los paramétros de la población e incluso conseguíamos un estimador de la probabilidad de que un individuo i perteneciese a cierto grupo, o componente de la mixtura, j y la denotamos por γ_{ij}. Más aún, en el capítulo anterior analizamos diversos métodos para determinar el número más adecuado de componentes para un conjunto de datos. Gracias a esto estamos en posición de analizar un conjunto de datos reales y comprobar la eficacia de nuestro modelo. Los datos en cuestión están extraídos de [14] y recogen la altura, peso y sexo de diferentes personas.

Asumiremos que las variables peso y altura se distribuyen cada una según una mixtura de dos normales unidimensionales y nuestro objetivo será estimar los parámetros de cada una de ellas y clasificar a los individuos de la población en cada uno de los dos grupos. Al finalizar compararemos los resultados obtenidos con el algoritmo EM y los resultados reales. Comenzamos reflejando los resultados para la variable altura en las Tabla 5.1. La estimación para esta variable es muy buena pues los parámetros estimados son muy

similares a los parámetros muestrales. Además el error cometido en la asignación de las observaciones (Tabla 5.2) es de aproximadamente el 15 % en ambos grupos.

A continuación, se presentan los resultados para la variable peso en las Tabla 5.3. Para esta variable las estimaciones no son tan precisas como en el caso anterior, sin embargo siguen siendo buenas. A pesar de esto, el error cometido al clasificar las observaciones es muy alto en el caso del grupo de los hombres en los que cometemos un error del 42 % aunque en el grupo de las mujeres conseguimos un error del 20 % la asignación no es buena.

El peor desempeño del algoritmo con la variable peso lo podemos explicar atendiendo a cómo se distribuye la variable en cada sexo. En la Figura 5.1 observamos que existe una mayor superposición, con respecto al sexo, en la variable peso que en la variable altura y, como hemos visto anteriormente, esto explica el mejor desempeño al utilizar la variable altura.

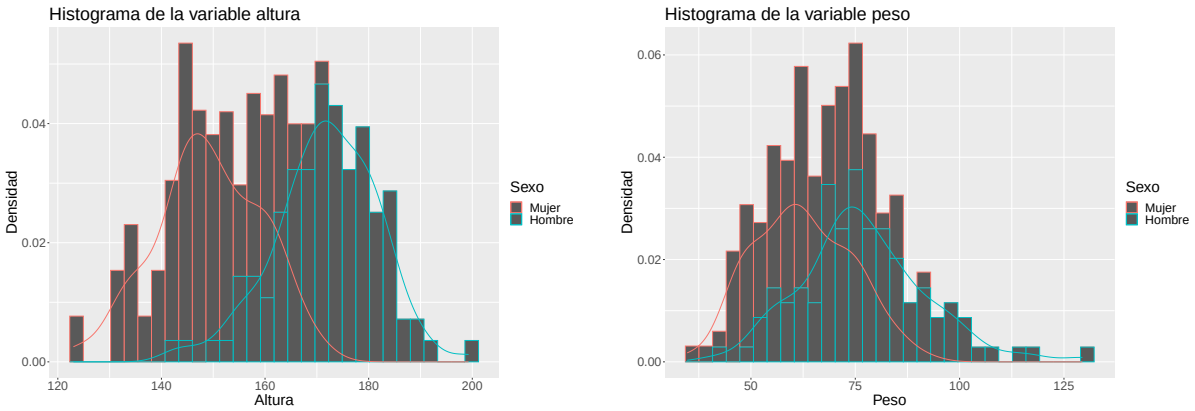

Figura 5.1: Histograma de las variables altura y peso, respectivamente, diferenciadas según la variable sexo y con la función de densidad estimada superpuesta.

Parámetro	**Componente 1**	**Componente 2**	**Mujer**	**Hombre**
Peso	0.4983	0.5017	0.4829	0.5171
Media	149.21	172.46	149.00	172.00
Desviación Típica	10.32	9.01	10.20	9.98

Tabla 5.1: Valores muestrales de pesos, medias y desviaciones típicas y valores estimados por el algoritmo EM para la variable altura.

Sin embargo, en el análisis anterior hemos analizado ambas variables por separado obviando cualquier relación existente entre ellas, perdiendo, por tanto, la información

	Predicción 1	Predicción 2
Mujer	84.85	15.15
Hombre	13.21	86.79

Tabla 5.2: Matriz de confusión para la clasificación de las observaciones de la variable altura.

Parámetro	Componente 1	Componente 2	Mujer	Hombre
Peso	0.4954	0.5046	0.4829	0.5171
Media	75.71	64.24	62.4	77.0
Desviación Típica	16.74	11.45	11.8	15.2

Tabla 5.3: Valores muestrales de pesos, medias y desviaciones típicas y valores estimados por el algoritmo EM para la variable peso.

que nos pudiera aportar dicha relación. Para incluirla podemos modelar la distribución conjunta de la altura y el peso como una mixtura de dos distribuciones normales 2-dimensionales y contrastar las diferencias. Los resultados obtenidos con este enfoque están recogidos en las Tablas 5.5 y 5.6.

Observamos que tanto la aproximación como la clasificación obtenida a partir del modelo bidimensional ofrece mejores resultados que la obtenida a partir de los modelos unidimensionales evidenciando la utilidad de considerar ambas variables a la vez.

En definitiva, hemos visto que las mixturas finitas son una herramienta muy útil para el análisis clúster que permite modelar fenómenos con una gran flexibilidad adaptándose a las características del mismo. En este caso, las hemos utilizado para modelar tanto un vector aleatorio como dos variables aleatorias obteniendo resultados aceptables en ambos casos. A pesar de todo, hemos visto como en el caso de que las componentes se encuentren superpuestas la capacidad de clasificación se ve disminuida, lo cual pudimos solucionar al considerar las dos variables de manera conjunta.

5.2. Otras aplicaciones

La flexibilidad que tiene el modelo de las mixturas le permite tener aplicaciones fuera del análisis clúster. Una de ellas es su importancia en las redes neuronales conformadas a

	Predicción 1	**Predicción 2**
Mujer	19.19	80.81
Hombre	57.55	42.45

Tabla 5.4: Matriz de confusión para la clasificación de las observaciones de la variable peso.

Componente	Pesos	Media	Matriz varianzas-covarianzas
1	0.4962	$\begin{pmatrix} 172.44 \\ 76.99 \end{pmatrix}$	$\begin{pmatrix} 83.65 & 15.69 \\ 15.69 & 239.81 \end{pmatrix}$
2	0.5038	$\begin{pmatrix} 149.48 \\ 62.96 \end{pmatrix}$	$\begin{pmatrix} 110.81 & 10.92 \\ 10.92 & 138.35 \end{pmatrix}$

Tabla 5.5: Valores de pesos, medias y matrices de varianzas-covarianzas estimadas por el algoritmo EM.

partir de funciones con base radial en las cuales se pueden modelar los datos de entrada mediante una mixtura lo que permite obtener las funciones a partir de los estimadores máximo verosímiles de la mixtura ([3] Sección 2.6).

Además, como hemos observado en el primer capítulo la densidad de una mixtura puede tener formas muy variadas, esto combinado con que permite tener en cuenta las posibles estructuras de la muestra hace de las mixturas un gran punto intermedio entre la inferencia paramétrica y la estimación tipo núcleo para estimar la función de densidad.

	Componente 1	**Componente 2**
Mujer	11.11	88.89
Hombre	86.79	13.21

Tabla 5.6: Matriz de confusión para la clasificación de las observaciones del vector conjunto peso y altura.

Capítulo 6

Conclusiones

En este trabajo hemos hecho un estudio del modelo de las mixturas finitas centrándonos principalmente en la aplicación de la inferencia estadística a este modelo, en la determinación del número de grupos dentro de una población y de la aplicación del modelo a los datos.

Los resultados obtenidos constatan el buen funcionamiento del modelo para el análisis clúster y, en particular, el buen desempeño del algoritmo EM y de la estrategia de inicio del mismo. Por otro lado, también muestran la dificultad para trabajar con conjuntos de datos donde los grupos se encuentren muy solapados. En esta línea, van también los resultados obtenidos para los criterios para determinar el número de grupos.

En resumen, las mixturas finitas son un concepto muy potente para el análisis de datos suponiendo un avance significativo al modelo paramétrico.

Bibliografía

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In BN Petrov and F Csaki, editors, *Proceedings of the 2nd International Symposium on Information Theory*, pages 267–281, Budapest, 1973. Akademiai Kiado.

[2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[3] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[4] J.G Campbell, C Fraley, F Murtagh, and A.E Raftery. Linear flaw detection in woven textiles using model-based clustering. *Pattern Recognition Letters*, 18(14):1539–1548, 1997.

[5] A. Dasgupta and A. Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93:294–, 04 1998.

[6] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 1977.

[7] Chris Fraley and Adrian Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[8] M. Ishiguro, Y. Sakamoto, and G. Kitagawa. Bootstrapping log likelihood and EIC, an extension of AIC. *Annals of the Institute of Statistical Mathematics*, 49:411–434, 1997.

[9] S. Konishi and G. Kitagawa. Generalised information criteria in model selection. *Biometrika*, 83(4):875–890, 1996.

[10] J. S. Marron and M. P. Wand. Exact Mean Integrated Squared Error. *The Annals of Statistics*, 20(2):712 – 736, 1992.

[11] G. J. McLachlan. On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(3):318–324, 1987.

[12] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, New York, 1997.

[13] Geoffrey McLachlan and David Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience, New York, 2000.

[14] Saran Pannasuriyaporn. Male female height and weight. `https://www.kaggle.com/datasets/saranpannasuriyaporn/male-female-height-and-weight`, 2019. Consultado: 2024-07-03.

[15] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London A*, 185, 1894.

[16] REAL ACADEMIA ESPAÑOLA. Diccionario de la lengua española, 23.ª ed., [versión 23.7 en línea]. `https://dle.rae.es`, 2024. [Consultado: 03/02/2024].

[17] Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[18] J. H. Wolfe. A Monte Carlo study of the sampling distribution of the likelihood ratio for mixtures of multinormal distributions. Technical Bulletin STB 72-2, U.S. Naval Personnel and Training Research Laboratory, San Diego, 1971.

Apéndice A

Código de R utilizado para la elaboración del trabajo

```r
# Implementación del algoritmo EM para mixturas de densidades normales
    unidimensionales
library(parallel)
library(MASS)
library(tidyverse)
mvmix <- function(n,p,mu,sigma){
  muestra <- matrix(rep(0,n*length(mu[[1]])),ncol = length(mu[[1]]))
  for (i in 1:n){
    u <- runif(1)
    for(j in 1:length(p)){
      if (between(u,sum(p[1:j])-p[j],sum(p[1:j])))
        muestra[i,] <- mvrnorm(1,mu[[j]],sigma[[j]])
    }
  }
  return(muestra)
}
mixE <- function(n,p,lambda){
  muestra <- rep(0,n)
  for (i in 1:n){
    u <- runif(1)
    for(j in 1:length(p)){
      if (between(u,sum(p[1:j])-p[j],sum(p[1:j])))
        muestra[i] <- rexp(1,lambda[j])
    }
```

```
  }
  return(muestra)
}
mix <- function(n,p,mu,sigma){
  muestra <- rep(0,n)
  for (i in 1:n){
    u <- runif(1)
    for(j in 1:length(p)){
      if (between(u,sum(p[1:j])-p[j],sum(p[1:j])))
        muestra[i] <- rnorm(1,mu[[j]],sigma[[j]])
    }
  }
  return(muestra)
}
verosimilitud <- function(x, pi, mu, sigma) {
  if (length(sigma) != length(mu)) {
    sigma <- rep(sigma, length(mu))
  }
  sum(log(rowSums(mapply(function(pi,mu, sigma) pi * dnorm(x, mean = mu
      , sd = sigma),pi, mu, sigma))))
}
gamma <- function(x,pi,mu,sigma){

  aux <- pi * mapply(function(mu, sigma) dnorm(x, mean = mu, sd = sigma
      ),mu, sigma)

  return(aux/rowSums(aux))
}

inicio_aleatorio <- function(x,g,m, iter = 10){
  pi <- lapply(split(runif(m*g,0,1),rep(1:m,each = g)),function(y) y/
      sum(y))
  mu <- split(runif(m*g,min(x),max(x)),rep(1:m,each = g))
  sigma <- split(runif(m*g,0,(max(x)-min(x))^2),rep(1:m,each = g))
  veros <- mapply(function(pi,mu,sigma) EM_iter(x,pi,mu,sigma,iter)$
      verosimilitud, pi = pi, mu = mu, sigma = sigma)
  inicio <- unname(which.max(veros))
  return(list(pi = pi[[inicio]], mu = mu[[inicio]], sigma = sigma[[
      inicio]]))
```

```
}

EM_iter <- function(x,pi,mu,sigma,iter){

  for (i in (1:iter)){
    vgamma <- gamma(x,pi,mu,sigma)
    pi <- apply(vgamma,2,mean)
    mu <- as.vector(x%*%vgamma/apply(vgamma,2,sum))
    for(k in 1:length(pi)){
      sigma[k] <- sqrt((((x-mu[k])^2%*%vgamma)[,k])/colSums(vgamma)[k])
    }
  }
  return(list(pesos = pi,media = mu, dt = sigma, verosimilitud =
     verosimilitud(x,pi,mu,sigma)))
}
EM_fijo <- function(x,pi,mu,sigma,tol){
  v_new <- 0
  v_old <- tol + 1
  lista_veros <- verosimilitud(x,pi,mu,sigma)
  while(abs(v_new - v_old) > tol){
    v_old <- verosimilitud(x,pi,mu,sigma)
    vgamma <- gamma(x,pi,mu,sigma)
    pi <- apply(vgamma,2,mean)
    mu <- as.vector(x%*%vgamma/apply(vgamma,2,sum))
    for(k in 1:length(pi)){
      sigma[k] <- sqrt((((x-mu[k])^2%*%vgamma)[,k])/colSums(vgamma)[k])
      sigma[is.nan(sigma)] <- 0
    }
    v_new <- verosimilitud(x,pi,mu,sigma)
    lista_veros <- c(lista_veros, v_new)
    if(is.nan(v_new)||is.nan(v_old)){
      warning("Algún peso es 0. Reducir grupos")
      break
    }

  }
  return(list(pesos = pi,media = mu, dt = sigma, verosimilitud = lista_
     veros[length(lista_veros)], verosimilitudes = lista_veros, gamma =
      vgamma ))
```

```
}

EM_normales <- function(x,g,m,tol){

  raiz_aleatoria <- inicio_aleatorio(x,g,m)
  return(EM_fijo(x,raiz_aleatoria$pi,raiz_aleatoria$mu,raiz_aleatoria$
      sigma,tol))

}

# Implementación del algoritmo EM para mixturas de densidades normales
   multidimensionales #
library(mvtnorm)
verosimilitudM <- function(x, pi, mu, sigma) {
  densidades <- mapply(function(pi, mu, sigma) {
    pi * dmvnorm(x, mu, sigma)
  }, pi, mu, sigma, SIMPLIFY = FALSE)
  sum(log(rowSums(do.call(cbind, densidades))))
}

gammaM <- function(x,pi,mu,sigma){

  aux <- pi * mapply(function(mu, sigma) dmvnorm(x, mu, sigma),mu,
      sigma)

  return(aux/rowSums(aux))
}
mu_aux <- function(x, post) {
  (colSums(x * post) / sum(post))
}
EM_iterM <- function(x,pi,mu,sigma,iter){
  for (i in (1:iter)){
    vgamma <- gammaM(x,pi,mu,sigma)
    pi <- apply(vgamma,2,mean)
    mu <- lapply(1:length(pi), function(p) {
      mu_aux(x, vgamma[,p])
    })
    for (k in 1:length(pi)){
      aux <- lapply(1:nrow(x), function(i) {
```

```
130         diff <- x[i, ] - mu[[k]]
131         vgamma[i, k] * (diff %*% t(diff))
132       })
133
134       sigma_sum <- Reduce("+", aux)
135       sigma[[k]] <- sigma_sum / colSums(vgamma)[k]
136       sigma[[k ]] <- sigma[[k]] + 10^-6*diag(ncol(x))
137
138
139     }
140   }
141   return(list(pesos = pi,media = mu, dt = sigma,gamma = vgamma,
        verosimilitudM = verosimilitudM(x,pi,mu,sigma)))
142 }
143
144 inicio_aleatorioM <- function(x,g,m, iter = 10){
145   n <- ncol(x)
146   pi <- lapply(split(runif(m * g, 0, 1), rep(1:m, each = g)), function(
        y) y / sum(y))
147   mu <- lapply(1:m, function(i) {
148     split(runif(g * n, min(x), max(x)), rep(1:g, each = n))
149   })
150   sigma <- lapply(1:m, function(i) {
151     lapply(1:g, function(j) {
152       mat <- matrix(runif(n * n, 0, (max(x) - min(x))^2), n, n)
153       var_mat <- (mat + t(mat)) / 2
154       var_mat <- var_mat + 10^-6*diag(n)
155
156     })
157   })
158   veros <- mapply(function(pi,mu,sigma) EM_iterM(x,pi,mu,sigma,iter)$
        verosimilitudM, pi = pi, mu = mu, sigma = sigma)
159   inicio <- unname(which.max(veros))
160   return(list(pi = pi[[inicio]], mu = mu[[inicio]], sigma = sigma[[
        inicio]]))
161 }
162
163
164 EM_fijoM <- function(x,pi,mu,sigma,tol){
```

```
165   v_new <- 0
166   v_old <- tol + 1
167   while(abs(v_new - v_old) > tol){
168     v_old <- verosimilitudM(x,pi,mu,sigma)
169     vgamma <- gammaM(x,pi,mu,sigma)
170     pi <- apply(vgamma,2,mean)
171     mu <- lapply(1:length(pi), function(p) {
172       mu_aux(x, vgamma[,p])
173     })
174     for (k in 1:length(pi)){
175       aux <- lapply(1:nrow(x), function(i) {
176         diff <- x[i, ] - mu[[k]]
177         vgamma[i, k] * (diff %*% t(diff))
178       })
179 
180       sigma_sum <- Reduce("+", aux)
181 
182       sigma[[k]] <- sigma_sum / colSums(vgamma)[k]
183 
184       v_new <- verosimilitudM(x,pi,mu,sigma)
185       if(is.nan(v_new)||is.nan(v_old)){
186         warning("Algún peso es 0. Reducir grupos")
187         break
188       }
189     }
190   }
191   return(list(pesos = pi,media = mu, dt = sigma, verosimilitudM =
        verosimilitudM(x,pi,mu,sigma),gamma = vgamma))
192 }
193 EM_normalesM <- function(x,g,m,tol){
194 
195   raiz_aleatoria <- inicio_aleatorioM(x,g,m)
196   return(EM_fijoM(x,raiz_aleatoria$pi,raiz_aleatoria$mu,raiz_aleatoria$
        sigma,tol))
197 
198 }
199 
200 ## Implementación del algoritmo EM para mixturas de densidades
      exponenciales ##
```

```
verosimilitudE <- function(x, pi, lambda) {

  aux <- sum(log(rowSums(mapply(function(pi,lambda) pi * dexp(x, rate
     = lambda),pi,lambda))))
}
gammaE <- function(x,pi,lambda){

  aux <- pi * mapply(function(lambda) dexp(x, rate = lambda),lambda)

  return(aux/rowSums(aux))
}
EM_iterE <- function(x,pi,lambda,iter){
  for (i in (1:iter)){
    vgamma <- gammaE(x,pi,lambda)
    pi <- apply(vgamma,2,mean)
    lambda <- as.vector(apply(vgamma,2,sum)/x%*%vgamma)

  }
  return(list(pesos = pi,lambda = lambda, verosimilitud =
     verosimilitudE(x,pi,lambda)))
}
EM_fijoE <- function(x,pi,lambda,tol){
  v_new <- 0
  v_old <- tol + 1
  while(abs(v_new - v_old) > tol){
    v_old <- verosimilitudE(x,pi,lambda)
    vgamma <- gammaE(x,pi,lambda)
    pi <- apply(vgamma,2,mean)
    lambda <- as.vector(colSums(vgamma)/x%*%vgamma)

    v_new <- verosimilitudE(x,pi,lambda)
    if(is.nan(v_new)||is.nan(v_old)){
      warning("Algún peso es 0. Reducir grupos")
      break
    }

  }
  return(list(pesos = pi,lambda = lambda, verosimilitud =
     verosimilitudE(x,pi,lambda),gamma = vgamma))
```

```
}
inicio_aleatorioE <- function(x,g,m, iter = 10, max_restarts = 20){
  pi <- lapply(split(runif(m*g,0,1),rep(1:m,each = g)),function(y) y/
     sum(y))
  lambda <- split(runif(m*g,1/max(x),1/(min(x)+10^-6)),rep(1:m,each = g
     ))
  veros <- mapply(function(pi,lambda) EM_iterE(x,pi,lambda,iter)$
     verosimilitud, pi = pi, lambda = lambda)
  inicio <- unname(which.max(veros))
  if (all(is.na(veros))) {
    if (max_restarts > 0) {
      print(paste("No se puede encontrar un punto de inicio válido.
         Reiniciando el algoritmo. Restan", max_restarts, "reintentos."
         ))
      return(inicio_aleatorioE(x, g, m, max_restarts = max_restarts -
         1, iter))
    } else {
      stop("Se ha alcanzado el número máximo de reinicios sin éxito.")
    }
  } else {
    return(list(pi = pi[[inicio]], lambda = lambda[[inicio]]))
  }
}

EM_exponencial <- function(x,g,m,tol){
  raiz_aleatoria <- inicio_aleatorioE(x,g,m)
  return(EM_fijoE(x,raiz_aleatoria$pi,raiz_aleatoria$lambda,tol))
}

## Ejemplos aportados en el Capítulo 2 ##
x1 <- function(x){
  medias <- 3 * ((2/3)^(0:8) - 1) - 1
  sds <- (2/3)^(0:8)
  sapply(x, function(xi) sum((1/8) * dnorm(xi, mean = medias, sd = sds)
     ))
}
x2 <- function(x){
  0.5*dnorm(x,-1,2/3) + 0.5*dnorm(x,1,2/3)
}
```

```r
x3 <- function(x){
  0.75*dnorm(x,-1,1) + 0.25*dnorm(x,1.50,1/3)
}
x4 <- function(x){
  9/20*dnorm(x,-1,1/3) + 9/20*dnorm(x,1,1/3) + 1/10*dnorm(x,0,1/5)
}
x5 <- function(x){
  medias <- (-2:2)+1/2
  sds <- 2^(-(-2:2))/10
  1/2*dnorm(x,0,1)+sapply(x, function(xi) sum(2^(1-(-2:2))/31 * dnorm(
      xi, mean = medias, sd = sds)))
}
x6 <- function(x){
  medias <- (0:4)/2 - 1
  1/2*dnorm(x,0,1)+sapply(x, function(xi) sum(1/10 * dnorm(xi, mean =
      medias, sd = 1/10)))
}


x <- seq(-5,5,0.1)
ggplot(data = tibble(x), aes(x = x))+
  stat_function(fun = x1)+
  labs(x = "x", y = "Densidad", title = "Función de densidad asimétrica
      ")+
  theme(text = element_text(size = 22))
ggplot(data = tibble(x), aes(x = x))+
  stat_function(fun = x2)+
  labs(x = "x", y = "Densidad", title = "Función de densidad bimodal")+
  theme(text = element_text(size = 22))
ggplot(data = tibble(x), aes(x = x))+
  stat_function(fun = x3)+
  labs(x = "x", y = "Densidad", title = "Función de densidad bimodal
      asimétrica")+
  theme(text = element_text(size = 22))
ggplot(data = tibble(x), aes(x = x))+
  stat_function(fun = x4)+
  labs(x = "x", y = "Densidad", title = "Función de densidad trimodal")

theme(text = element_text(size = 22))
```

```
ggplot(data = tibble(x), aes(x = x))+
  stat_function(fun = x5)+
  labs(x = "x", y = "Densidad", title = "Función de densidad claw asimétrica")+
  theme(text = element_text(size = 22))
ggplot(data = tibble(x), aes(x = x))+
  stat_function(fun = x6)+
  labs(x = "x", y = "Densidad", title = "Función de densidad claw ")+
  theme(text = element_text(size = 22))


## Simulaciones de los ejemplos aportados en los Capítulos 3 y 4 ##

# Mixturas normales univariantes #
set.seed(33)
x1 <- mix(50,c(0.5,0.5),c(3,-3),c(1,1))
x2 <- mix(100,c(0.5,0.5),c(3,-3),c(1,1))
x3 <- mix(200,c(0.5,0.5),c(3,-3),c(1,1))
x4 <- mix(500,c(0.5,0.5),c(3,-3),c(1,1))
x5 <- mix(1000,c(0.5,0.5),c(3,-3),c(1,1))
x6 <- mix(2000,c(0.5,0.5),c(3,-3),c(1,1))
solx1 <- EM_normales(x1,2,100,10^-6)
solx1$pesos; solx1$media; solx1$dt
solx2 <- EM_normales(x2,2,100,10^-6)
solx2$pesos; solx2$media; solx2$dt
solx3 <- EM_normales(x3,2,100,10^-6)
solx3$pesos; solx3$media; solx3$dt


ggplot(data = tibble(x1), aes(x = x1))+
  geom_histogram(aes(y = ..density..), color = "white", bins =21)+
  geom_density()+
  labs(x = "Datos simulados", y = "Densidad",
       title = "Histograma de los datos simulados \ncon n = 50", color = "")+
  theme(text=element_text(size=33))
ggplot(data = tibble(x2), aes(x = x2))+
  geom_histogram(aes(y = ..density..), color = "white", bins =18)+
  geom_density()+
```

```
341   labs(x = "Datos simulados", y = "Densidad",
342        title = "Histograma de los datos simulados \ncon n = 100", color
           = "")+
343   theme(text=element_text(size=33))
344 ggplot(data = tibble(x3), aes(x = x3))+
345   geom_histogram(aes(y = ..density..), color = "white", bins =30)+
346   geom_density()+
347   labs(x = "Datos simulados", y = "Densidad",
348        title = "Histograma de los datos simulados \ncon n = 200", color
           = "")+
349   theme(text=element_text(size=33))
350
351 y1 <- mix(50,c(0.3,0.3,0.4),c(1.5,-1.5,0),c(sqrt(2),sqrt(2),0.5))
352 y2 <- mix(100,c(0.3,0.3,0.4),c(1.5,-1.5,0),c(sqrt(2),sqrt(2),0.5))
353 y3 <- mix(200,c(0.3,0.3,0.4),c(1.5,-1.5,0),c(sqrt(2),sqrt(2),0.5))
354 y4 <- mix(500,c(0.3,0.3,0.4),c(1.5,-1.5,0),c(sqrt(2),sqrt(2),0.5))
355 y5 <- mix(1000,c(0.3,0.3,0.4),c(1.5,-1.5,0),c(sqrt(2),sqrt(2),0.5))
356 y6 <- mix(2000,c(0.3,0.3,0.4),c(1.5,-1.5,0),c(sqrt(2),sqrt(2),0.5))
357 ggplot(data = tibble(y1), aes(x = y1))+
358   geom_histogram(aes(y = ..density..), color = "white", bins = 12)+
359   geom_density()+
360   labs(x = "Datos simulados", y = "Densidad",
361        title = "Histograma de los datos simulados \ncon n = 50", color
           = "")+
362   theme(text=element_text(size=33))
363
364 ggplot(data = tibble(y2), aes(x = y2))+
365   geom_histogram(aes(y = ..density..), color = "white", bins = 17)+
366   geom_density()+
367   labs(x = "Datos simulados", y = "Densidad",
368        title = "Histograma de los datos simulados \ncon n = 100", color
           = "")+
369   theme(text=element_text(size=33))
370 ggplot(data = tibble(y3), aes(x = y3))+
371   geom_histogram(aes(y = ..density..), bins = 11,color = "white")+
372   geom_density()+
373   labs(x = "Datos simulados", y = "Densidad",
374        title = "Histograma de los datos simulados \ncon n = 200", color
           = "")+
```

```
  theme(text=element_text(size=33))
soly1 <- EM_normales(y1,3,100,10^-6)
soly2 <- EM_normales(y2,3,100,10^-6)
soly3 <- EM_normales(y3,3,100,10^-6)

soly1$pesos;soly1$media;soly1$dt
soly2$pesos;soly2$media;soly2$dt
soly3$pesos;soly3$media;soly3$dt

# Mixturas exponenciales #
set.seed(33)
y2 <- mixE(500,c(0.75,0.25),c(1,1/2))
y4 <- mixE(500,c(0.75,0.25),c(1,1/4))
y8 <- mixE(500,c(0.75,0.25),c(1,1/8))
y16 <- mixE(500,c(0.75,0.25),c(1,1/16))
y32 <- mixE(500,c(0.75,0.25),c(1,1/32))
y64 <- mixE(500,c(0.75,0.25),c(1,1/64))
soly2 <- EM_exponencial(y2,2,100,10^-3)
soly4 <- EM_exponencial(y4,2,100,10^-3)
soly8 <- EM_exponencial(y8,2,100,10^-3)
soly16 <-EM_exponencial(y16,2,100,10^-3)
soly32 <-EM_exponencial(y32,2,100,10^-3)
soly64 <-EM_exponencial(y64,2,100,10^-3)
soly2$pesos; soly2$lambda
soly4$pesos; soly4$lambda
soly8$pesos; soly8$lambda
soly16$pesos; soly16$lambda
soly32$pesos; soly32$lambda
soly64$pesos; soly64$lambda

# Mixturas normales univariantes #
set.seed(33)
mu1 <- c(-2,-2)
mu2 <- c(2,2)
mu3 <- c(-2,2)
mu4 <- c(2,-2)
sigma <- diag(2)
z1 <- mvmix(200,c(0.25,0.25, 0.25, 0.25),list(mu1,mu2,mu3,mu4),list(
    sigma,sigma,sigma,sigma))
```

```
413
414 solz1 <- EM_normalesM(z1,4,100,10^-3)
415 solz1$pesos; solz1$media; solz1$dt
416 dfclass <- data.frame(x = z1[,1], y = z1[,2], class = factor(apply(
      solz1$gamma, 1, which.max)))
417 ggplot(data = dfclass, aes(x = x, y = y, color = class)) +
418   geom_point() +
419   labs(title = "Asignación de las observaciones \nen el primer caso",
420        x = "Eje x",
421        y = "Eje y",
422        color = "Componente"
423   )+
424   scale_color_brewer(palette = "Set2")
425 z2 <- mvmix(200,c(0.20,0.20, 0.20, 0.20, 0.20),list(mu1,mu2,mu3,mu4,c
      (0,0)),list(sigma,sigma,sigma,sigma,sigma))
426 solz2 <- EM_normalesM(z2,5,100,10^-3)
427 solz2$pesos; solz2$media; solz2$dt
428 dfclass2 <- data.frame(x = z2[,1], y = z2[,2], class = factor(apply(
      solz2$gamma, 1, which.max)))
429 ggplot(data = dfclass2, aes(x = x, y = y, color = class)) +
430   geom_point() +
431   labs(title = "Asignación de las observaciones \nen el segundo caso",
432        x = "Eje x",
433        y = "Eje y",
434        color = "Componente"
435   )+
436   scale_color_brewer(palette = "Set2")
437
438 ## Implementación de los criterios descritos en el Capítulo 4 para
      mixturas de distribuciones normales unidimensionales ##
439 # Metodo AIC #
440 AIC <- function(x,g){
441   -2*EM_normales(x,g,100,10^-3)$verosimilitud + 6*g-2
442 }
443
444 # Método BIC #
445 BIC <- function(x,g){
446   -2*EM_normales(x,g,100,10^-3)$verosimilitud + (3*g-1)*log(length(x))
447 }
```

```
448
449 # Método CLC #
450 CLC <- function(x,g){
451   aux <- EM_normales(x,g,100,10^-3)
452   return(-2*aux$verosimilitud - 2*sum(aux$gamma*log(aux$gamma),na.rm =
        TRUE))
453
454 }
455
456 # Método EIC #
457 EIC <- function(x,g,nr){
458   psi <- EM_normales(x,g,10,10^-3)
459   boot <- replicate(nr,{
460     xb <- sample(x, replace = TRUE)
461     verosimilitud(xb,psi$pesos,psi$media,psi$dt) - EM_normales(xb,g
          ,10,10^-3)$verosimilitud
462   })
463   return(-2*psi$verosimilitud +2*mean(boot))
464 }
465 EIC_E <- function(x,g,nr){
466   psi <- EM_normales(x,g,10,10^-3)
467   boot <- replicate(nr,{
468     xb <- sample(x, replace = TRUE)
469     psib <- EM_normales(xb,g,10,10^-3)
470     psib$verosimilitud + psi$verosimilitud - verosimilitud(xb,psi$pesos
          ,psi$media,psi$dt) - verosimilitud(x,psib$pesos,psib$media,psib$
          dt)
471   })
472   if (any(is.nan(boot))) {
473     warning("Número de grupos no verosímil.")
474     return(Inf)  # En el caso de que la verosimilitud devuelva NaN (
          debido a que haga logaritmos de números cercanos a 0, es decir
          verosimilitud nula)
475   }
476   return(-2*psi$verosimilitud +2*mean(boot))
477
478 }
479
480 ## Experimentación con los métodos ##
```

```
481
482 sapply(1:5,function(g) AIC(x1,g))
483 sapply(1:5,function(g) AIC(x2,g))
484 sapply(1:5,function(g) AIC(x3,g))
485 sapply(1:5,function(g) AIC(x4,g))
486 sapply(1:5,function(g) AIC(x5,g))
487 sapply(1:5,function(g) AIC(x6,g))
488
489 (sapply(1:5,function(g) AIC(y1,g)))
490 sapply(1:5,function(g) AIC(y2,g))
491 sapply(1:5,function(g) AIC(y3,g))
492 sapply(1:5,function(g) AIC(y4,g))
493 sapply(1:5,function(g) AIC(y5,g))
494 sapply(1:5,function(g) AIC(y6,g))
495
496 (sapply(1:5,function(g) BIC(x1,g)))
497 (sapply(1:5,function(g) BIC(x2,g)))
498 (sapply(1:5,function(g) BIC(x3,g)))
499 (sapply(1:5,function(g) BIC(x4,g)))
500 (sapply(1:5,function(g) BIC(x5,g)))
501 (sapply(1:5,function(g) BIC(x6,g)))
502
503 (sapply(1:5,function(g) BIC(y1,g)))
504 (sapply(1:5,function(g) BIC(y2,g)))
505 (sapply(1:5,function(g) BIC(y3,g)))
506 (sapply(1:5,function(g) BIC(y4,g)))
507 (sapply(1:5,function(g) BIC(y5,g)))
508 (sapply(1:5,function(g) BIC(y6,g)))
509
510 (sapply(1:5,function(g) CLC(x1,g)))
511 (sapply(1:5,function(g) CLC(x2,g)))
512 (sapply(1:5,function(g) CLC(x3,g)))
513 (sapply(1:5,function(g) CLC(x4,g)))
514 (sapply(1:5,function(g) CLC(x5,g)))
515 (sapply(1:5,function(g) CLC(x6,g)))
516
517 (sapply(1:5,function(g) CLC(y1,g)))
518 (sapply(1:5,function(g) CLC(y2,g)))
519 (sapply(1:5,function(g) CLC(y3,g)))
```

```
(sapply(1:5,function(g) CLC(y4,g)))
(sapply(1:5,function(g) CLC(y5,g)))
(sapply(1:5,function(g) CLC(y6,g)))

## Análisis de una base de datos real ##
set.seed(33)
clasificacion <- function(gamma){
  apply(gamma,1, which.max)
}
Test_set <- read_csv("C:/Users/juanc/Downloads/Test set.csv")
Test_set$Sex <- as.factor(Test_set$Sex)
prop.table(table(Test_set$Sex))
sal <- EM_normales(Test_set$Height,2,100,10^-5)
round(sal$media,2)
round(sal$dt,2)
round(sal$pesos,4)
summarise(group_by(Test_set,Sex), mean(Height))
summarise(group_by(Test_set,Sex), sd(Height))
prop.table(table(Test_set$Sex,Test_set$Sex))
clasificacion(sal$gamma)
100*round(prop.table(table(Test_set$Sex,clasificacion(sal$gamma)),1),
    digits = 4)

sal2 <- EM_normales(Test_set$Weight,2,1000,10^-5)
round(sal2$pesos,4)
round(sal2$media,2)
round(sal2$dt,2)
summarise(group_by(Test_set,Sex), mean(Weight))
summarise(group_by(Test_set,Sex), sd(Weight))
clasificacion(sal2$gamma)
100*round(prop.table(table(Test_set$Sex,clasificacion(sal2$gamma)),1),
    digits = 4)
ggplot(data = Test_set, aes(x = Height, color = Sex))+
  geom_histogram(aes(y = ..density..))+
  geom_density()+
  labs(x = "Altura", y = "Densidad", title = "Histograma de la variable
      altura", color = "Sexo")+
  theme(text = element_text(size = 22))
ggplot(data = Test_set, aes(x = Weight, color = Sex))+
```

```
556   geom_histogram(aes(y = ..density..))+
557   geom_density()+
558   labs(x = "Peso", y = "Densidad", title = "Histograma de la variable
          peso", color = "Sexo")+
559   theme(text = element_text(size = 22))
560 set.seed(33)
561 multi <- EM_normalesM(cbind(Test_set$Height,Test_set$Weight)
      ,2,100,10^-3)
562 round(multi$media,2)
563 round(multi$pesos,4)
564 round(multi$dt,2)
565 multiclas <- factor(clasificacion(multi$gamma))
566 100*round(prop.table(table(Test_set$Sex,clasificacion(multi$gamma)),1),
      digits = 4)
567 Test_set$clasi <- multiclas
```