MASTER EN MATEMÁTICA AVANZADA

Modelos basados en clustering mediante mixturas. Aplicación a la mejora de biomarcadores genéticos

Autor
Lucía Rodríguez Ríos

Tutores
Manuel Franco Nicolás
Juana María Vivo Molina

UNIVERSIDAD DE MURCIA
2017/2018

Declaración de originalidad

Lucía Rodríguez Ríos, autora del Trabajo Fin de Máster "*Modelos basados en clustering mediante mixturas. Aplicación a la mejora de biomarcadores genéticos*", bajo la tutela de los profesores Manuel Franco Nicolás y Juana María Vivo Molina, declara que el trabajo que presenta es original, en el sentido de que ha puesto el mayor empeño en citar debidamente todas las fuentes utilizadas.

Para que conste a efectos de la evaluación de este Trabajo Fin de Máster, firma el presente documento.

Murcia, a 10 de septiembre de 2018

Fdo: Lucía Rodríguez Ríos

Índice general

Introducción

Los modelos de mixturas finitas se emplean en multitud de disciplinas, como puede comprobarse en la literatura, con aplicaciones centradas en la modelización de datos sobre el tráfico en internet, para el estudio de la calidad del aire (Gómez, 2014), en la estimación modelos financieros (Ferreira and Garín, 2010), en el estudio de la demanda de atención médica (Atienza, 2003), en el campo de la genética (Delmar et al., 2005; Scharl et al., 2009), etc. Este hecho es debido a que, en muchas ocasiones, no es posible modelar una muestra a partir de una única distribución y es necesario una combinación de ellas. En este sentido, se puede considerar que tales datos provienen de distintos grupos, subpoblaciones o clusters que se corresponden con las componentes de la mixtura y cuya pertenencia a uno u otro grupo se desconoce. Así, por ejemplo en 1894 Karl Pearson planteó una combinación de dos distribuciones univariadas normales para modelizar las dimensiones del caparazón de cangrejos, a partir de una muestra procedente de dos subpoblación, donde de antemano se desconocía la pertenencia de los datos a una u otra subpoblación.

En este trabajo, nos centraremos en la aplicación de los modelos de mixturas para la agrupación o clustering de datos. Este procedimiento se conoce como modelos basados en agrupamiento mediante mixturas, y lo aplicaremos en el diagnóstico de cáncer de ovario, sobre datos de expresión de genes. También cabe señalar que, previamente a dicha aplicación, llevaremos a cabo la selección de los mejores biomarcadores genéticos de la enfermedad estudiada, es decir, aquellos genes cuyas expresiones discriminen mejor entre tejidos ováricos cancerosos y tejidos ováricos sanos.

Con este fin, este trabajo de fin de máster se estructura en cinco capítulos, que a continuación se describen brevemente.

En el primer capítulo, se recogen los principales resultados relativos a modelos de mixturas finitas y cómo se aplican en la clasificación o clustering de datos.

En el segundo capítulo, se explica cómo estimar los parámetros de las mixturas finitas usando la Estimación de Máxima Verosimilitud (MLE) y el algoritmo *Expectation-Maximization* (EM). A continuación, se comentan algunas características propias de dicho algoritmo tales como su convergencia o su fuerte dependencia a los métodos de inicialización usados, y se proponen algunos criterios de selección de modelos, variantes del algoritmo y estrategias usuales de inicialización del mismo. También en este capítulo, se estudia la capacidad de clasificación de los modelos de mixturas introduciendo algunos conceptos re-

lacionados con pruebas-diagnóstico dicotómicas como la sensibilidad, la especificidad, las curvas ROC y algunos índices asociados como el área bajo la curva, el área parcial bajo la curva o el índice ajustado del área parcial.

En el tercer capítulo, se introduce el concepto y aplicaciones de los microarrays genéticos, herramienta que entre otras cosas permite la visualización de las expresiones de muchos genes en una representación matricial. Posteriormente se describe la base de datos de microarray de expresión genética utilizada por Pepe et al. (2003), y se plantean los posibles modelos de clasificación de la muestra según tres criterios distintos de selección de modelos y para cuatro métodos de inicialización. Esta aplicación ha sido desarrollada a través del paquete estadístico del software R llamado FlexMix, que implementa el algoritmo EM para modelos de regresión de mixturas. A continuación, se comentan los resultados obtenidos en relación a los criterios de selección y a los métodos de inicio del algoritmo considerados. Asimismo, se obtienen los parámetros estimados de la mixtura y se estudia el solapamiento entre los grupos o componentes obtenidas. Por último, se estudia la bondad de la clasificación de los datos realizada por los modelos obtenidos, es decir, la sensibilidad, especificidad, valor predictivo global y el área bajo la curva ROC, y la estabilidad de los mismos basada en el remuestreo aleatorio.

En el último capítulo, se realiza una recopilación de los principales resultados del estudio, respondiendo así a las tres preguntas que se plantearon al inicio de este trabajo:

► ¿Cuál es el mejor método de inicialización del algoritmo EM para los datos de nuestro estudio? Y ¿cuál es el mejor criterio de selección del modelo?

► ¿Cuál es el grupo de genes que produce una mejor clasificación de los datos, y que por tanto, se pueden considerar mejores biomarcadores genéticos?

► Y por último, determinar la consistencia y estabilidad del modelo de mixturas de los biomarcadores a través de técnicas de remuestreo.

Los apéndices incluyen el código implementado en R para la obtención de los modelos y las conclusiones.

Capítulo 1

Mixturas finitas para el agrupamiento de datos

Este capítulo se centra en los modelos de mixturas finitas, con el fin de explicar su aplicación al clustering o agrupamiento de datos, por lo que resulta conveniente iniciarlo con una breve introducción sobre la noción de clúster o grupo.

Basándonos en la revisión histórica realizada por McNicholas (2016), la primera referencia sobre modelos de mixturas y clustering data de 1963 de mano de John Harman Wolfe. Este autor define un clúster como "una distribución que se corresponde con una de las componentes de la mixtura". Asimismo discute dos definiciones alternativas de clúster: la primera en la que se considera que cada clúster se corresponde con la moda de una distribución, y la segunda que se centraba en el concepto de similaridad, es decir, define un clúster como "un conjunto de objetos que son más similares entre sí que otros objetos que no pertencen a este conjunto". Este autor comenta que ambas definiciones presentan serias dificultades: con la primera puede ocurrir que haya más modas que componentes reales de la mixtura debido a la existencia de clusters superpuestos o solapados, y con la segunda resulta difícil cuantificar la similaridad entre objetos.

Por otro lado,McNicholas (2016) propone una definición de clúster basada en modelos de mixturas como "una componente unimodal de un modelo de mixturas apropiado". Se dice que el modelo es apropiado si se adapta correctamente a los datos con los que se trabaja. La condición de unimodalidad es importante ya que de no serlo se estaría considerando una distribución errónea de la mixtura o un número insuficiente de componentes.

Aaron D. Gordon propone en 1981 dos características deseables para un clúster: tener cohesión interna y aislamiento externo. Aunque el aislamiento completo no es posible en la mayor parte de los casos prácticos, la idea de cohesión interna afianza el concepto de clúster de McNicholas.

Tras esta pequeña introducción al concepto de clúster, estamos ya en condiciones de estudiar el uso de los modelos de mixturas para el clustering de datos.

1.1. Modelos de mixturas finitas

Los modelos de mixturas integran la presencia de dos o más subpoblaciones dentro de una misma población. Son usados para modelizar el comportamiento estocástico de individuos de una población heterogénea, con el objetivo de analizar e inferir características de interés de los subgrupos homogéneos, sólo a partir de los datos observados de la población completa y sin ninguna información sobre la pertenencia de cada dato a uno u otro grupo. En adelante, las subpoblaciones existentes se corresponden con las componentes de la mixtura y cada una estará definida por una función de distribución y un peso en la mixtura.

Definición 1.1. Sea $\{Y_1, \ldots, Y_n\}$ un conjunto de vectores aleatorios q-dimensionales de una muestra de tamaño n e Y_t un vector $q-$dimensional cuya función de densidad es $f(y_t)$ con $y_t \in \mathbb{R}^q$. Se llama **modelo de mixturas finitas** de K componentes a la distribución de una variable aleatoria Y_t cuya función de densidad puede expresarse como

$$f(y_t|\Psi) = \sum_{i=1}^{K} \pi_i \cdot f_i(y_t|\theta_i) \quad \text{para todo } y_t \in \mathbb{R}^q \tag{1.1}$$

donde $f_i(y_t|\theta_i)$ se corresponde con la densidad de la componente i-ésima de la mixtura de parámetro θ_i, respectivamente; π_i son los pesos de cada componente en la mixtura y $\Psi = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$ es el vector de parámetros desconocidos de la mixtura.

Dado que los pesos π_i son las probabilidades de que y_t provenga de la i-ésima componente de la mixtura, se cumple que

$$\sum_{i=1}^{K} \pi_i = 1 \quad \text{y} \quad \pi_i \in [0,1] \text{ para } i = 1, \ldots, K \tag{1.2}$$

Observación 1.2. Nótese que las componentes de la mixtura no tienen por qué pertenecer a la misma familia de distribuciones, y que θ_i puede designar un vector de parámetros si así lo requiere la distribución de la componente i-ésima. Sin embargo, en la mayor parte de los casos se estudian modelos de mixturas cuyas componentes pertenecen a la misma familia paramétrica (Gómez, 2014).

Observación 1.3. Se llama número de parámetros de una mixtura, y se denota por $d(K)$, al número de parámetros desconocidos Ψ. Para simplificar la notación consideraremos π_K determinado por los demás pesos de la mixtura, dado que $\pi_K = 1 - \sum_{i=1}^{K-1} \pi_i$. Por tanto, el número de parámetros de la mixtura se corresponde con $d(K) = K - 1 + \sum_{i=1}^{K} \dim(\theta_i)$.

Un modelo de mixturas finito se puede interpretar como una variable aleatoria Y que está generada por K distintos procesos, cada uno de los cuales está modelado por la función de densidad $f_i(y_t|\theta_i)$ y π_i representan la proporción de observaciones de cada uno de los procesos.

Observación 1.4. Si el modelo es predictivo o de regresión lineal con Y_t la variable dependiente o respuesta y X_t la variable independiente o predictora, el modelo de mixturas se

expresa como

$$f(y_t|x_t, \Psi) = \sum_{i=1}^{K} \pi_i \cdot f_i(y_t|x_t, \theta_i)$$

donde cada $f_i(y_t|x_t)$ representa la función de densidad de la i-ésima componente condicionada a la variable X_t.

Veamos dos ejemplos particulares de modelos de mixturas de dos componentes que serán utilizados en la parte computacional de este trabajo (Capítulo 3), el primer caso para componentes normales y el segundo para componentes gamma.

Ejemplo 1.5. Los modelos de mixturas cuyas componentes siguen distribuciones normales de parámetros $\theta_i = (\mu_i, \sigma_i^2)$ son denotados por GMM (*Gaussian Mixture Models*), y a partir de (1.1) su expresión viene dada por:

$$f(y_t|\Psi) = \sum_{i=1}^{K} \pi_i \cdot \frac{1}{\sigma_i\sqrt{2\pi}} \cdot e^{\frac{-(y_t-\mu_i)^2}{2\sigma_i^2}} \quad \text{para } y_t \in \mathbb{R}^q \tag{1.3}$$

A continuación se representa un modelo de mixturas de dos componentes normales, una la normal estándar y la otra con parámetros $\mu_2 = 4$ y $\sigma_2^2 = 0{,}5$, con pesos $\pi_1 = 0{,}7$ y $\pi_2 = 0{,}3$, respectivamente, es decir, la función de densidad del modelo de mixturas es de la forma

$$f(y_t|\psi) = 0{,}7 \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{y_t^2}{2}} + 0{,}3 \cdot \frac{1}{\sqrt{2\pi \cdot 0{,}5}} \cdot e^{-\frac{(y_t-6)^2}{2 \cdot 0{,}5}}$$

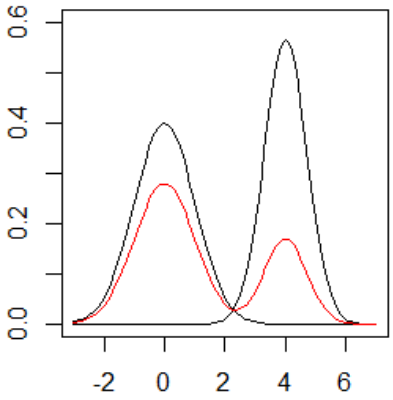

Figura 1.1: Representación del modelo de mixturas gausiano de 2 componentes (en rojo), junto con las densidades de cada una de las componentes (en negro).

Ejemplo 1.6. Los modelos de mixturas cuyas componentes siguen distribuciones gamma de parámetros $\theta_i = (\alpha_i, \beta_i)$ se expresan como

$$f(y_t|\Psi) = \sum_{i=1}^{K} \pi_i \cdot \frac{1}{\Gamma(\alpha_i) \cdot \beta_i^{\alpha_i}} \cdot y_t^{\alpha_i - 1} \cdot e^{-y_t/\beta_i} \quad \text{para } y_t \in \mathbb{R}^q \tag{1.4}$$

donde $\Gamma(\cdot)$ es la función gamma, $\Gamma(z) = \int_0^\infty t^{z-1} \cdot e^{-t} dt$. Si p es un entero positivo se tiene que $\Gamma(p) = (p-1)!$.

A continuación representamos una mixtura de dos componentes gamma de parámetros $\alpha_1 = 2$, $\alpha_2 = 5$, $\beta_1 = 2$ y $\beta_2 = 1$ y pesos $\pi_1 = 0{,}7$ y $\pi_2 = 0{,}3$, respectivamente.

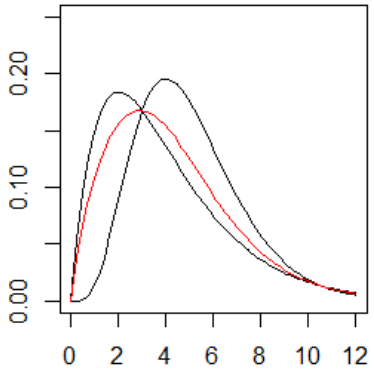

Figura 1.2: Representación del modelo de mixturas gamma de 2 componentes (en rojo), junto con las densidades de cada una de las componentes (en negro).

Para completar esta introducción a los modelos de mixturas se propone un ejemplo de un modelo de mixturas con componentes bivariantes.

Ejemplo 1.7. Se representa un modelo de mixturas de dos componentes normales bivariantes, una con parámetros $\mu_1 = (0,0)$ y $\Sigma_1 = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$ y la otra $\mu_2 = (4,6)$ y $\Sigma_1 = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}$, y con pesos $\pi_1 = 0{,}7$ y $\pi_2 = 0{,}3$, respectivamente.

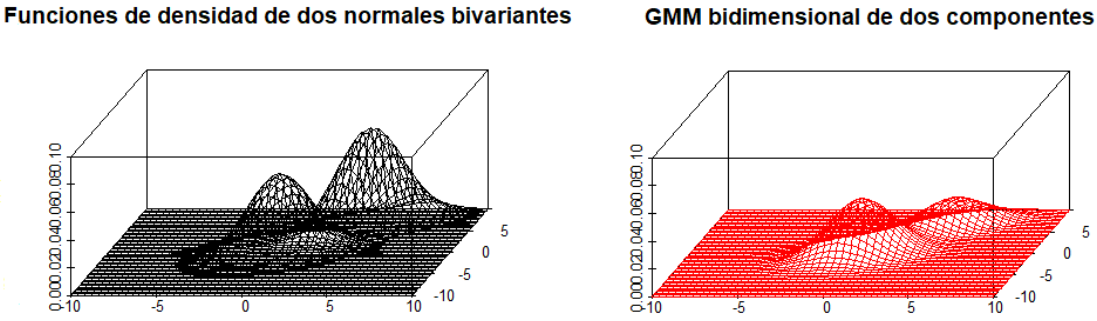

Figura 1.3: Representación del modelo de mixturas gausiano bivariante de 2 componentes (en rojo a la derecha), junto con las densidades de cada una de las componentes (en negro a la izquierda).

Cabe destacar, que a pesar de los ejemplos previos, en la mayor parte de los casos no resulta sencillo determinar el número de componentes de un mixtura o su familia paramétrica, como veremos en el siguiente ejemplo de STHDA (2017).

Ejemplo 1.8. Para este ejemplo se tienen en cuenta los datos recogidos en el paquete *MASS* sobre la duración y el tiempo de espera entre las erupciones de agua de un géiser. La representación de los datos es la siguiente.

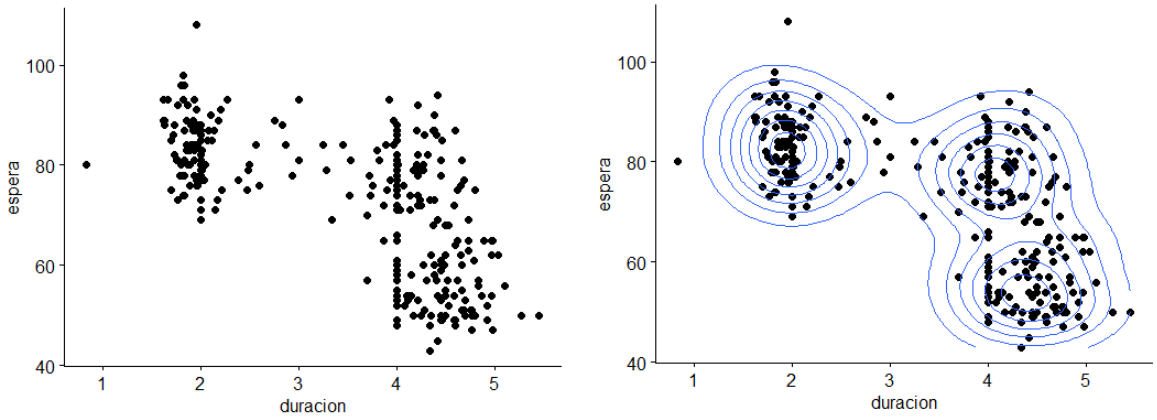

Figura 1.4: Representación de la duración frente al tiempo de espera de erupciones de agua (derecha e izquierda) y de las curvas de nivel de las densidades (izquierda)

En la primera representación no se aprecia el número exacto de clusters en los que se distribuyen los datos. En la segunda gráfica se han representado también las curvas de nivel de las densidades estimadas de los datos lo que permite determinar que se trata de un modelo de mixturas de tres componentes.

Por otro lado, para estimar los parámetros Ψ del modelo de mixturas a partir de unas observaciones $y_1, \ldots, y_n$ es necesario la identificabilidad de Ψ, es decir, que exista una única caracterización para el modelo de mixturas (Gómez, 2014). Por ello, a continuación se define la identificabilidad de los modelos de mixturas y algunos casos problemáticos de no identificabilidad que pueden surgir posteriormente en este trabajo.

Definición 1.9. Sean dos mixturas de la misma familia paramétrica, con componentes K y K^*, y vectores de parámetros Ψ y Ψ^*, se dice que son identificables si

$$f(y_t|\Psi) = f(y_t|\Psi^*),$$

esto es, si y sólo si $K = K^*$ y las etiquetas de las componentes pueden permutarse, es decir, que $\pi_i = \pi_i^*$ y $f(y_t|\theta_i) = f(y_t|\theta_i^*)$ para $i = 1, \ldots, K$.

Para los modelos de mixturas que trataremos en este trabajo fin de máster hay que distinguir dos casos problemáticos de no identificabilidad que se pueden presentar en su aplicación práctica, debido en parte al desconocimiento del número de subgrupos en la población (Leisch, 2004; Gómez, 2014).

Reetiquetado de las componentes de la mixtura

Veamos el problema del reetiquetado con un caso particular. Por ejemplo, sean dos modelos de mixturas de tipo GMM con dos componentes y con parámetros $\Psi = (\theta_1, \theta_2, \pi_1, \pi_2)$ y $\Psi^* = (\theta_2, \theta_1, \pi_2, \pi_1)$. Se tiene de forma trivial que $f(y_t|\Psi) = f(y_t|\Psi^*)$, pero no se trata de mixturas identificables ya que $\Psi \neq \Psi^*$.

No obstante, esto no resulta verdaderamente un problema porque afecta exclusivamente al etiquetado de las componentes y por tanto, sólo a la interpretación de los resultados.

Sobreestimación del modelo

El problema de sobreestimación lo ilustramos mediante dos situaciones concretas.

En primer lugar, sea un GMM de dos componentes y parámetros $\Psi = (\theta_1, \theta_2, \pi_1, \pi_2)$ y un GMM de tres componentes y parámetros $\Psi^* = (\theta_1, \theta_2, \theta_3, \pi_1, \pi_2, 0)$. Evidentemente se tiene que $f(y_t|\Psi) = f(y_t|\Psi^*)$, pero no son identificables ya que $K \neq K^*$ y $\Psi \neq \Psi^*$.

En este caso una de las componentes no afecta a la mixtura por lo que el modelo de mixturas con parámetros Ψ^* puede ser representado por un modelo de menor tamaño, el modelo obtenido con los parámetros Ψ. De manera que este tipo de no identificabilidad se puede evitar simplemente imponiendo que todos los pesos sean no nulos.

En segundo lugar, sea un GMM de dos componentes y parámetros $\Psi = (\theta_1, \theta_2, \pi_1, \pi_2)$ y un GMM de tres componentes y parámetros $\Psi^* = (\theta_1, \theta_2, \theta_2, \pi_1, \pi_2 - \pi_3, \pi_3)$. También ocurre que las mixturas no son identificables ya que $K \neq K^*$ y $\Psi \neq \Psi^*$, pero $f(y_t|\Psi) = f(y_t|\Psi^*)$.

Se observa que dos de las componentes tienen los mismos parámetros θ_i, por lo que el GMM obtenido con los parámetros Ψ^* puede ser representado por otro reducido que se crea acumulando los pesos de las componentes con los mismos parámetros θ_i, es decir, el GMM de parámetros Ψ. Así, se puede evitar esta situación imponiendo que los parámetros θ_i sean distintos para cada componente de la mixtura.

1.2. Clustering mediante modelos de mixturas

La idea general del clustering o clasificación de datos mediante modelos de mixturas consiste en asignar a cada dato observado una etiqueta que indique su procedencia de una u otra componente de la mixtura, es decir, su pertenencia al subgrupo de la población correspondiente a la componente de la mixtura, formando así un agrupamiento o clustering de los datos.

En este contexto, se considera que la variable Y, que modeliza los datos de la muestra, está incompleta ya que requiere otra variable Z que indique la pertenencia de cada dato a una u otra componente. En definitiva, cada componente de la mixtura se identifica con un clúster al que pertenecen los datos muestrales.

Por tanto, se introduce una variable $Z = \{Z_1, \ldots, Z_n\}$, llamada variable latente, que está asociada a la variable Y distribuída según una mixtura de K componentes. Las variables Z_t con $t = 1, \ldots, n$ son independientes y $C = (Y, Z)$ representa el vector completo de datos muestrales (Gómez, 2014; Picard, 2007).

Dado que el elemento i-ésimo de Z_t indica si la observación y_t proviene o no de la componente i-ésima de la mixtura, se define para $i = 1, \ldots, K$ y $t = 1, \ldots, n$

$$Z_{ti} = \begin{cases} 1 & \text{si } y_t \text{ proviene de la componente i-ésima} \\ 0 & \text{en otro caso} \end{cases}$$

Observación 1.10. La variable Z_t toma valores $z_t = (z_{t1}, \ldots, z_{t_K})$, donde z_t pertence a la base canónica de $\mathbb{R}^K$.

Observación 1.11. Dado que Z_{ti} indica si la observación y_t procede o no de la i-ésima componente de la mixtura, se tiene que $Z_t = (Z_{t1}, \ldots, Z_{tK})$ sigue una distribución multinomial de parámetros 1 y π, siendo $\pi = (\pi_1, \ldots, \pi_K)$ (Aitkin and Rubin, 1985). Por tanto, se tiene que

$$p(z_t) = P(Z_t = z_t) = \frac{1}{z_{t1}! \cdots z_{tK}!} \cdot \pi_1^{z_{t1}} \cdots \pi_K^{z_{tK}} = \prod_{i=1}^{K} \pi_i^{z_{ti}}$$

En términos de clustering de datos, el i-ésimo peso de la mixtura, π_i, puede verse como la probabilidad a priori de que una observación pertenezca a la componente i de la mixtura, es decir, $P(Z_{ti} = 1) = \pi_i$ para $i = 1, \ldots K$.

Se tiene, de forma sencilla, que $f(Y_t = y_t | Z_{ti} = 1)$ es equivalente a $f_i(y_t|\theta_i)$, ya que el hecho de que $Z_{ti} = 1$ implica que la observación y_t procede de la i-ésima componente de la mixtura, y por tanto la función de densidad condicionada es la función de densidad de dicha componente (Gómez, 2014).

La probabilidad a posteriori de Z_{ti}, o lo que es lo mismo, la probabilidad a posteriori de que y_t provenga de la i-ésima componente de la mixtura, es

$$\hat{\tau}_{it} = P(Z_{ti} = 1 | Y_t = y_t) = \frac{P(Z_{ti} = 1) \cdot f(Y_t = y_t | Z_{ti} = 1)}{f(Y_t = y_t)} = \frac{\pi_i \cdot f_i(y_t|\theta_i)}{\sum_{l=1}^{K} \pi_l \cdot f_l(y_t|\theta_l)} \tag{1.5}$$

El siguiente resultado se puede encontrar en Gómez (2014).

Proposición 1.12. *Siendo C el vector de datos completos de la muestra, se tiene que la función de densidad de su componente $C_t = (Y_t, Z_t)$ es*

$$f(c_t|\Psi) = \prod_{i=1}^{K} (\pi_i \cdot f_i(y_t|\theta_i))^{z_{ti}}$$

Demostración. La función de densidad de la variable C_t es

$$f(c_t|\Psi) = f(y_t, z_t|\Psi) = f(y_t|z_t; \Psi) \cdot p(z_t)$$

Por un lado, se ha visto que $f(Y_t = y_t | Z_{ti} = 1)$ es equivalente a $f_i(y_t|\theta_i)$. Además, las variables Z_{ti} son independientes y sólo toman valores 0 ó 1, por lo que

$$f(y_t|z_t; \Psi) = P(Y_t = y_t | Z_{t1} = z_{t1}, \ldots, Z_{tK} = z_{tK}; \Psi) = \prod_{i=1}^{K} f_i(y_t|\theta_i)^{z_{ti}}$$

Por tanto

$$f(c_t|\Psi) = \prod_{i=i}^{K} f_i(y_t|\theta_i)^{z_{ti}} \cdot \prod_{i=i}^{K} \pi_i^{z_{ti}} = \prod_{i=1}^{K} \left(\pi_i \cdot f_i(y_t|\theta_i)\right)^{z_{ti}}.$$

□

En este trabajo se estudiarán los modelos de mixturas como métodos de clasificación de datos por lo que se considerará que se dispone de datos incompletos y se usará la estructura anteriormente expuesta. También se pretende estimar los parámetros de la mixtura que mejor se ajusta a los datos estudiados por lo que en el siguiente capítulo se introduce un algoritmo muy útil para ello llamado algoritmo EM.

Capítulo 2

Estimación de mixturas finitas

En este capítulo el objetivo es estimar los parámetros Ψ de un modelo de mixturas y la clasificación que realizan de los datos modelizados. En primer lugar se propone realizar una estimación por el método de máxima verosimilitud (MLE) y se constata la dificultad de calcular de forma analítica los parámetros con este método. Para solventar este problema, se propone como alternativa el algoritmo esperanza-maximización (EM) para el cual es necesario considerar los datos como incompletos y usar el modelo basado en clustering mediante mixturas. A continuación, se estudia el algoritmo EM: la estructura del algoritmo, la convergencia, los criterios de selección que nos permiten elegir el modelo que mejor se ajusta a la muestra y la influencia de los valores de inicialización en la convergencia del algoritmo.

Posteriormente, se introducen nociones básicas relativas a curvas ROC y a algunos de sus índices asociados, que permiten el estudio de la efectividad de una prueba o modelo de clasificación con dos posibles resultados. También se incluyen conceptos como la especificidad, sensibilidad y valor predictivo global que permitirán comparar distintos modelos de clasificación sobre una misma muestra. Esto será de gran utilidad en el siguiente capítulo dado que se pretende clasificar una muestra de tejidos en dos grupos: enfermos o sanos.

2.1. Estimación de parámetros mediante MLE

Para estimar el parámetro Ψ de la mixtura, existen diversos métodos de estimación puntual entre los que destacan el método de los momentos, que consiste en igualar algunos momentos poblacionales con los muestrales, y el método de máxima verosimilitud, que se basa en hallar los parámetros que hacen máxima la probabilidad de obtener una muestra dada. Evitando los numerosos inconvenientes que presenta el método de los momentos, en este apartado usaremos el método de máxima verosimilitud (Gómez, 2014; McNicholas, 2016).

Definición 2.1. Sean $Y_1, \ldots, Y_n$, n vectores aleatorios q-dimensionales observables y cuya pertenencia a una u otra subpoblación desconocemos. Su función de verosimilitud

se define como

$$L(\Psi|y) = \prod_{t=1}^{n} f(y_t|\Psi) = \prod_{t=1}^{n} \sum_{i=1}^{K} \pi_i \cdot f_i(y_t|\theta_i) \tag{2.1}$$

siendo Ψ el vector de parámetros desconocidos e $y = (y_1, \ldots, y_n)$ las observaciones independientes de la variable Y.

Se pretende estimar los parámetros desconocidos Ψ mediante la maximización de la función de verosimilitud (2.1), es decir, hallar $\hat{\Psi} = \underset{\Psi}{\max} L(\Psi|y)$.

Habitualmente resulta más sencillo maximizar la función de log-verosimilitud, que tiene, en este caso, la siguiente expresión

$$l(\Psi|y) = \log L(\Psi|y) = \sum_{t=1}^{n} \log f(y_t|\Psi) = \sum_{t=1}^{n} \log \left(\sum_{i=1}^{K} \pi_i \cdot f_i(y_t|\theta_i) \right) \tag{2.2}$$

Dado que el máximo de (2.1) y (2.2) es el mismo, podemos estimar Ψ cómo $\hat{\Psi} = \underset{\Psi}{\max} l(\Psi|y)$.

Observación 2.2. El valor de $\hat{\Psi}$ se conoce como un estimador de máxima verosimilitud de Ψ y no tiene porqué ser único.

Se trata entonces de hallar los máximos de (2.2), es decir, es necesario considerar los valores para los que se cumple que

$$\frac{\partial}{\partial \Psi} \left[\sum_{t=1}^{n} \log \left(\sum_{i=1}^{K} \pi_i \cdot f_i(y_t|\theta_i) \right) \right] = 0 \tag{2.3}$$

Resulta evidente que la resolución de (2.3) es difícil e incluso puede llegar a no tener solución analítica, por lo que se requieren métodos iterativos que nos permitan aproximarla. A continuación, se introduce el algoritmo EM para datos incompletos. Se usará la función de verosimilitud para los datos completos C que tiene la siguiente expresión (Gómez, 2014; Picard, 2007; Aitkin and Rubin, 1985).

$$L(\Psi|c_t) = \prod_{t=1}^{n} \prod_{i=1}^{K} (\pi_i \cdot f_i(y_t|\theta_i))^{z_{ti}} \tag{2.4}$$

y por lo tanto, la función de log-verosimilitud es

$$l(\Psi|c_t) = \sum_{t=1}^{n} \sum_{i=1}^{K} z_{ti} \log (\pi_i \cdot f_i(y_t|\theta_i)) = \sum_{t=1}^{n} \sum_{i=1}^{K} z_{ti} \log \pi_i + \sum_{t=1}^{n} \sum_{i=1}^{K} z_{ti} \cdot \log (f_i(y_t|\theta_i)) \tag{2.5}$$

2.2. Algoritmo EM

El algoritmo EM fue propuesto para estimar mediante máxima verosimilitud los parámetros de modelos con datos incompletos. Los datos incompletos son muy comunes en la práctica, pueden ser producidos por limitaciones de los dispositivos de medición o bien ser inherentes a ciertos modelos. Hemos visto que los datos producidos por modelos de mixturas finitas pueden considerarse de forma natural datos incompletos, y por tanto, el algortimo EM resulta un buen método de estimación de parámetros del modelo (Gómez, 2014).

La estructura del algoritmo EM se puede encontrar en numerosos artículos como Gómez (2014), Aitkin and Rubin (1985) y Ng et al. (2012), entre otros. A continuación se desarrolla su expresión para la m-ésima iteración:

Paso E

Se trata de calcular la esperanza de la función de log-verosimilitud de los datos completos condicionada a la observación y_t:

$$Q(\Psi|\Psi^{(m)}) = E\left[l(\Psi|c_t)|\, Y_t = y_t\right]_{\Psi^{(m)}} \tag{2.6}$$

Usando la expresión (2.5) y por la linealidad de la esperanza, a partir de (2.6) se tiene que

$$Q(\Psi|\Psi^{(m)}) = \left(\sum_{t=1}^{n}\sum_{i=1}^{K} E\left[Z_{ti}|\, Y_t = y_t\right]_{\Psi^{(m)}} \cdot \log \pi_i\right) + \left(\sum_{t=1}^{n}\sum_{i=1}^{K} E\left[Z_{ti}|\, Y_t = y_t\right]_{\Psi^{(m)}} \cdot \log(f_i(y_t|\theta_i))\right)$$

Dado que $E\left[Z_{ti}|\, Y_t = y_t\right]_{\Psi^{(m)}} = P(Z_{ti} = 1|Y_t = y_t)_{\Psi^{(m)}}$ y por (1.5), se tiene que

$$E\left[Z_{ti}|\, Y_t = y_t\right]_{\Psi^{(m)}} = \hat{\tau}_{it}^{(m)}$$

El valor $\hat{\tau}_{it}^{(m)}$ se corresponde con la probabilidad a posteriori estimada de que y_t provenga de la componente i en la m-ésima iteración del algoritmo. Por tanto,

$$Q(\Psi|\Psi^{(m)}) = \sum_{t=1}^{n}\sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \log \pi_i + \sum_{t=1}^{n}\sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \log(f_i(y_t|\theta_i)) \tag{2.7}$$

Paso M

Consiste en maximizar respecto a Ψ la función $Q(\Psi|\Psi^{(m)})$ obtenida en el paso anterior, es decir, se busca

$$\Psi^{(m+1)} = \max_{\Psi} Q(\Psi|\Psi^{(m)})$$

que por (2.7) se tiene que

$$\Psi^{(m+1)} = \max_{\Psi}\left(\sum_{t=1}^{n}\sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \log \pi_i + \sum_{t=1}^{n}\sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \log(f_i(y_t|\theta_i))\right) \tag{2.8}$$

Dado que $\Psi = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$ y el primer término de (2.8) depende sólo de π_i y el segundo de θ_i, la maximización se puede realizar respecto a π_i el primer término, y respecto a θ_i el segundo, es decir

$$\Psi^{(m+1)} = \max_{\pi_i} \left(\sum_{t=1}^{n} \sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \log \pi_i \right) + \max_{\theta_i} \left(\sum_{t=1}^{n} \sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \log(f_i(y_t|\theta_i)) \right) \tag{2.9}$$

Veamos en primer lugar la maximización del primer término de (2.9).

Sean $\pi = (\pi_1, \ldots, \pi_K)$, $g(\pi) = \sum_{t=1}^{n} \sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \log \pi_i$ y $h(\pi) = \sum_{i=1}^{K} \pi_i$, entonces se trata de

$$\begin{array}{cc} \text{maximizar} & g(\pi) \\ \text{s.a} & h(\pi) = 1 \end{array}$$

Usando multiplicadores de Lagrange podemos maximizar $g(\pi)$ bajo la condición $h(\pi)$. Se tiene que

$$\frac{\partial}{\partial \pi_i} \left(g(\pi) + \lambda \cdot (h(\pi) - 1) \right) = 0 \Leftrightarrow \frac{\partial}{\partial \pi_i} g(\pi) + \lambda \cdot \frac{\partial}{\partial \pi_i} h(\pi) = 0$$

Para cada componente i, se tiene que la expresión anterior es equivalente a

$$\sum_{t=1}^{n} \frac{\hat{\tau}_{it}^{(m)}}{\pi_i} + \lambda = 0 \Leftrightarrow \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} = -\lambda \pi_i \tag{2.10}$$

Sumando sobre i, (2.10) es equivalente a

$$\sum_{i=1}^{K} \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} = -\lambda \tag{2.11}$$

Además, por la expresión (1.5), sabemos que

$$\sum_{i=1}^{K} \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} = \sum_{t=1}^{n} \sum_{i=1}^{K} P(Z_{ti} = 1 | Y_t = y_t)_{\Psi^{(m)}} = \sum_{t=1}^{n} \sum_{i=1}^{K} \frac{\pi_i \cdot f_i(y_t|\theta_i)}{\sum_{l=1}^{K} \pi_l \cdot f_l(y_t|\theta_l)} \Bigg|_{\Psi^{(m)}} =$$

$$\sum_{t=1}^{n} \frac{1}{\sum_{l=1}^{K} \pi_l \cdot f_l(y_t|\theta_l)} \cdot \sum_{i=1}^{K} \pi_i \cdot f_i(y_t|\theta_i) \Bigg|_{\Psi^{(m)}} = \sum_{t=1}^{n} \frac{f(y_t|\Psi^{(m)})}{f(y_t|\Psi^{(m)})} = n$$

Por tanto,

$$\sum_{i=1}^{K} \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} = n \tag{2.12}$$

De (2.11) y (2.12) se concluye que $\lambda = -n$ y substituyendo en (2.10) obtenemos el valor de π_i para el cual se maximiza $g(\pi)$ sujeto a $h(\pi)$. Este nuevo valor de los pesos de

la mixtura para la m-ésima iteración es denotado por $\hat{\pi}_i^{(m+1)}$ y su expresión se corresponde con

$$\hat{\pi}_i^{(m+1)} = \frac{\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)}}{n} \tag{2.13}$$

La maximización del segundo término de (2.9) dependerá de las funciones de densidad de las componentes de la mixtura. En los siguientes ejemplos veremos la estimación del parámetro θ_i para el caso de un GMM y de un modelo de mixturas gamma, que serán los modelos considerados en el capítulo de la puesta en práctica.

Ejemplo 2.3. Para un GMM, se tiene que

$$\log(f_i(y_t|\theta_i)) = \log\left(\frac{1}{\sigma_i\sqrt{2\pi}} \cdot e^{\frac{-(y_t-\mu_i)^2}{2\sigma_i^2}}\right) = -\log(\sigma_i) - \log(\sqrt{2\pi}) - \frac{(y_t-\mu_i)^2}{2\sigma_i^2}$$

Por tanto, el segundo término de la expresión (2.9) es

$$\max_{\theta_i}\left(\sum_{t=1}^{n}\sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \left(-\log\sigma_i - \log(\sqrt{2\pi}) - \frac{(y_t-\mu_i)^2}{2\sigma_i^2}\right)\right)$$

Se trata entonces de calcular

$$\frac{\partial}{\partial\theta_i}\left(\sum_{t=1}^{n}\sum_{i=1}^{K} \hat{\tau}_{it}^{(m)} \cdot \left(-\log\sigma_i - \log(\sqrt{2\pi}) - \frac{(y_t-\mu_i)^2}{2\sigma_i^2}\right)\right) = 0$$

Dado que $\theta_i = (\mu_i, \sigma_i^2)$, maximizamos en primer lugar con respecto a μ_i.
Sea

$$g(\theta_i) = -\sum_{i=1}^{K}\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \log\sigma_i - \sum_{i=1}^{K}\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \log(\sqrt{2\pi}) - \sum_{i=1}^{K}\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot \frac{(y_t-\mu_i)^2}{2\sigma_i^2}$$

Se calcula

$$\frac{\partial}{\partial\mu_i} g(\theta_i) = 0$$

Para cada componente i se tiene que la expresión anterior es equivalente a

$$\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot \frac{y_t-\mu_i}{2\sigma_i^2} = 0 \Leftrightarrow \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot y_t = \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot \mu_i$$

Por tanto, el estimador de μ_i en la m-ésima iteración para la componente i es

$$\hat{\mu_i}^{(m+1)} = \frac{\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot y_t}{\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)}} \tag{2.14}$$

De forma análoga, maximizando respecto a σ_i^2, se obtiene que para la i-ésima componente

$$\frac{\partial}{\partial \sigma_i^2} g(\theta_i) = 0 \Leftrightarrow -\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot \left(-\frac{1}{2\sigma_i^2} + \frac{(y_t - \mu_i)^2}{2\sigma_i^4} \right) = 0 \Leftrightarrow \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} = \frac{1}{\sigma_i^2} \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot (y_t - \mu_i)^2$$

y por tanto el estimador de σ_i^2 en la m-ésima iteración para la componente i resulta ser

$$\left(\hat{\sigma_i}^2\right)^{(m+1)} = \frac{\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot (y_t - \mu_i)^2}{\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)}} \tag{2.15}$$

Ejemplo 2.4. Para un modelo de mixturas gamma, se tiene que

$$\log(f_i(y_t|\theta_i)) = \log\left(\frac{y_t^{\alpha_i - 1} \cdot e^{-y_t/\beta_i}}{\Gamma(\alpha_i) \cdot \beta_i^{\alpha_i}} \right) = (\alpha_i - 1) \cdot \log(y_t) - \frac{y_t}{\beta_i} - \log(\Gamma(\alpha_i)) - \alpha_i \cdot \log(\beta_i)$$

Por tanto el segundo término de la expresión (2.9) es

$$\max_{\theta_i} \left(\sum_{t=1}^{n} \sum_{i=1}^{K} \hat{\tau}_{it} \cdot \left((\alpha_i - 1) \cdot \log(y_t) - \frac{y_t}{\beta_i} - \log(\Gamma(\alpha_i)) - \alpha_i \cdot \log(\beta_i) \right) \right) \tag{2.16}$$

Dado que $\theta_i = (\alpha_i, \beta_i)$, maximizamos respecto a α_i en primer lugar.

Para cada componente i realizando la derivada parcial respecto a α_i e igualando a cero se tiene que

$$\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot \left(\log\left(\frac{y_t}{\beta_i} \right) - \psi(\alpha_i) \right) = 0 \Leftrightarrow \psi(\alpha_i) = \frac{\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot \log\left(\frac{y_t}{\beta_i} \right)}{\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)}}$$

donde $\psi(x) = \frac{\partial}{\partial x} \log(\Gamma(x))$. Por tanto, no se puede dar una expresión analítica para el estimador de $\hat{\alpha}_i$.

A continuación se maximiza (2.16) respecto a β_i.

Para cada componente i realizando la derivada parcial respecto a β_i e igualando a cero se tiene que

$$\sum_{t=1}^{n} \hat{\tau}_{it}^{(m)} \cdot \left(\frac{y_t}{\beta_i^2} - \frac{\alpha_i}{\beta_i} \right) =\Leftrightarrow \frac{1}{\alpha_i} \cdot \sum_{t=1}^{n} \hat{\tau}_{it} \cdot y_t = \beta_i \cdot \sum_{t=1}^{n} \hat{\tau}_{it}$$

Por tanto, el estimador de β_i en la m-ésima iteración para la componente i es

$$\hat{\beta}_i^{(m+1)} = \frac{\sum_{t=1}^{n} \hat{\tau}_{it} \cdot y_t}{\hat{\alpha}_i^{(m+1)} \cdot \sum_{t=1}^{n} \hat{\tau}_{it}^{(m)}}$$

2.3. Convergencia del algoritmo EM

En esta sección se pretende estudiar la convergencia del algoritmo y algunas de las problemáticas que presenta. A continuación se encuentra el resultado que garantiza la monotonía de las soluciones del algoritmo y que, posteriormente, nos permitirá demostrar que las soluciones obtenidas en los sucesivos pasos del algoritmo no empeoran en términos del valor de la función de verosimilitud.

Teorema 2.5 (Monotonía del algoritmo EM)**.** *Si* $Q(\Psi|\Psi^{(m)}) \geq Q(\Psi^{(m)}|\Psi^{(m)})$ *entonces* $l(\Psi|C) \geq l(\Psi^{(m)}|C)$.

El siguiente resultado es necesario para la demostración del teorema anterior (Freien Universität Berlin, 2010).

Observación 2.6. La divergencia de Kullback-Leibler o entropía relativa entre dos funciones de probabilidad f y g, se define como

$$\sum_i f(i) \log\left(\frac{f(i)}{g(i)}\right)$$

y una de sus propiedades es que siempre es positiva.

Demostración. Por definición, se tiene que

$$l(\Psi|C) - l(\Psi^{(m)}|C) = Q(\Psi|\Psi^{(m)}) - Q(\Psi^{(m)}|\Psi^{(m)}) + \sum_{i=1}^{K} f_i(y_t|\theta_i)|_{\Psi^{(m)}} \cdot \log\left(\frac{f_i(y_t|\theta_i)|_{\Psi^{(m)}}}{f_i(y_t|\theta_i)}\right) \tag{2.17}$$

El último término de la expresión (2.17) es una entropía relativa entre dos funciones de densidad, por lo que es positiva. Por hipótesis $Q(\Psi|\Psi^{(m)}) - Q(\Psi^{(m)}|\Psi^{(m)}) \geq 0$. Por tanto, es claro que

$$l(\Psi|C) \geq l(\Psi^{(m)}|C)$$

□

Como $\Psi^{(m+1)} = \max_{\Psi} Q(\Psi|\Psi^{(m)})$, entonces $Q(\Psi^{(m+1)}|\Psi^{(m)}) \geq Q(\Psi^{(m)}|\Psi^{(m)})$. Así, por el teorema anterior, la sucesión $\{\Psi^{(m)}\}$ generada por el algoritmo hace que $l(\Psi|C)$ sea monótona creciente, es decir, que $l(\Psi^{(m+1)}|C) \geq l(\Psi^{(m)}|C)$ para todo m. Por tanto, la monotonía del algoritmo garantiza que para las sucesivas iteraciones, la estimación obtenida no empeora en términos de la función de verosimilitud, pero no se puede garantizar la convergencia del algoritmo. De hecho, no existe ningún resultado que garantice dicha convergencia, dado que depende de la función de log-verosimilitud $l(\Psi|C)$, de $Q(\Psi|\Psi^{(m)})$ y del valor de inicio $\Psi^{(0)}$ (Gupta et al., 2011).

Convergencia a un máximo local

Como hemos visto, a pesar de la obtención de un máximo de la función de log-verosimilitud en el paso M, no se garantiza la convergencia del algortimo EM al máximo global de la función. Sin embargo, bajo ciertas condiciones sobre la función de log-verosimilitud y sobre el conjunto al que pertenecen los parámetros Ψ, es posible probar que $\{\Psi^{(m)}\}$ converge a un máximo local de $l(\Psi|C)$ o al menos a un punto estacionario de la misma (Couvreur, 1997).

Se puede encontrar más información sobre la convergencia del algoritmo EM en Couvreur (1997), Gupta et al. (2011) y Ng et al. (2012), entre otros.

2.4. Selección del mejor modelo

Uno de los problemas más importantes de las técnicas de clustering de datos es decidir el número de clusters o componentes de la mixtura que se van a tomar para aplicar el algoritmo EM. Para elegir el número de clusters que mejor se ajusta a los datos se utilizan criterios de selección. La idea detrás de estos criterios es encontrar el equilibrio entre una buena modelización y un número razonable de parámetros, por lo que los modelos con demasiados parámetros deben ser penalizados de alguna forma (Olivier, 1999). Una manera de conseguirlo es usar un tipo de criterio de log-verosimilitud, llamado criterio de información (IC). Este criterio es una especie de función de verosimilitud penalizada compuesta por una función negativa de log-verosimilitud y un término de penalización que es mayor cuantos más parámetros compongan el modelo en cuestión. La forma general de un IC es

$$IC(K) = -2 \cdot l(\hat{\Psi}|C) + d(K) \cdot a_n$$

donde $\hat{\Psi}$ es la estimación de los parámetros del modelo de K componentes de la mixtura, $d(K)$ el número de parámetros del modelo de mixturas de K componentes, $l(\hat{\Psi}|C)$ la función de log-verosimilitud de la forma (2.5), n el tamaño de la muestra y a_n una función creciente. El orden óptimo del modelo es aquel que minimiza el IC, es decir, el orden del modelo será $\hat{d}(K) = \min_K IC(K)$.

A continuación exponemos los criterios de información más conocidos y usados.

- El criterio de información Akaike (AIC) cuya expresión es

  $$AIC(K) = -2 \cdot l(\hat{\Psi}|C) + 2 \cdot d(K)$$

  Se sabe que este criterio sobreestima el orden del modelo (Hu, 2015; Olivier, 1999).

- El criterio de información bayesiano (BIC) que tiene la siguiente expresión

  $$BIC(K) = -2 \cdot l(\hat{\Psi}|C) + d(K) \cdot \log n \tag{2.18}$$

  En este caso, el término de penalización depende del tamaño de la muestra n, por lo que conforme $n \to \infty$ la penalización es mayor y no sobreestima tanto el orden del

modelo como pasaba con el AIC. Se considera que en algunas ocasiones subestima el orden de la mixtura (Hu, 2015; Olivier, 1999; Baudry et al., 2015).

- El criterio *Integrated complete likelihood* (ICL)

$$ICL(K) = -2 \cdot l(\hat{\Psi}|C) + d(K) \cdot \log(n) + Ent(K)$$

donde $Ent(K)$ es la entropía media que se expresa como

$$Ent(K) = -2 \sum_{i=1}^{K} \sum_{t=1}^{n} \tau_{it} \cdot \log \tau_{it}$$

Se trata de un criterio derivado del BIC ya que $ICL(K) = BIC(K) + Ent(K)$. En Baudry et al. (2015) se concluye que se debería usar el criterio BIC cuando se sabe que las componentes siguen distribuciones gausianas y no sea un problema obtener clusters solapados. Por el contrario recomienda el uso del criterio ICL para obtener grupos bien separados y cuando las distribuciones de las componentes no son claramente gausianas .

- El criterio Akaike corregido

$$AIC_c(K) = AIC(K) + \frac{2(d(K)+1) \cdot (d(K)+2)}{n - d(K) - 2}$$

que se propuso para corregir la sobreestimación caractarística del criterio Akaike (Hu, 2015).

Los dos últimos criterios proporcionan un resultado intermedio entre el AIC y el BIC que sobreestiman y subestiman el orden de la mixtura respectivamente (Hu, 2015).

En el capítulo de computación usaremos los tres primeros criterios para comparar modelos por ser los más utilizados y encontrarse implementados en el paquete de estudio FlexMix.

2.5. Inicialización del algoritmo EM

La elección de los valores de inicio del algoritmo EM influye notablemente en la velocidad de convergencia del algoritmo y en su capacidad para encontrar el máximo global, o al menos el mejor máximo local en el caso de que la función de verosimilitud no esté acotada.

Para responder a ciertas limitaciones del algoritmo como la dependencia de los valores de inicio, la convergencia a puntos de silla de la función de verosimilitud o la lenta convergencia, que normalmente se dan cuando existe superposición entre los clusters o cuando el número de componentes es grande, se proponen variantes del algoritmo EM y varias estrategias de inicialización del mismo. Es importante destacar que aquí solo se analizarán algunos de los métodos de inicio (de entre la multitud de estrategias existentes), y que la elección del mejor de ellos no es una tarea sencilla y todavía es fruto de estudio y discusión.

En primer lugar hablaremos de dos métodos generales de clasificación de datos, ambos independientes del modelo de clustering mediante mixturas: el método *K-means* y el clustering jerárquico, cuyos resultados pueden usarse también como valores de inicialización del algoritmo EM (Hu, 2015). Posteriormente, se introducen algunas de las muchas variantes del algoritmo que pueden ser usadas para estimar directamente los parámetros de la mixtura o para calcular unos buenos valores iniciales para el algoritmo EM. A continuación se comentan algunas estrategias de inicialización propuestas por Scharl et al. (2009), Biernacki et al. (2003) y Michael and Melnykov (2016).

2.5.1. Métodos generales de agrupamiento de datos

Como hemos comentado previamente el análisis clúster engloba infinidad de algoritmos de clasificación de una muestra con la finalidad de establecer grupos, clusters o conglomerados de la población de estudio. Entre estas técnicas se encuentra el uso de modelos de mixturas. A continuación estudiaremos otros métodos de clustering de datos que pueden ser usados como estrategias de inicialización del algoritmo EM para obtener la clasificación de los datos mediante modelos de mixturas finitas.

Las agrupaciones producidas por los algoritmos del análisis clúster se basan en la similitud o disimilitud entre los individuos de la muestra. Es necesario tomar medidas numéricas de disimilitud, esto es, una medida de distancia entre observaciones.

Los métodos de análisis clúster pueden clasificarse en: jerárquicos, cuando se organizan los individuos en conglomerados anidados mostrando relaciones y estructuras entre los datos; y no jerárquicos, que clasifican los individuos en conglomerados no anidados. A continuación, se comentan brevemente las características generales de los métodos jerárquicos y del método no jerárquico *K-means*.

- **Clustering usando el método *K-means***

El método de agrupamiento *K-means* es uno de los algoritmos de clustering más populares. La idea general es que cada clúster puede ser representado por un valor medio que se denomina el centro (media) del clúster y se denota por μ_i. Los demás datos del conglomerado estarán distribuidos alrededor del centro del clúster. La finalidad del algoritmo es encontrar el vector de agrupamiento $Z = (Z_1, \ldots, Z_n)$ que minimice la suma de cuadrados dentro de cada conglomerado, es decir, minimizar

$$WCSS = \sum_{i=1}^{K} \sum_{t=1}^{n} z_{ti} \cdot ||y_t - \mu_i||^2$$

donde $\mu_i = \frac{\sum_{t=1}^{n} z_{ti} \cdot y_t}{\sum_{t=1}^{n} z_{ti}}$ y $||\cdot||$ es la distancia euclídea.

El algoritmo introduce o elimina individuos de los clusters para conseguir los resultados de ANOVA más significativos. El ANOVA o análisis de la varianza evalúa la variabilidad entre grupos frente a la dispersión de los datos dentro de un mismo grupo respecto a su media haciendo uso de la distribución F de Snedecor. Por tanto, el método *K-means* consigue minimizar la variabilidad entre los individuos de un mismo clúster y maximizarla entre individuos de clusters distintos.

El vector de clustering de datos Z obtenido mediante este algoritmo puede ser usado para inicializar el algoritmo EM.

El método *K-means* requiere unos valores de inicio para los centros de los clusters (μ_i), por lo que también es necesaria una buena inicialización. Habitualmente se escogen de forma aleatoria K observaciones de los datos y se usan como centros. Se ejecuta el algoritmo varias veces y se escoge aquel que proporcione el menor $WCSS$. La inicialización óptima de este algoritmo se produciría si cada uno de los valores iniciales tomados de los centros perteneciese a un clúster verdadero distinto. La probabilidad de que esto ocurra es de

$$\frac{\prod_{i=1}^{K} n_i}{\binom{n}{K}} \leq \frac{(n/K)^K}{\binom{n}{K}} \tag{2.19}$$

donde n_i es el tamaño del i-ésimo clúster.

Se tiene que para una muestra grande, la expresión anterior tiene un valor aproximado de $\frac{K!}{K^K}$, que, conforme $K \to \infty$, tiende a 0. Se concluye entonces que si el número de clusters es grande resulta difícil conseguir una buena inicialización del algoritmo K-means y por ende, del algoritmo EM (Hu, 2015).

- **Clustering jerárquico**

Se trata de un método de agrupamiento que clasifica los datos en conglomerados anidados. Existen dos tipos: el aglomerativo y el divisivo. En el primero, cada observación se considera un conglomerado distinto inicialmente y estos se van mezclando conforme avanza el algoritmo. En el segundo ocurre a la inversa ya que se considera que todos los datos pertenecen a un mismo conglomerado y conforme avanza el algoritmo los grupos se van dividiendo formando otros nuevos. Para decidir qué clusters se combinan (aglomerativo) o se dividen (divisivo) hay que tomar una función de enlace $D(C, C')$ que define la disimilitud entre dos clusters C y C'. Así, para el clustering jerárquico aglomerativo, por ejemplo, se empieza con n clusters distintos y, de forma recursiva, se van mezclando los dos grupos más parecidos, es decir, aquellos que minimizan la función $D(C, C')$.

Algunas funciones de enlace clásicas para el agrupamiento jerárquico son por ejemplo, la función de enlace simple o método del vecino más cercano $D(C, C') = \text{mín}\{d(a, b), a \in C, b \in C'\}$ y la función de enlace completa o método del vecino más alejado $D(C, C') = \text{máx}\{d(a, b), a \in C, b \in C'\}$, donde $d(\cdot, \cdot)$ es la distancia que se toma entre dos observaciones (la disimilitud). En Hu (2015) se propone una función de enlace concreta para los modelos de mixturas finitas, cuya finalidad es proporcionar una buena partición de los datos para iniciar el algoritmo EM. La función es la siguiente

$$D(C, C') = - \triangle L_C(\Psi|Y)$$

siendo

$$L_C(\Psi|Y) = \prod_{t=1}^{n} \prod_{i=1}^{K} f_i(y_t|\Psi)^{z_{ti}} \tag{2.20}$$

y donde $\triangle L_C(\Psi|Y)$ representa el cambio de (2.20) si el clúster C y C' se mezclan.

La expresión (2.20) es una aproximación de la función de verosimilitud (2.4), por lo que la partición que maximiza (2.20) será una buena inicialización del algoritmo EM.

En general, para garantizar un buen clustering jerárquico se usan distintas disimilitudes y se comprueba si la clasificación que se obtiene se mantiene.

Otra posible estrategia del clustering de datos es ejecutar primero un método jerárquico, que no requiere la especificación del número de grupos, para definir precisamente este número de clusters, y posteriormente utilizar el método *K-means* para realizar la clasificación de los datos.

2.5.2. Variantes del algoritmo EM

Existen infinidad de modificaciones del algoritmo EM en función de los datos con los que se trabaja. A continuación se plantean algunas de las encontradas en los artículos revisados: el CEM, el SEM, el CAEM y el SAEM.

- **Algoritmo *Classification Expectation-Maximization* (CEM)**

El algoritmo CEM añade al algortimo EM un paso intermedio entre el paso E y el paso M, llamado el paso C. En él se asigna cada dato observado y_t al clúster que proporcione la mayor probabilidad a posteriori calculada en el paso anterior $\hat{\tau}_{it}^{(m)}$. Si este no es único, se escoge el clúster con menor subíndice (Celeux and Govaert, 1992). Mediante este nuevo paso, se obtiene una partición de los datos $P_1^{(m)}, \ldots, P_K^{(m)}$ en cada iteración del algoritmo, que denotaremos por $P^{(m)}$. Se tiene entonces que $y_t \in P_{i_0}^{(m)}$ si y sólo si $\hat{\tau}_{i_0 t}^{(m)} = \max\limits_{i \in \{1,\ldots,K\}} \hat{\tau}_{it}^{(m)}$ para $t = 1, \ldots, n$.

En el paso M, se calcula la estimación de máxima verosimilitud para $\Psi^{(m+1)}$ usando la partición obtenida $\left(P^{(m)}\right)$. De forma análoga a la obtención de la expresión (2.13), se tiene que

$$\hat{\pi}_i^{(m+1)} = \frac{\sum\limits_{y_t \in P_i^{(m)}} \hat{\tau}_{it}^{(m)}}{n} = \frac{\#P_i^{(m)}}{n}$$

donde # designa el cardinal del conjunto.

La estimación de θ_i dependerá de las distribución de las componentes. A continuación veremos el ejemplo de la estimación de $\theta_i = (\mu_i, \sigma_i^2)$ de un GMM.

Ejemplo 2.7. A partir de (2.14) y (2.15), se tiene que

$$\hat{\mu}_i^{(m+1)} = \frac{\sum_{y_t \in P_i^{(m)}} y_t}{\#P_i^{(m)}}$$

$$\left(\hat{\sigma}_i^2\right)^{(m+1)} = \frac{\sum_{y_t \in P_i^{(m)}} |y_t - \hat{\mu}_i^{(m+1)}|^2}{\#P_i^{(m)}}$$

Observación 2.8. Para modelos de mixturas gausianas se tiene que el algortimo CEM se corresponde con el algoritmo EM inicializado mediante el método de clustering *K-means* (Celeux and Govaert, 1992).

También en Celeux and Govaert (1992) existe un resultado que garantiza, bajo ciertas hipótesis, la convergencia lineal del algoritmo a un máximo local. Si tras realizar varias ejecuciones del algoritmo, se obtiene el mismo agrupamiento de los datos se puede decir con cierta confianza que se ha obtenido el óptimo global de la función de log-verosimilitud.

- **Algoritmo *Stochastic Expectation-Maximization* (SEM)**

Se trata de una modificación estocástica del algortimo CEM, también propuesta por Celeux and Govaert (1992). El paso intermedio que se añade en este caso se llama paso S y en él se asigna al azar cada y_t a los clúster $P_1^{(m)}, \ldots, P_K^{(m)}$, es decir, se genera una muestra aleatoria $\left(y_1, z_{1i}^{(m)}\right), \ldots, \left(y_n, z_{ni}^{(m)}\right)$.

Debido a este paso estocástico se garantiza que el algoritmo no se detenga al alcanzar el primer óptimo local, es decir, en cada iteración existe un probabilidad no nula de aceptar el parámetro actualizado $\Psi^{(m+1)}$ a pesar de proporcionar menor verosimilitud que el parámetro anterior $\Psi^{(m)}$ (Celeux et al., 1995). También a causa de este paso aleatorio la sucesión $\{\Psi^{(m)}\}$ no converge puntualmente. Se trata de una cadena de Markov, que en caso de ser regular, se tiene que la sucesión converge cuando m tiende a infinito, a la única solución estacionaria de la cadena (Celeux and Govaert, 1992; Celeux et al., 1995).

En conclusión, el algoritmo SEM supera gran parte de las limitaciones comentadas del algoritmo EM como la convergencia lenta y la gran dependencia de los valores iniciales para obtener el óptimo. Especialmente elude soluciones subóptimas que suele proporcionar el algoritmo CEM pero resulta menos fiable que este para muestras de pequeño tamaño.

- **Algoritmo *Classification Annealing Expectation-Maximization* (CAEM)**

El algoritmo CAEM se basa en el algoritmo SEM y en algoritmos de recocido simulado (*Simulated annealing* (SA)) (Celeux and Govaert, 1992). Los algoritmos de recocido simulado permiten resolver de forma aproximada grandes problemas de optimización combinatoria. Gracias al recocido simulado se consigue que, conforme el número de iteraciones del algoritmo aumenta ($m \to \infty$), la varianza de las asignaciones aleatorias decrezca y tienda a cero. Se define entonces una sucesión $\{a_m\}$ decreciente tal que $\{a_m\} \to 0$ conforme $m \to \infty$. Para el buen funcionamiento de un algoritmo que use SA, se requiere una convergencia lenta de la sucesión. Para simplificar, se construye la sucesión como $a_{m+1} = b \cdot a_m$ con $b \in [0,0,1]$ y $a_0 = 1$.

Paso A-E: Se calculan los valores $s_{it}^{(m)}$ para $i = 1, \ldots, K$ y $t = 1, \ldots, n$, que están asociados a las probabilidades a posteriori $\hat{\tau}_{it}^{(m)}$.

$$s_{it}^{(m)} = \frac{\left[\hat{\pi}_i^{(m)} \cdot f_i(y_t, \hat{\theta}_i^{(m)})\right]^{1/a_m}}{\sum_{l=1}^{K} \left[\hat{\pi}_l^{(m)} \cdot f_l(y_t, \hat{\theta}_l^{(m)})\right]^{1/a_m}}$$

Paso C: para $t = 1, \ldots, n$ se asigna de forma aleatoria cada y_t a uno de los clusters $P_1^{(m)}, \ldots, P_K^{(m)}$ con probabilidades $s_{it}^{(m)}$. La partición obtenida se denota por $P^{(m)}$.

Nótese que conforme m crece el CAEM va desde un algoritmo SEM puro ($a_m = 1$) hacia un algoritmo CEM puro ($a_m = 0$).

Como ocurría con el algoritmo SEM, el CAEM tampoco se detiene al alcanzar el primer óptimo local, superando así una de las limitaciones más importantes del algoritmo EM.

En el artículo de Celeux and Govaert (1992) se realizan comparaciones entre los tres algoritmos CEM, SEM y CAEM para un modelo de mixturas gausiano (GMM) y se concluye que los algoritmos SEM y CAEM parecen más eficientes que el algoritmo CEM a la hora de evitar soluciones subóptimas, pero ambos requieren mayor número de iteraciones y por tanto son computacionalmente más costosos. También en este artículo recomiendan el uso del algoritmo CAEM para muestras de tamaños pequeños y el SEM para aquellas más grandes, en especial cuando no existe una estructura clara del agrupamiento de los datos.

- **Algoritmo *Stochastic Annealing Expectation-Maximization* (SAEM)**

Se trata de una variante del SEM similar al CAEM, donde se toma una sucesión $\{a_m\}$ con las mismas características y se calculan unos parámetros $r_{it}^{(m)}$ que se consideran probabilidades artificiales a posteriori y sustituyen a $\hat{\tau}_{it}^{(m)}$ en la estimación de máxima verosimilitud del paso M.

En este algoritmo se incorporan dos pasos entre el paso E y el M, llamados el paso S y paso A.

Paso E: como en el algoritmo EM, se calculan las probabilidades a posteriori $\hat{\tau}_{it}^{(m)}$.

Paso S: se generan de forma aleatoria mediante una distribución multinomial de parámetro $\hat{\tau}_{it}^{(m)}$ una muestra para la variable indicadora Z_{ti} para cada iteración $m > 0$, es decir, se tiene $\left(y_1, z_{1i}^{(m)}\right), \ldots, \left(y_n, z_{ni}^{(m)}\right)$ para $i = 1, \ldots, K$. Si existe i tal que $\#\{y_t,\, Z_{ti}^{(m)} = 1\} \leq \frac{q+1}{n}$ el algoritmo se reinicia tomando en este caso $K - 1$ componentes y si no, se continua con el paso A.

Paso A: consiste en calcular unas cantidades $r_{it}^{(m)} = \hat{\tau}_{it}^{(m)} + a_m \cdot \left(Z_{ti}^{(m)} - \hat{\tau}_{it}^{(m)}\right)$ que se considerará como la probabilidad de que y_t provenga de la i-ésima componente.

Paso M: se sustituye el parámetro $\hat{\tau}_{it}^{(m)}$ por $r_{it}^{(m)}$ y se obtiene a partir de (2.13) que

$$\hat{\pi}_i^{(m+1)} = \frac{\sum_{t=1}^{n} r_{it}^{(m)}}{n}$$

Según Celeux et al. (1995) se tiene la siguiente relación del SAEM con el algoritmo EM y SEM

$$\Psi^{(m+1)} = (1 - a_{m+1}) \cdot \Psi_{\text{EM}}^{(m+1)} + a_{m+1} \cdot \Psi_{\text{SEM}}^{(m+1)}$$

donde $\Psi_{\text{EM}}^{(m+1)}$ y $\Psi_{\text{SEM}}^{(m+1)}$ son las sucesiones que se obtendrían en el m-ésimo paso del algoritmo EM y SEM, respectivamente.

Nótese que el algoritmo SAEM, conforme m aumenta, va de un SEM puro ($a_m = 1$) hacia un EM puro ($a_m = 0$).

En Celeux et al. (1995) se garantiza la convergencia casi segura del algoritmo a un máximo local tomando una sucesión $\{a_m\}$ tal que $\lim_{m\to\infty} \frac{a_m}{a_{m+1}} = 1$ y $\sum_m a_m = \infty$.

Como ocurría con el algoritmo SEM y CAEM, también el SAEM elude soluciones subóptimas. Además, como el CAEM, corrige el comportamiento errático del SEM en muestras pequeñas.

2.5.3. Estudio de otras estrategias de inicialización del algoritmo EM

A continuación se introducen algunos de los métodos de inicio del algoritmo estudiados en Scharl et al. (2009):

- **Inicialización aleatoria (RndEM)**: consiste en ejecutar EM p veces con valores iniciales aleatorios y seleccionar aquellos de los p posibles que maximizan la función de log-verosimilitud.
- **Algoritmo CEM**: consiste en ejecutar p veces el algoritmo CEM con valores iniciales aleatorios y escoger aquellos que proporcionan el mayor valor de la función de log-verosimilitud. Posteriormente, se usan estos valores iniciales para el algoritmo EM.
- **Algoritmo SEM**: el procedimiento es el mismo que en el caso anterior pero con el algoritmo SEM.
- **Ejecuciones cortas del EM (emEM)**: se ejecuta EM p veces para valores iniciales aleatorios sin esperar a que se produzca la convergencia del algoritmo. El criterio de parada usado es $|l_q - l_{q-1}|/(|l_q| + 0{,}1) < tol$, donde tol es la tolerancia y se fija de antemano y l_q es la función de log-verosimilitud en la q-ésima iteración. Se escogen los valores que maximizan la función de log-verosimilitud. Los K puntos elegidos sirven de valores iniciales de los clusters para ejecutar el algoritmo EM.
- **Muestreo**: consiste en aplicar el algoritmo EM p veces para c muestras pequeñas y aleatorias de los datos y aplicar el modelo de estimación resultante al conjunto completo de los datos.
- **Método incremental**: consiste en extraer muestras aleatorias de los datos, seleccionar el modelo que subestima el número de componentes de la mixtura, y posteriormente, extender el modelo al conjunto completo de los datos. Se ejecuta el método incremental en c muestras con un número de clusters fijado de antemano. El algoritmo se detiene si, al aumentar el número de componentes, el valor de las funciones de log-verosimilitud no crece o si se ha alcanzado un número máximo de componentes prefijado.

  Como con la estrategia anterior, se ejecuta el algortimo EM p veces y se toman como valores de inicio aquellos que maximizan la función de log-verosimilitud.

Las conclusiones del dicho artículo son que: la estrategia de muestreo y el método incremental son computacionalmente costosos y los resultados obtenidos son similares a otras estrategias más rápidas; la inicialización aleatoria (RndEM) también requiere un tiempo de ejecución alto comparado con las estrategias CEM y emEM y los resultados de clustering son similares; usar como métodos de obtención de los valores iniciales del algoritmo el CEM y SEM para posteriormente ejecutar el algortimo EM apenas mejora el clustering en relación al obtenido aplicando los dos métodos directamente; y por último, que el emEM proporciona buenos resultados en relación a la calidad del clustering y el tiempo de ejecución empleado.

También en el estudio realizado por Biernacki et al. (2003) se comparan las estrategias de inicialización RndEM, emEM, CEM y SEM. Concluyen que las diferencias entre estos métodos de inicio no son muy significativas pero que, generalmente, el método emEM proporciona mejores resultados que los demás. En este artículo también se constata que el algoritmo CEM es menos estable que el emEM y el SEM debido a su fuerte dependencia a sus valores iniciales, como se comentaba en Celeux and Govaert (1992).

Sin embargo en Baudry and Celeux (2015), se obtiene que una inicialización de tipo SEM proporciona mejores resultados que cualquiera de las otras tres estrategias, incluida la inicialización emEM.

En Michael and Melnykov (2016) se propone otra estrategia de inicialización llamado algoritmo **emaEM** y se trata de una mejora del emEM. La estrategia emEM tiene limitaciones cuando la muestra es grande y el número de clusters también lo es. La estrategia de inicialización emaEM consiste en:

1) Para unos puntos K dados y un número fijo de modelos m, se realizan m ejecuciones cortas del algoritmo EM y se obtienen $M_1, \ldots, M_m$ posibles modelos de los datos y una partición $P_{j1}, \ldots, P_{jn}$ para cada modelo M_j con $j = 1, \ldots, m$. Se calculan los criterios de información bayesianos para cada modelo usando la fórmula (2.18), es decir, se calculan $BIC_{M_1}, \ldots, BIC_{M_m}$.

2) Se calculan los valores de w_j mediante la siguiente expresión

$$w_j = \frac{e^{\frac{-BIC_{M_j}}{2}}}{\sum_{j=1}^{m} e^{\frac{-BIC_{M_j}}{2}}}$$

También se podrían tomar $w_j = \frac{1}{m}$ para $j = 1, \ldots, m$.

3) Se define la siguiente función: $h_j(i,k) = \begin{cases} 1 & \text{si } i,k \in P_{jt} \text{ para algún } t = 1, \ldots, n. \\ 0 & \text{en otro caso} \end{cases}$

4) Se crea la matriz $A = (a_{ik})$ con $a_{ik} = \sum_{j=1}^{m} w_j \cdot h_j(i,k)$. Se tiene que A es una matriz simétrica $n \times n$ y contiene los pesos que indican cuan a menudo dos datos (i,k) están en el mismo clúster teniendo en cuenta todos los modelos.

5) Se usa un clustering jerárquico tomando A como distancia métrica o disimilitud y la función de enlace que queramos. Así se obtiene un vector de particiones para un K específico.

6) Se usa la partición obtenida en el paso anterior para inicializar el algoritmo EM.

En el capítulo computacional se usarán y contrastarán cuatro de las estrategias de inicio estudiadas en esta sección: una inicialización aleatoria (RndEM), una inicialización usando ejecuciones cortas del algoritmo (emEM) y la inicialización mediante las variantes CEM y SEM del algoritmo. Para la elección de estas cuatro estrategias de inicialización se ha tenido en cuenta que:

- Las cuatro estrategias son ampliamente usadas y se encuentran implementadas en el paquete seleccionado para este proyecto (FlexMix), así como en otros muchos paquetes de R. Algunos ejemplos de uso de estas estrategias se pueden encontrar en Baudry and Celeux (2015), Scharl et al. (2009), Biernacki et al. (2003), Celeux and Diebolt (1989) y Celeux and Govaert (1992).
- La muestra con la que finalmente se trabaja no es excesivamente pequeña, 53 observaciones frente a 10 variables, por lo que el algoritmo SEM no tendrá un comportamiento errático y proporcionará iguales o mejores resultados que el CEM, CAEM y SAEM (Celeux and Govaert, 1992; Celeux and Diebolt, 1989).
- Las conclusiones de los artículo de Scharl et al. (2009) y Biernacki et al. (2003) estipulan que, a pesar de no existir grandes diferencias entre los métodos considerados en los respectivos estudios, los mejores resultados se obtienen con el algoritmo emEM, y posteriormente con el SEM y CEM.
- Se considera la inicialización aleatoria del algoritmo para comparar el comportamiento de este frente a los demás métodos de inicio.

2.6. Evaluación de la clasificación. Curvas ROC

Las curvas ROC son una de las herramientas estadísticas más usadas para valorar la precisión de una prueba diagnóstica dicotómica, esto es, una prueba basada en una variable de decisión D que sigue una distribución de Bernouilli y toma valor $D = 1$ si el individuo está enfermo y $D = 0$ si está sano. Se dice que un individuo da positivo en la prueba si ésta considera que está enfermo y negativo si considera que no lo está (Valle-Benavides, 2017). Por tanto, resulta una buena herramienta para valorar la clasificación de los datos generada por modelos de mixturas de dos componentes, que son los que consideraremos en el siguiente capítulo.

La prueba entonces realiza un contraste de hipótesis donde se toma como hipótesis nula que el individuo esté sano $H_0 : D = 0$ y como hipótesis alternativa que esté enfermo $H_1 : D = 1$. Recordemos que en el contraste de hipótesis el error de tipo I se comete cuando se rechaza la hipótesis nula, siendo esta verdadera (falso positivo) y el error de tipo II cuando se acepta la hipótesis nula siendo esta falsa (falso negativo). Podemos expresar estos dos errores como $P(H_1|H_0)$ y $P(H_0|H_1)$ respectivamente. Se busca que las probabilidades de ambos tipos de error sean pequeñas (Vivo et al., 2017).

Al aplicar esta prueba se genera una variable aleatoria X que divide la población en dos grupos: sanos y enfermos. Existen, por tanto, cuatro posibles resultados: que la prueba clasifique como enfermo a un individuo enfermo (verdadero positivo que denotaremos por V_+), que clasifique como enfermo a un individuo sano (falso positivo denotado por F_+), que clasifique como sano a un individuo enfermo (falso negativo F_-) y como sano a un individuo sano (verdadero negativo V_-).

El buen funcionamiento de este tipo de test se mide a través de la especifidad o valor predictivo negativo y la sensibilidad o valor predictivo positivo (Valle-Benavides, 2017).

- La **especificidad** se corresponde con la probabilidad de que dado un sujeto sano, este haya sido clasificado como sano, es decir, $P(X = sano|H_0) = \frac{V_-}{V_-+F_+}$. Nótese que los verdaderos negativos y falsos positivos suponen la totalidad de los sujetos sanos ($V_- + F_+ =$ total sanos), y los verdaderos negativos los sanos detectados por la prueba. Por tanto, la especificidad mide la capacidad de nuestra prueba-diagnóstico de detectar correctamente los sujetos sanos.

  La fracción de falsos positivos o FPR se define como 1− especificidad, es decir, $1 - P(X = sano|H_0) = P(X = enfermo|H_0) = \frac{F_+}{V_-+F_+}$, que se corresponde con la probabilidad de cometer un error de tipo I.

- La **sensibilidad** es la probabilidad de que dado un sujeto enfermo la prueba lo clasifique como tal, es decir, $P(X = enfermo|H_1) = \frac{V_+}{V_++F_-}$. Nótese que los verdaderos positivos y falsos negativos suponen la totalidad de los sujetos enfermos ($V_+ + F_- =$ total enfermos), y los verdaderos positivos los enfermos que detecta la prueba. Por tanto, la sensibilidad mide la capacidad de nuestra prueba-diagnóstico de detectar correctamente la enfermedad en sujetos enfermos.

  La fracción anterior se define como la fracción de verdaderos positivos o TPR y se corresponde con el complementario de la probabilidad de cometer un error de tipo II.

Observación 2.9. Se puede comprobar el buen funcionamiento de una prueba calculando su valor predictivo global (GR), que es la proporción de resultados válidos obtenidos en la prueba

$$GR = \frac{V_+ + V_-}{V_+ + F_+ + V_- + F_-}$$

Cuando la variable generada al realizar la prueba, X, es continua, se requiere un umbral c para clasificar los resultados como sanos o enfermos. En este caso las fracciones FPR y TPR son funciones definidas para todo posible valor de c como

$$FPR(c) = P(X > c|H_0) \qquad \text{y} \qquad TPR(c) = P(X > c|H_1)$$

Definición 2.10. La curva ROC representa los pares $(FPR(c), TPR(c))$ para cada posible valor umbral c, es decir, representa la 1−especificada frente a la sensibilidad de la prueba. Se puede escribir como

$$ROC(t) = TPR(FPR^{-1}(t)) \quad \text{para } t \in [0,1] \tag{2.21}$$

Gráficamente con una curva ROC se observa el balance entre la sensibilidad y la especificidad de la prueba para todos los posibles umbrales, es decir, ilustra la proporción de verdaderos positivos (eje Y) frente a la proporción de falsos positivos (eje X).

En un gráfico de curvas ROC se representa también la diagonal de referencia o línea de no-discriminación. Esta línea se corresponde con la curva ROC que describiría una prueba incapaz de discriminar entre individuos sanos y enfermos, es decir, aquel en el que existe la misma proporción de verdaderos y falsos positivos. Por tanto, un test diagnóstico será más efectivo conforme su curva ROC se encuentre lo más alejada de la diagonal.

Definición 2.11. Se define la razón de verosimilitud positiva (PLR) (Silva and Molina, 2017) como la probabilidad de que una persona enferma haya sido diagnosticada como tal entre la probabilidad de que una persona sana haya sido clasificada como enferma, es decir,

$$\frac{P(X = enfermo|H_1)}{P(X = sano|H_1)} = \frac{\text{sensibilidad}}{1 - \text{especificidad}} = \frac{ROC(t)}{t}$$

La razón de verosimilitud negativa (NLR) es la probabilidad de que un individuo enfermo haya sido clasificado como sano dividido entre la probabilidad de que un individuo sano haya sido clasificado como tal, es decir,

$$\frac{P(X = enfermo|H_0)}{P(X = sano|H_0)} = \frac{1 - \text{sensibilidad}}{\text{especificidad}} = \frac{1 - ROC(t)}{1 - t}$$

Observación 2.12. Habitualmente las curvas ROC son cóncavas, pero esto no es una condición necesaria. Por definición se tiene que son funciones no decrecientes, pero en algunos casos pueden presentar trozos de curva bajo la línea de no-discriminación. Cuando esto ocurre se denominan curvas ROC impropias.

Definición 2.13. El área bajo la curva ROC (AUC) es un estadístico que se usa para medir la capacidad discriminante de la prueba y comparar pruebas entre sí y saber cuál es más eficaz (Valle-Benavides, 2017). Se define como

$$AUC = \int_0^1 ROC(t)dt = \int_0^1 TPR\left(FPR^{-1}(t)\right) dt$$

Se tiene que cuanto más se aproxime el valor AUC a 1, mayor será su capacidad discriminativa y por tanto, tendrá mayor eficacia.

Para curvas ROC cuyos puntos se encuentran por encima de la línea de no-discriminación (curvas ROC propias), el valor del área varía entre 0,5, correspondiente a una prueba sin capacidad discriminante, y 1, valor que determinaría que los dos grupos de población son bien detectados y diferenciados por la prueba. Por tanto, cuánto mayor sea el valor de AUC más eficaz se considera la prueba.

Otra medida interesante relativa a curvas ROC es el área entre dos valores FPR, que denotaremos FPR_1 y FPR_2, es decir, una porción del área bajo la curva ROC. Esta área representa el promedio de sensibilidad de todos los valores FPR entre FPR_1 y FPR_2.

2.6.1. Área bajo la curva y área parcial

Definición 2.14. Se define el área parcial bajo la curva entre dos fracciones de falsos positivos $FPR_1 < FPR_2$ como (Vivo et al., 2017)

$$pAUC = \int_{FPR_1}^{FPR_2} ROC(t)dt = \int_{FPR_1}^{FPR_2} TPR\left(FPR^{-1}(t)\right) dt$$

Es claro que $pAUC$ es un valor positivo y acotado superiormente por el área del rectángulo determinado por el intervalo (FPR_1, FPR_2), esto es, el rectángulo cuya altura es 1 y de base $FPR_2 - FPR_1$.

Para las curvas ROC propias (aquellas cuyos puntos se encuentran por encima de la diagonal) se tiene que el área parcial $pAUC$ está acotada inferiormente por el área del trapecio con vértices en $(FPR_1, 0)$, (FPR_1, FPR_1), $(FPR_2, 0)$ y (FPR_2, FPR_2). Se forma así un trapecio cuyas bases miden FPR_1 y FPR_2 respectivamente, y la altura $FPR_2 - FPR_1$. Por tanto, se tiene que

$$\frac{(FPR_2 + FPR_1) \cdot (FPR_2 - FPR_1)}{2} \leq pAUC \leq FPR_2 - FPR_1 \tag{2.22}$$

Para curvas ROC impropias, este límite inferior no siempre se satisface (gráficos 2 y 3 de la Figura 2.1). Sin embargo, para cualquier curva ROC siempre se satisface que el área parcial bajo la curva está acotada inferiormente por el área del rectángulo de base $FPR_2 - FPR_1$ y altura TPR_1.

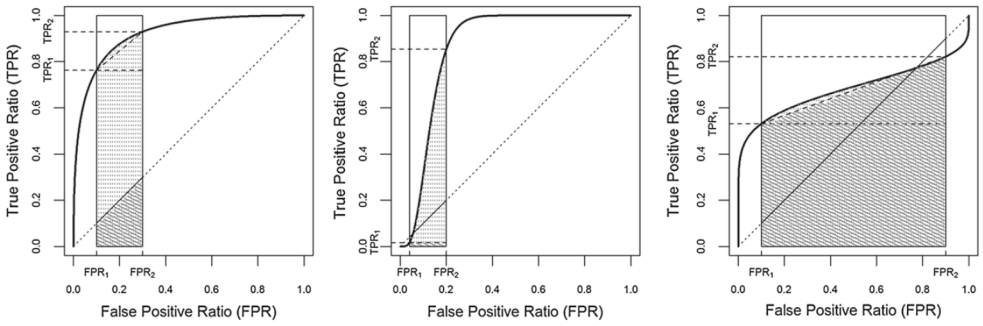

Figura 2.1: Curvas ROC propias e impropias (Vivo et al., 2017).

Si se observa el gráfico de la Figura 2.1, el límite superior se puede ajustar mejor en los tres casos, tomando el rectángulo de misma base $FPR_2 - FPR_1$ y altura TPR_2.

Por tanto, para cualquier curva ROC se tendrá que

$$TPR_1 \cdot (FPR_2 - FPR_1) \leq pAUC \leq TPR_2 \cdot (FPR_2 - FPR_1) \tag{2.23}$$

Si además la función $PLR(t)$ alcanza su mínimo local en el extremo superior FPR_2 (como ocurre en la primera y tercera imagen de 2.1), entonces el $pAUC$ estará acotado inferiormente por el trapecio con vértices $(FPR_1, 0)$, (FPR_1, TPR_1), $(FPR_2, 0)$ y (FPR_2, TPR_2), es decir, aquél con bases TPR_1 y TPR_2 y altura $FPR_2 - FPR_1$. Por tanto

$$\frac{(TPR_2 + TPR_1) \cdot (FPR_2 - FPR_1)}{2} \leq pAUC \leq TPR_2 \cdot (FPR_2 - FPR_1) \tag{2.24}$$

2.6.2. Índice del área parcial

Definición 2.15. Se define el índice estandarizado de $pAUC$ como

$$SpAUC = \frac{1}{2}\left(1 + \frac{pAUC - pAUC_{min}}{pAUC_{max} - pAUC_{min}}\right) \tag{2.25}$$

donde $pAUC_{min} = \frac{(FPR_2+FPR_1)\cdot(FPR_2-FPR_1)}{2}$ y $pAUC_{max} = FPR_2 - FPR_2$, es decir las cotas de $pAUC$ definidas en (2.22).

Este índice se usa para medir la capacidad de discriminación de la prueba.

Como se comenta previamente, la cota inferior $pAUC_{min}$ es sólo válida para curvas ROC propias, por lo que, para curvas ROC impropias el índice (2.25) no estaría bien definido. Este índice también presenta problemas cuando se comparan dos curvas ROC que se cruzan en el intervalo que se considera (FPR_1, FPR_2). Por estos dos motivos se propone en Vivo et al. (2017) un índice más estricto, que es definido a partir de (2.25), tomando en esta ocasión las acotación descrita en (2.23) y (2.24).

2.6.3. Índice ajustado del área parcial

Definición 2.16. Se define el índice ajustado del área parcial bajo la curva ROC como

$$TpAUC = \frac{1}{2}\left(1 + \frac{pAUC - pAUC^*_{min}}{pAUC^*_{max} - pAUC^*_{min}}\right)$$

donde $pAUC^*_{min}$ y $pAUC^*_{max}$ toman los valores de las acotaciones descritas en (2.23) o (2.24).

1) Para cualquier curva ROC se tiene por (2.23) que $pAUC^*_{min} = TPR_1 \cdot (FPR_2 - FPR_1)$ y $pAUC^*_{max} = TPR_2 \cdot (FPR_2 - FPR_1)$. Por tanto, este índice se puede expresar de la forma

$$TpAUC = 1 - \frac{TPR_2 \cdot (FPR_2 - FPR_1) - pAUC}{2 \cdot (TPR_2 - TPR_1) \cdot (FPR_2 - FPR_1)}$$

siempre que $TPR_1 \neq TPR_2$.

2) Si la función $PLR(t)$ alcanza su mínimo local en el extremo superior FPR_2, entonces por (2.24) se tiene que $pAUC^*_{min} = \frac{(TPR_2+TPR_1)\cdot(FPR_2-FPR_1)}{2}$ y $pAUC^*_{max} = TPR_2 \cdot (FPR_2 - FPR_1)$. La expresión del índice es de la forma

$$TpAUC = \frac{pAUC - TPR_1 \cdot (FPR_2 - FPR_1)}{(TPR_2 - TPR_1) \cdot (FPR_2 - FPR_1)}$$

siempre que $TPR_1 \neq TPR_2$.

A continuación expondremos un ejemplo propuesto en Vivo et al. (2017) en el que el valor del área parcial bajo la curva no es suficiente para determinar si una prueba u otra es más eficaz.

Ejemplo 2.17. En este caso la prueba consiste en determinar si un gen u otro es mejor biomarcador del cáncer de ovario. Es preciso que los valores FPR sean bajos dado que en este caso se necesita que la proporción de individuos sanos diagnosticados con cáncer sea mínimo. En el artículo de Pepe et al. (2003) se justifica que se toma un valor máximo para FPR pequeño, pero no tanto como para no conseguir estimar adecuadamente el valor de $pAUC$.

Se toman los datos referentes al gen 645 y al gen 1158, cuyas curvas ROC se cruzan en el intervalo $(0, 0{,}1)$. Se calcula el valor de $pAUC$ en este intervalo y se obtiene que para ambos genes vale lo mismo.

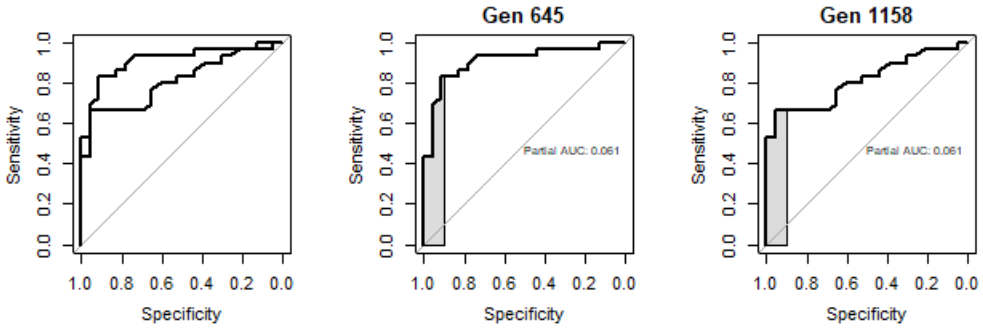

Figura 2.2: Curvas ROC del gen 645 y 1158.

Por tanto, se requiere de otro índice que permita determinar en el intervalo de interés, qué gen es mejor biomarcador para el cáncer, es decir, qué test diagnóstico es más eficaz.

Si calculamos los índices ajustados $TpAUC$, se obtiene que $TpAUC_{g645} = 0{,}7304348$ y $TpAUC_{g1158} = 0{,}9130435$, por lo que, entre los dos, elegiríamos el segundo gen como mejor biomarcador.

Observación 2.18. El paquete de R usado para la representación de las curvas ROC se llama *pROC* y por defecto representa la especificidad frente a la sensibilidad, en lugar del FPR (1-especificidad) frente a la sensibilidad. Por ello, el área bajo la curva en el intervalo $(0, 0{,}1)$ se corresponde con el área bajo la curva en el intervalo $(0{,}9, 1)$.

Capítulo 3

Aplicación al clustering de datos de microarray mediante mixturas finitas

El objetivo de este capítulo es poner en práctica la teoría desarrollada en los capítulos previos, usando la base de datos del artículo de Pepe et al. (2003). En ella se recogen las expresiones de varios genes de distintos tejidos ováricos cancerosos y sanos. Se pretende estudiar la capacidad de predicción de los mejores biomarcadores genéticos del cáncer ovárico propuestos en dicho artículo mediante modelos de mixturas y usando el algoritmo EM para la estimación de los parámetros. En especial, se estudiará la eficacia de los distintos métodos de inicialización del algoritmo y los diferentes criterios de selección del modelo.

En primer lugar, se introduce el concepto de microarrays, una herramienta muy potente en el campo de la genética que facilita, entre otras cosas, el estudio simultáneo de la expresión genética de distintos tejidos. La base de datos utilizada se ha obtenido mediante el uso de un microarray por lo que se pueden estudiar las diferencias entre las expresiones de un mismo gen para tejidos enfermos y sanos.

Posteriormente se desarrolla la puesta en práctica descrita previamente. Se usará el software estadístico R, concretamente el paquete FlexMix que permite clasificar los datos usando modelos de mixturas e implementar el algoritmo EM con distintas inicializaciones. También se introducirán algunas herramientas útiles implementadas en dicho paquete que permiten obtener las estimaciones de los parámetros de las mixturas o comprobar la existencia de solapamiento entre los grupos obtenidos.

Para finalizar, se usarán algunos de los conceptos relativos a curvas ROC, con el objetivo de comprobar la efectividad de clasificación de los modelos y también su estabilidad usando técnicas de remuestreo o bootstrap.

3.1. Introducción a los microarrays

Un microarray es un micro-chip que contiene fragmentos de material genético que permite el estudio simultáneo de los distintos niveles de expresión de miles de genes, por lo que, por ejemplo, se pueden examinar los datos relativos a una célula de una muestra sana y otra enferma a la vez. Se comparan por tanto las expresiones de distintos genes de dos cepas, una de ellas de control.

Cada gen se representa por un punto en el micro-chip. Un punto negro significa que el gen no se expresa, un punto amarillo que los genes de ambas cepas se expresan igual, uno verde que hay más genes expresados de la cepa de estudio que de la de control y un punto rojo que hay más genes expresados de la cepa de control. La siguiente imagen representa un microchip con la expresión de cada gen

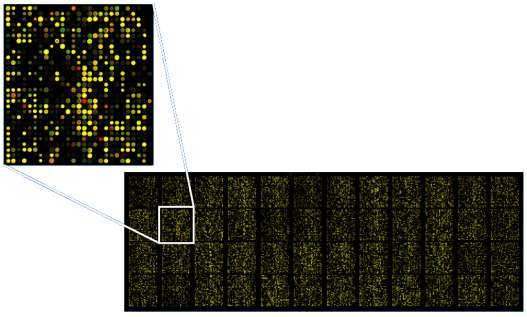

Figura 3.1: Microchip de ADN.

Cada punto del chip tiene un valor numérico para cada intensidad de color, obteniendo así un conjunto de datos numéricos (Vijverberg, 2007).

Una de las mayores ventajas de los microarrays es poder analizar y visualizar tal cantidad de datos para una muestra de forma simultánea. El mayor inconveniente es que en las investigaciones habituales se tiene un número limitado de muestras, por lo que se obtiene un conjunto de datos con miles de variables y sólo unas pocas observaciones, dificultando en gran medida el proceso de análisis de los resultados. Por ello se usan técnicas de clustering de datos y selección de los genes considerados mejores biomarcadores genéticos según el tipo de muestra analizada.

Algunos ejemplos propuestos en López et al. (2006) de aplicación de los microarrays en salud humana son:

- identificación de los mejores biomarcadores genéticos de cada enfermedad. Como hemos visto, los microarrays permiten comparar los distintos niveles de expresión de genes en tejidos sanos y enfermos a la vez, lo que facilita la identificación de los genes implicados en la enfermedad.

- el posible diagnóstico clínico a nivel molecular. Facilitan la detección de mutaciones genéticas asociadas a una enfermedad concreta y permiten, por ejemplo, clasificar mejor los tipos de tumores y así tener más información sobre su posible tratamiento y evolución. Se pueden usar para identificar microorganimos patógenos en el individuo, tales como virus, bacterias u hongos, e investigar los mecanismos de resistencia de algunos de ellos ante los antibióticos.

- el descubrimiento y desarrollo de fármacos, dado que permiten observar los cambios de la expresión genética durante la administración de un fármaco y realizar ensayos de toxicidad, seguridad y efectividad de los mismos.

- el desarrollo de la farmacogenómica. La farmacogenómica busca aplicar la tecnología genómica en el desarrollo de fármacos y terapias individualizadas. Su objetivo es identificar las enfermedades a nivel molecular y desarrollar estrategias más efectivas en su cura según el perfil genómico del paciente. Esto haría posible el desarrollo de tratamientos personalizados y también conocer la predisposición individual de padecer una enfermedad, permitiendo el desarrollo de una medicina preventiva adecuada.

3.2. Base de datos y modelización

En esta sección, se analiza una base de datos real con el objetivo de realizar un agrupamiento de los datos usando los conocidos modelos de mixturas finitas. Se utiliza el paquete FlexMix, que implementa el algoritmo EM en modelos de regresión de mixturas finitas, y se estudian dos problemáticas clave de este algoritmo: cómo determinar el número de componentes o clusters usando criterios de selección de modelos, y qué método de inicio utilizar para garantizar la convergencia a la solución óptima. A continuación, sobre los modelos seleccionados, se estudia el solapamiento de sus componentes, la estimación de sus parámetros y la eficacia de clasificación del modelo, sobre los datos y sobre una muestra generada de forma aleatoria a partir del mismo.

La base de datos que se usará es la correspondiente al artículo Pepe et al. (2003). Esta recoge las observaciones de 1536 genes en 53 tejidos ováricos de los cuales se sabe que 23 son cancerosos y 30 sanos. Los datos no recogidos en el microarray son substituidos por valores cero.

Se pretende clasificar los tejidos en cancerosos y no cancerosos mediante modelos de mixturas. En el artículo Pepe et al. (2003), de los 100 primeros genes se escogen los diez mejores biomarcadores siguiendo cuatro criterios distintos. Un buen biomarcador es un gen cuya expresión indica en cierta medida si el tejido está enfermo. Los cuatro criterios usados para la selección de biomarcadores en Pepe et al. (2003) son: según el valor de la curva ROC en 0,1, ROC(0,1); según el índice que indica el área bajo la curva ROC en el intervalo $(0, 0{,}1)$, pAUC(0,1); el valor del área bajo la curva completa, AUC; y el índice Z-stat, que explicaremos a continuación.

Como hemos visto en un ejemplo previo se exige que FPR tome valores bajos para que la proporción de individuos sanos diagnosticados con cáncer sea ínfima. En el artículo que tomamos de referencia se justifica que se toma un valor máximo para FPR pequeño, pero no demasiado como para no ser capaces de estimar adecuadamente el valor de pAUC.

En el artículo también se considera el índice Z-stat o t-estadístico de dos muestras, que se define para cada gen $t = 1, \ldots, n$ (en este caso $n = 1536$) como

$$\text{Z-stat}_t = \frac{\overline{x}_{2t} - \overline{x}_{1t}}{\sqrt{\frac{s_{1t}^2}{n_1} + \frac{s_{1t}^2}{n_2}}}$$

siendo $\overline{x}_{1t}$, $\overline{x}_{2t}$, s_{1t}^2 y s_{2t}^2 las medias y varianzas de los dos grupos en los que se divide la muestra (tejidos canceroso y no canceroso) y n_1 y n_2 los tamaños correspondientes, en este caso $n_1 = 23$ y $n_2 = 30$.

Se considera que los genes con índice Z-stat mayor en valor absoluto son mejores biomarcadores ya que se produce una mayor diferencia en su expresión para los tejidos enfermos y sanos (Marín, 2018).

En el artículo Vivo et al. (2017) se propone un nuevo índice que está bien definido incluso para curvas ROC impropias llamado *Tighter Partial Area Index*, TpAUC, del que hemos hablado en el capítulo anterior. Además de los genes seleccionados por los otros cuatro índices, usaremos aquellos seleccionados por el índice TpAUC, también de entre los 100 primeros de la muestra. La función que determina dichos genes se puede encontrar en los ficheros suplementarios del mencionado artículo.

Tomamos entonces los genes seleccionados por los índices ROC, pAUC, AUC, Z-stat y TpAUC, según hemos descrito previamente.

ROC	pAUC	AUC	Z-stat	TpAUC
g93	g93	g93	g93	g93
g76	g65	g42	g65	g34
g65	g5	g76	g42	g25
g42	g23	g65	g97	g10
g5	g42	g16	g39	g23
g16	g51	g5	g23	g51
g39	g52	g52	g35	g73
g35	g35	g97	g76	g84
g23	g73	g39	g63	g52
g52	g76	g75	g5	g5

Tabla 3.1: Genes seleccionados por los distintos criterios.

Suponemos que cada gen es una variable independiente y que puede ser modelizado por un modelo de mixturas. Trabajaremos suponiendo que no disponemos de la información necesaria para saber si un tejido es o no canceroso por lo que aplicaremos el algoritmo EM para determinar el número de componentes de la mixtura de los datos de cada gen. Usaremos distintos métodos de inicialización y escogeremos los modelos con los criterios de selección AIC, BIC e ICL.

El modelo teórico de los datos es un modelo de mixturas de dos componentes o clusters para cada variable independiente o gen, una que determina los tejidos cancerosos y otra para los sanos; y con pesos similares ya que el número de muestras de tejidos sanos y cancerosos es de 30 y 23 respectivamente.

Aplicaremos el comando *initFlexmix* o *stepFlexmix* del paquete FlexMix para mixturas gamma y posteriormente normales, a cada uno de los cinco conjuntos de genes seleccionados, y comprobaremos el ajuste de los datos a los modelos de mixturas de 1 a 5 componentes, con distintos métodos de inicio. Posteriormente seleccionaremos para cada método de inicio el mejor modelo en cada caso mediante el criterio AIC, BIC e ICL y analizaremos los resultados obtenidos.

En primer lugar definimos el modelo de los datos con los que trabajamos. La fórmula que debemos introducir para ejecutar los comandos del paquete FlexMix tiene la siguiente estructura $y \sim x|g$, donde y es la variable respuesta o dependiente, x la variable predictora

o explicativa y g es un factor de agrupamiento opcional para mediciones repetidas (Leisch, 2004). En nuestro caso, se tiene que para cada grupo disponemos de 10 genes, es decir, 10 variables respuesta independientes entre sí $Y = (Y_1, \ldots, Y_{10})$ y no hay variables predictoras. Por ser independientes, para un modelo de mixturas gamma, la función de densidad del modelo es el producto de las funciones de densidad de cada una de las variables cuya expresión está dada en (1.4).

De forma análoga, si trabajamos con un GMM, se tiene que el modelo de mixturas de Y se expresa como el productorio de las funciones de densidad de cada una de las variables cuya expresión también obtuvimos en el primer capítulo (1.3).

Para especificar esto sobre los comandos del paquete FlexMix se crea una lista de modelos para cada grupo compuesta por todas las funciones de densidad $f_i(y_t|\cdot)$ para $t = 1, \ldots, 10$. Por ejemplo, para el grupo seleccionado por el índice ROC, se tienen 10 genes, los dos primeros son el 93 y el 76. El modelo se corresponderá con una lista donde los dos primeros elementos sean las funciones de densidad del gen 93 y el 76.

Empezamos trabajando con distribuciones gamma, por ser más habituales en modelos de supervivencia dado que incluyen otras como la exponencial. Definimos los modelos correspondientes como una lista de variables cada una siguiendo una distribución gamma (ver Apéndice A). Cada modelo será de la forma

```
Modelo<-list(FLXMRglm(genes_grupo[,1]~., family="Gamma"),
             FLXMRglm(genes_grupo[,2]~.,family="Gamma"),
             FLXMRglm(genes_grupo[,3]~., family="Gamma"),
             FLXMRglm(genes_grupo[,4]~.,family="Gamma"),
             FLXMRglm(genes_grupo[,5]~., family="Gamma"),
             FLXMRglm(genes_grupo[,6]~.,family="Gamma"),
             FLXMRglm(genes_grupo[,7]~., family="Gamma"),
             FLXMRglm(genes_grupo[,8]~.,family="Gamma"),
             FLXMRglm(genes_grupo[,9]~., family="Gamma"),
             FLXMRglm(genes_grupo[,10]~.,family="Gamma"))
```

Los puntos son reemplazados por las variables predictoras o explicativas x, de forma que estas sólo tienen que ser especificadas una vez, en lugar de 10 por cada uno. En nuestro caso, estamos ante un modelo sin variables explicativas y para indicarlo se le asigna el valor 1, como se propone en Grün and Leisch (2007). Tampoco disponemos de un factor de agrupamiento g, que por ser opcional, basta con no especificarlo.

Inicializamos el algoritmo EM con cuatro de los métodos de inicio vistos en el Capítulo 2, que son los más utilizados en la práctica y que ya están implementados en el paquete: inicio aleatorio, inicio con el método emEM, y usando las variantes CEM y SEM. Seleccionaremos los modelos con respecto al criterio AIC, BIC e ICL. Para cada uno de ellos se obtienen los siguientes modelos de mixturas:

- Para el criterio AIC

En las Tablas 3.2 y 3.3 se observa que para el grupo ROC el mejor modelo según el criterio AIC es el que tiene una inicialización de tipo CEM y para el AUC aquel con método de inicio aleatorio. Para los demás grupos de genes se seleccionan los modelos con una inicialización de tipo SEM.

	Rnd	emEM
ROC	prior size post>0 ratio Comp.1 0.0566 3 6 0.500 Comp.2 0.0962 5 21 0.238 Comp.3 0.4136 22 27 0.815 Comp.4 0.4335 23 24 0.958 'log Lik.' 34.57511 (df=83) AIC: 96.84978 BIC: 260.384	prior size post>0 ratio Comp.1 0.3061 17 28 0.607 Comp.2 0.4498 24 26 0.923 Comp.3 0.0566 3 3 1.000 Comp.4 0.1875 9 27 0.333 'log Lik.' 35.56681 (df=83) AIC: 94.86638 BIC: 258.4006
pAUC	prior size post>0 ratio Comp.1 0.1924 10 16 0.625 Comp.2 0.2184 11 25 0.440 Comp.3 0.0754 4 10 0.400 Comp.4 0.2594 14 19 0.737 Comp.5 0.2544 14 27 0.519 'log Lik.' 123.3824 (df=104) AIC: -38.76488 BIC: 166.1455	prior size post>0 ratio Comp.1 0.0949 5 18 0.278 Comp.2 0.4149 22 28 0.786 Comp.3 0.0566 3 3 1.000 Comp.4 0.4336 23 23 1.000 'log Lik.' 101.9947 (df=83) AIC: -37.98945 BIC: 125.5448
AUC	prior size post>0 ratio Comp.1 0.0755 4 9 0.444 Comp.2 0.1555 8 25 0.320 Comp.3 0.2978 16 19 0.842 Comp.4 0.1496 8 13 0.615 Comp.5 0.3216 17 25 0.680 'log Lik.' 7.84076 (df=104) AIC: 192.3185 BIC: 397.2288	prior size post>0 ratio Comp.1 0.464 25 27 0.926 Comp.2 0.134 7 20 0.350 Comp.3 0.403 21 28 0.750 'log Lik.' -56.54518 (df=62) AIC: 237.0904 BIC: 359.2485
Z-stat	prior size post>0 ratio Comp.1 0.292 15 28 0.536 Comp.2 0.111 6 13 0.462 Comp.3 0.167 9 20 0.450 Comp.4 0.431 23 26 0.885 'log Lik.' 70.01179 (df=83) AIC: 25.97642 BIC: 189.5107	prior size post>0 ratio Comp.1 0.133 7 16 0.438 Comp.2 0.126 7 14 0.500 Comp.3 0.359 19 24 0.792 Comp.4 0.235 12 26 0.462 Comp.5 0.147 8 11 0.727 'log Lik.' 95.92253 (df=104) AIC: 16.15495 BIC: 221.0653
TpAUC	prior size post>0 ratio Comp.1 0.0755 4 7 0.571 Comp.2 0.3069 16 41 0.390 Comp.3 0.2981 16 20 0.800 Comp.4 0.1505 8 23 0.348 Comp.5 0.1690 9 11 0.818 'log Lik.' 58.97552 (df=104) AIC: 90.04896 BIC: 294.9593	prior size post>0 ratio Comp.1 0.133 7 13 0.538 Comp.2 0.243 13 30 0.433 Comp.3 0.153 8 20 0.400 Comp.4 0.169 9 11 0.818 Comp.5 0.301 16 23 0.696 'log Lik.' 61.24195 (df=104) AIC: 85.5161 BIC: 290.4265

Tabla 3.2: Modelos para el criterio AIC.

	CEM	SEM
ROC	prior size post>0 ratio Comp.1 0.3401 18 24 0.750 Comp.2 0.0941 5 14 0.357 Comp.3 0.4343 23 28 0.821 Comp.4 0.1314 7 11 0.636 'log Lik.' 49.78276 (df=83) AIC: 66.43449 BIC: 229.9687	prior size post>0 ratio Comp.1 0.4710 25 28 0.893 Comp.2 0.1091 6 12 0.500 Comp.3 0.3443 18 26 0.692 Comp.4 0.0756 4 7 0.571 'log Lik.' 39.76673 (df=83) AIC: 86.46653 BIC: 250.0008
pAUC	prior size post>0 ratio Comp.1 0.3405 18 27 0.667 Comp.2 0.4434 24 26 0.923 Comp.3 0.1592 8 15 0.533 Comp.4 0.0569 3 5 0.600 'log Lik.' 96.94175 (df=83) AIC: -27.8835 BIC: 135.6507	prior size post>0 ratio Comp.1 0.0566 3 3 1.000 Comp.2 0.3501 18 31 0.581 Comp.3 0.4699 25 26 0.962 Comp.4 0.1234 7 12 0.583 'log Lik.' 104.116 (df=83) AIC: -42.23196 BIC: 121.3023
AUC	prior size post>0 ratio Comp.1 0.0848 4 19 0.211 Comp.2 0.0946 5 11 0.455 Comp.3 0.4041 22 24 0.917 Comp.4 0.4165 22 28 0.786 'log Lik.' -29.50803 (df=83) AIC: 225.0161 BIC: 388.5503	prior size post>0 ratio Comp.1 0.403 21 28 0.750 Comp.2 0.464 25 27 0.926 Comp.3 0.134 7 20 0.350 'log Lik.' -56.54532 (df=62) AIC: 237.0906 BIC: 359.2487
Z-stat	prior size post>0 ratio Comp.1 0.3288 17 27 0.630 Comp.2 0.1669 9 19 0.474 Comp.3 0.4315 23 26 0.885 Comp.4 0.0727 4 6 0.667 'log Lik.' 73.74085 (df=83) AIC: 18.51829 BIC: 182.0525	prior size post>0 ratio Comp.1 0.0754 4 12 0.333 Comp.2 0.0567 3 4 0.750 Comp.3 0.2541 14 18 0.778 Comp.4 0.4150 22 27 0.815 Comp.5 0.1987 10 22 0.455 'log Lik.' 98.28643 (df=104) AIC: 11.42713 BIC: 216.3375
TpAUC	prior size post>0 ratio Comp.1 0.0942 5 5 1.000 Comp.2 0.2499 13 24 0.542 Comp.3 0.3902 21 27 0.778 Comp.4 0.1726 9 13 0.692 Comp.5 0.0931 5 12 0.417 'log Lik.' 55.89586 (df=104) AIC: 96.20828 BIC: 301.1186	prior size post>0 ratio Comp.1 0.3911 21 37 0.568 Comp.2 0.0566 3 4 0.750 Comp.3 0.2773 15 20 0.750 Comp.4 0.2750 14 28 0.500 'log Lik.' 41.51126 (df=83) AIC: 82.97748 BIC: 246.5117

Tabla 3.3: Modelos para el criterio AIC.

• Para el criterio BIC:

En las Tablas 3.4 y 3.5 se observa que para el grupo ROC, los modelos obtenidos por este criterio con inicializaciones CEM y SEM son los mismos y ambos son considerados los mejores modelos para este grupo por tener menor BIC. Se escogerá el modelo que converja a la solución con un menor número de iteraciones del algoritmo, que en este caso, se corresponde con el de inicialización SEM.

Para el grupo pAUC y TpAUC los modelos que se seleccionan usando de nuevo el criterio BIC, son los obtenidos mediante una inicialización de tipo emEM. Para el grupo AUC, se selecciona el modelo con método de inicio aleatorio y para el Z-stat aquel con una inicialización de tipo SEM.

	Rnd	emEM
ROC	prior size post>0 ratio Comp.1 0.485 26 29 0.897 Comp.2 0.515 27 34 0.794 'log Lik.' -30.78427 (df=41) AIC: 143.5685 BIC: 224.3505	prior size post>0 ratio Comp.1 0.485 26 29 0.897 Comp.2 0.515 27 34 0.794 'log Lik.' -30.78428 (df=41) AIC: 143.5686 BIC: 224.3505
pAUC	prior size post>0 ratio Comp.1 0.0926 5 11 0.455 Comp.2 0.4170 22 29 0.759 Comp.3 0.4903 26 28 0.929 'log Lik.' 74.66185 (df=62) AIC: -25.32371 BIC: 96.83439	prior size post>0 ratio Comp.1 0.454 24 31 0.774 Comp.2 0.112 6 22 0.273 Comp.3 0.434 23 24 0.958 'log Lik.' 75.49043 (df=62) AIC: -26.98086 BIC: 95.17723
AUC	prior size post>0 ratio Comp.1 0.4215 22 28 0.786 Comp.2 0.4839 26 28 0.929 Comp.3 0.0946 5 11 0.455 'log Lik.' -55.52632 (df=62) AIC: 235.0526 BIC: 357.2107	prior size post>0 ratio Comp.1 0.464 25 27 0.926 Comp.2 0.134 7 20 0.350 Comp.3 0.403 21 28 0.750 'log Lik.' -56.54518 (df=62) AIC: 237.0904 BIC: 359.2485
Z-stat	prior size post>0 ratio Comp.1 0.467 25 28 0.893 Comp.2 0.533 28 35 0.800 'log Lik.' 11.43615 (df=41) AIC: 59.12769 BIC: 139.9097	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43615 (df=41) AIC: 59.12769 BIC: 139.9097
TpAUC	prior size post>0 ratio Comp.1 0.64 34 41 0.829 Comp.2 0.36 19 33 0.576 'log Lik.' -31.53945 (df=41) AIC: 145.0789 BIC: 225.8609	prior size post>0 ratio Comp.1 0.278 15 20 0.750 Comp.2 0.322 17 30 0.567 Comp.3 0.401 21 39 0.538 'log Lik.' 11.80196 (df=62) AIC: 100.3961 BIC: 222.5542

Tabla 3.4: Modelos para el criterio BIC.

	CEM	SEM
ROC	prior size post>0 ratio Comp.1 0.547 29 36 0.806 Comp.2 0.453 24 28 0.857 'log Lik.' -24.75693 (df=41) AIC: 131.5139 BIC: 212.2958	prior size post>0 ratio Comp.1 0.453 24 28 0.857 Comp.2 0.547 29 36 0.806 'log Lik.' -24.75693 (df=41) AIC: 131.5139 BIC: 212.2958
pAUC	prior size post>0 ratio Comp.1 0.471 25 27 0.926 Comp.2 0.529 28 36 0.778 'log Lik.' 28.4403 (df=41) AIC: 25.1194 BIC: 105.9014	prior size post>0 ratio Comp.1 0.471 25 27 0.926 Comp.2 0.529 28 36 0.778 'log Lik.' 28.4403 (df=41) AIC: 25.1194 BIC: 105.9014
AUC	prior size post>0 ratio Comp.1 0.134 7 20 0.350 Comp.2 0.464 25 27 0.926 Comp.3 0.403 21 28 0.750 'log Lik.' -56.54517 (df=62) AIC: 237.0903 BIC: 359.2484	prior size post>0 ratio Comp.1 0.403 21 28 0.750 Comp.2 0.464 25 27 0.926 Comp.3 0.134 7 20 0.350 'log Lik.' -56.54532 (df=62) AIC: 237.0906 BIC: 359.2487
Z-stat	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43615 (df=41) AIC: 59.12769 BIC: 139.9097	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43621 (df=41) AIC: 59.12758 BIC: 139.9095
TpAUC	prior size post>0 ratio Comp.1 0.378 20 35 0.571 Comp.2 0.622 33 39 0.846 'log Lik.' -35.01426 (df=41) AIC: 152.0285 BIC: 232.8105	prior size post>0 ratio Comp.1 0.64 34 41 0.829 Comp.2 0.36 19 33 0.576 'log Lik.' -31.53945 (df=41) AIC: 145.0789 BIC: 225.8609

Tabla 3.5: Modelos para el criterio BIC.

• Para el criterio ICL:

En las Tablas 3.7 y 3.8 se observa que con este criterio se obtienen, en muchos casos, los mismos modelos que con el criterio BIC.

Para el grupo ROC, los modelos seleccionados por el criterio ICL son los obtenidos mediante un método de inicio CEM y SEM, que de nuevo son iguales (Tablas 3.6 y 3.8). Como ocurría anteriormente, se escogerá el modelo que converja a la solución con un menor número de iteraciones del algoritmo, que en este caso también se corresponde con el de inicialización SEM.

Para el grupo pAUC y TpAUC los modelos que se seleccionan con el criterio ICL son los obtenidos mediante una inicialización de tipo emEM. Para el grupo AUC, se selecciona el modelo con método de inicio aleatorio y para el Z-stat aquel con una inicialización de tipo SEM.

	Rnd	emEM	CEM	SEM
ROC	225,2094	225,2094	212.4381	212.4381
pAUC	97,07684	95,74085	105.9301	105.9301
AUC	358,2112	360,5624	360,5618	360,5699
Z-stat	200,8069	206,3115	200,8069	200,8062
TpAUC	226,9724	223,9393	233,7345	226,9724

Tabla 3.6: Valores del ICL de los modelos.

	Rnd	emEM
ROC	prior size post>0 ratio Comp.1 0.485 26 29 0.897 Comp.2 0.515 27 34 0.794 'log Lik.' -30.78427 (df=41) AIC: 143.5685 BIC: 224.3505	prior size post>0 ratio Comp.1 0.485 26 29 0.897 Comp.2 0.515 27 34 0.794 'log Lik.' -30.78428 (df=41) AIC: 143.5686 BIC: 224.3505
pAUC	prior size post>0 ratio Comp.1 0.0926 5 11 0.455 Comp.2 0.4170 22 29 0.759 Comp.3 0.4903 26 28 0.929 'log Lik.' 74.66185 (df=62) AIC: -25.32371 BIC: 96.83439	prior size post>0 ratio Comp.1 0.454 24 31 0.774 Comp.2 0.112 6 22 0.273 Comp.3 0.434 23 24 0.958 'log Lik.' 75.49043 (df=62) AIC: -26.98086 BIC: 95.17723
AUC	prior size post>0 ratio Comp.1 0.4215 22 28 0.786 Comp.2 0.4839 26 28 0.929 Comp.3 0.0946 5 11 0.455 'log Lik.' -55.52632 (df=62) AIC: 235.0526 BIC: 357.2107	prior size post>0 ratio Comp.1 0.464 25 27 0.926 Comp.2 0.134 7 20 0.350 Comp.3 0.403 21 28 0.750 'log Lik.' -56.54518 (df=62) AIC: 237.0904 BIC: 359.2485
Z-stat	prior size post>0 ratio Comp.1 0.467 25 28 0.893 Comp.2 0.533 28 35 0.800 'log Lik.' 11.43615 (df=41) AIC: 59.12769 BIC: 139.9097	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43615 (df=41) AIC: 59.12769 BIC: 139.9097
TpAUC	prior size post>0 ratio Comp.1 0.64 34 41 0.829 Comp.2 0.36 19 33 0.576 'log Lik.' -31.53945 (df=41) AIC: 145.0789 BIC: 225.8609	prior size post>0 ratio Comp.1 0.278 15 20 0.750 Comp.2 0.322 17 30 0.567 Comp.3 0.401 21 39 0.538 'log Lik.' 11.80196 (df=62) AIC: 100.3961 BIC: 222.5542

Tabla 3.7: Modelos para el criterio ICL.

	CEM	SEM
ROC	prior size post>0 ratio Comp.1 0.547 29 36 0.806 Comp.2 0.453 24 28 0.857 'log Lik.' -24.75693 (df=41) AIC: 131.5139 BIC: 212.2958	prior size post>0 ratio Comp.1 0.453 24 28 0.857 Comp.2 0.547 29 36 0.806 'log Lik.' -24.75693 (df=41) AIC: 131.5139 BIC: 212.2958
pAUC	prior size post>0 ratio Comp.1 0.3939 21 23 0.913 Comp.2 0.5120 27 40 0.675 Comp.3 0.0941 5 11 0.455 'log Lik.' 72.09476 (df=62) AIC: -20.18952 BIC: 101.9686	prior size post>0 ratio Comp.1 0.446 24 30 0.800 Comp.2 0.434 23 24 0.958 Comp.3 0.120 6 24 0.250 'log Lik.' 75.57968 (df=62) AIC: -27.15935 BIC: 94.99874
AUC	prior size post>0 ratio Comp.1 0.134 7 20 0.350 Comp.2 0.464 25 27 0.926 Comp.3 0.403 21 28 0.750 'log Lik.' -56.54517 (df=62) AIC: 237.0903 BIC: 359.2484	prior size post>0 ratio Comp.1 0.403 21 28 0.750 Comp.2 0.464 25 27 0.926 Comp.3 0.134 7 20 0.350 'log Lik.' -56.54532 (df=62) AIC: 237.0906 BIC: 359.2487
Z-stat	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43615 (df=41) AIC: 59.12769 BIC: 139.9097	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43621 (df=41) AIC: 59.12758 BIC: 139.9095
TpAUC	prior size post>0 ratio Comp.1 0.378 20 35 0.571 Comp.2 0.622 33 39 0.846 'log Lik.' -35.01426 (df=41) AIC: 152.0285 BIC: 232.8105	prior size post>0 ratio Comp.1 0.64 34 41 0.829 Comp.2 0.36 19 33 0.576 'log Lik.' -31.53945 (df=41) AIC: 145.0789 BIC: 225.8609

Tabla 3.8: Modelos para el criterio ICL.

En general se seleccionan los siguientes modelos para cada grupo de genes y cada criterio

	AIC	BIC	ICL
ROC	prior size post>0 ratio Comp.1 0.3401 18 24 0.750 Comp.2 0.0941 5 14 0.357 Comp.3 0.4343 23 28 0.821 Comp.4 0.1314 7 11 0.636 'log Lik.' 49.78276 (df=83) AIC: 66.43449 BIC: 229.9687	prior size post>0 ratio Comp.1 0.453 24 28 0.857 Comp.2 0.547 29 36 0.806 'log Lik.' -24.75693 (df=41) AIC: 131.5139 BIC: 212.2958	prior size post>0 ratio Comp.1 0.453 24 28 0.857 Comp.2 0.547 29 36 0.806 'log Lik.' -24.75693 (df=41) AIC: 131.5139 BIC: 212.2958 ICL=212.4381
pAUC	prior size post>0 ratio Comp.1 0.0566 3 3 1.000 Comp.2 0.3501 18 31 0.581 Comp.3 0.4699 25 26 0.962 Comp.4 0.1234 7 12 0.583 'log Lik.' 104.116 (df=83) AIC: -42.23196 BIC: 121.3023	prior size post>0 ratio Comp.1 0.454 24 31 0.774 Comp.2 0.112 6 22 0.273 Comp.3 0.434 23 24 0.958 'log Lik.' 75.49043 (df=62) AIC: -26.98086 BIC: 95.17723	prior size post>0 ratio Comp.1 0.454 24 31 0.774 Comp.2 0.112 6 22 0.273 Comp.3 0.434 23 24 0.958 'log Lik.' 75.49043 (df=62) AIC: -26.98086 BIC: 95.17723 ICL=95,74085
AUC	prior size post>0 ratio Comp.1 0.0755 4 9 0.444 Comp.2 0.1555 8 25 0.320 Comp.3 0.2978 16 19 0.842 Comp.4 0.1496 8 13 0.615 Comp.5 0.3216 17 25 0.680 'log Lik.' 7.84076 (df=104) AIC: 192.3185 BIC: 397.2288	prior size post>0 ratio Comp.1 0.4215 22 28 0.786 Comp.2 0.4839 26 28 0.929 Comp.3 0.0946 5 11 0.455 'log Lik.' -55.52632 (df=62) AIC: 235.0526 BIC: 357.2107	prior size post>0 ratio Comp.1 0.4215 22 28 0.786 Comp.2 0.4839 26 28 0.929 Comp.3 0.0946 5 11 0.455 'log Lik.' -55.52632 (df=62) AIC: 235.0526 BIC: 357.2107 ICL=358,2112
Z-stat	prior size post>0 ratio Comp.1 0.0754 4 12 0.333 Comp.2 0.0567 3 4 0.750 Comp.3 0.2541 14 18 0.778 Comp.4 0.4150 22 27 0.815 Comp.5 0.1987 10 22 0.455 'log Lik.' 98.28643 (df=104) AIC: 11.42713 BIC: 216.3375	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43621 (df=41) AIC: 59.12758 BIC: 139.9095	prior size post>0 ratio Comp.1 0.533 28 35 0.800 Comp.2 0.467 25 28 0.893 'log Lik.' 11.43621 (df=41) AIC: 59.12758 BIC: 139.9095 ICL=200,8062
TpAUC	prior size post>0 ratio Comp.1 0.3911 21 37 0.568 Comp.2 0.0566 3 4 0.750 Comp.3 0.2773 15 20 0.750 Comp.4 0.2750 14 28 0.500 'log Lik.' 41.51126 (df=83) AIC: 82.97748 BIC: 246.5117	prior size post>0 ratio Comp.1 0.278 15 20 0.750 Comp.2 0.322 17 30 0.567 Comp.3 0.401 21 39 0.538 'log Lik.' 11.80196 (df=62) AIC: 100.3961 BIC: 222.5542	prior size post>0 ratio Comp.1 0.278 15 20 0.750 Comp.2 0.322 17 30 0.567 Comp.3 0.401 21 39 0.538 'log Lik.' 11.80196 (df=62) AIC: 100.3961 BIC: 222.5542 ICL=223,9393

Tabla 3.9: Selección de un modelo para cada grupo y criterio.

A partir de la Tabla 3.9, se puede observar que:

► Para el grupo de genes AUC y cualquiera de los tres criterios, se selecciona el modelo obtenido mediante una inicialización de tipo aleatorio.

► Se seleccionan más modelos con una inicialización de tipo SEM, concretamente siete modelos son seleccionados con este método de inicio: 3 con el criterio AIC, 2 con el BIC y 2 con el ICL.

► Para todos los grupos los modelos seleccionados por el criterio ICL y BIC son los mismos.

► Se constata, como veíamos en Capítulo 2, que el criterio AIC sobreestima el orden del modelo, es decir, proporciona modelos con bastantes más componentes que el modelo teórico de los datos. Por tanto, en adelante trabajaremos con los seleccionados por el criterio BIC o ICL.

► Se tiene que los modelos que mejor se ajustan a los datos según el valor de cualquiera de los criterios de selección son el pAUC, posteriormente el Z-stat y el TpAUC.

Observación 3.1. Se han estudiado también los modelos obtenidos usando distribuciones normales en lugar de gamma. Se obtienen modelos que en general se ajustan peor a los datos reales de los que disponemos (ver Apéndice B), ya que los valores del AIC, BIC e ICL son mayores que para los modelos de mixturas gamma. Por tanto, en adelante trabajaremos con los modelos de mixturas gamma en lugar de normales.

3.2.1. Parámetros estimados de la mixtura

En este apartado veremos como conseguir la estimación de los parámetros de los modelos de mixturas obtenidos. Los pesos de la muestra real se corresponden con las proporciones de tejidos enfermos y sanos, es decir, el peso del clúster que comprende los tejidos sanos es de $\frac{30}{53} \approx 0{,}566$ y el de la componente que designa los tejidos enfermos de $\frac{23}{53} \approx 0{,}434$. Los pesos de cada una de las componentes de los modelos obtenidos se pueden obtener con la función *prior* del paquete FlexMix. La siguiente tabla resume los pesos de cada uno de los modelos:

	ROC	pAUC	AUC	Z-stat	TpAUC
π_1	0,547	0,454	0,4215	0,533	0,278
π_2	0,453	0,112	0,4839	0,467	0,322
π_3		0,434	0,0946		0,401

Tabla 3.10: Tabla de pesos de las componentes de los modelos.

Se observa en la Tabla 3.10, que el modelo cuyos pesos más se aproximan a los pesos reales de la mixtura es el pAUC, considerando que la componente 3 designa los tejidos cancerosos y la 1 y 2 los sanos. También se podría considerar que la componente 1 designa los tejidos enfermos y la 2 y 3 los sanos, resultando en este caso un modelo de clasificación con pesos aproximados menos ajustados a los reales.

Mediante la función *parameters* se obtiene la estimación de los parámetros de la distribución gamma para cada gen. Por ejemplo, para el modelo ROC y el primer gen de este grupo, el gen 93, se tienen los siguientes parámetros para cada una de las dos componentes que forman el modelo de mixturas:

```
                  Comp.1    Comp.2
coef.(Intercept) 3.374894 0.6197586
shape            3.146177 2.3465280
```

3.2.2. Solapamiento entre componentes

En las tablas anteriores, se ha usado el comando *summary* para cada uno de los casos estudiados. Obtenemos así el valor *prior* que se corresponde con el peso estimado $\hat{\pi}_i$ de cada una de las componentes en la mixtura, *size* que informa del tamaño de cada uno de los clusters, *post>0* que se corresponde con el número de componentes cuya probabilidad a posteriori satisfacen $\hat{\tau}_{it} > 10^{-4}$ y por último, *ratio* que informa sobre si existe solapamiento entre los clusters generados. Para componentes o clusters bien separadas (sin solapamiento), se debe obtener un número alto de observaciones con probabilidades a posterior $\hat{\tau}_{it}$ mayores que 10^{-4} y una proporción (*ratio*) próxima a 1.

La separación o por el contrario, el solapamiento entre las componentes del modelo de mixturas obtenido también se puede estudiar realizando un rotograma, en el que se representen las probabilidades a posteriori de cada observación según la componente de la que provengan. La diferencia entre un rotograma y un histograma es que el primero escala las barras de acuerdo a la raíz cuadrada del número de observaciones. Habitualmente, en cada componente de los modelos de mixturas, para muchas observaciones se obtienen probabilidades a posteriori muy próximas a cero, lo que daría lugar a una barra muy alta en el rotograma para el valor cero. Para facilitar la visualizacién de los datos, todas las probabilidades a posteriori menores que un cierto valor, que por defecto es 10^{-4}, son ignoradas en la representación del rotograma. Un pico cerca de la probabilidad 1 indicaría que la componente de la mixtura está bien separada de las otras componentes, mientras que si esto no ocurre o se observa una masa importante de valores en la parte central del intervalo, se concluiría que existe solapamiento con otras componentes.

A continuación veremos algún ejemplo del estudio del solapamiento entre componentes teniendo en cuenta la representación del rotograma y los valores obtenidos para *post>0* y *ratio*.

Para el modelo de mixturas obtenido para el índice pAUC, se tiene el siguiente rotograma:

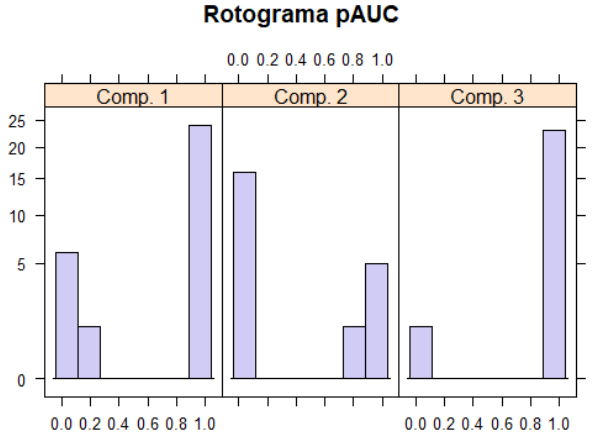

Figura 3.2: Rotograma grupo pAUC.

En la Figura 3.2 y teniendo en cuenta los valores obtenidos en la Tabla 3.9, se constata que la componente 1 y 3 (*ratios* de 0,774 y 0,958 respectivamente) se encuentran relativamente bien separadas, mientras que para la componente 2 no se observan picos cerca del valor 1 y el *ratio* es de 0,273, por lo que está bastante solapada con las otras dos.

Sin embargo, para el grupo ROC se obtiene el siguiente rotograma:

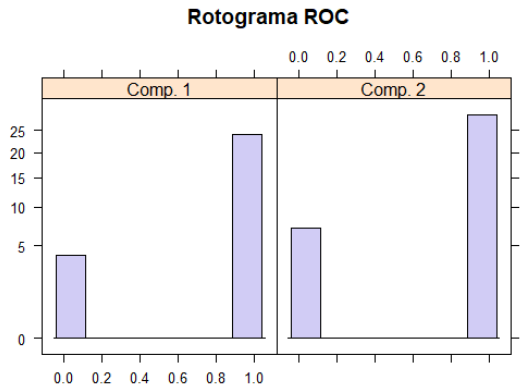

Figura 3.3: Rotograma grupo ROC.

donde se observa, junto con los valores de los *ratios* de cada componente (0,806 y 0,857 respectivamente), que no existe apenas solapamiento entre las dos componentes o clusters.

Para los otros tres grupos de genes se obtienen los siguientes rotogramas:

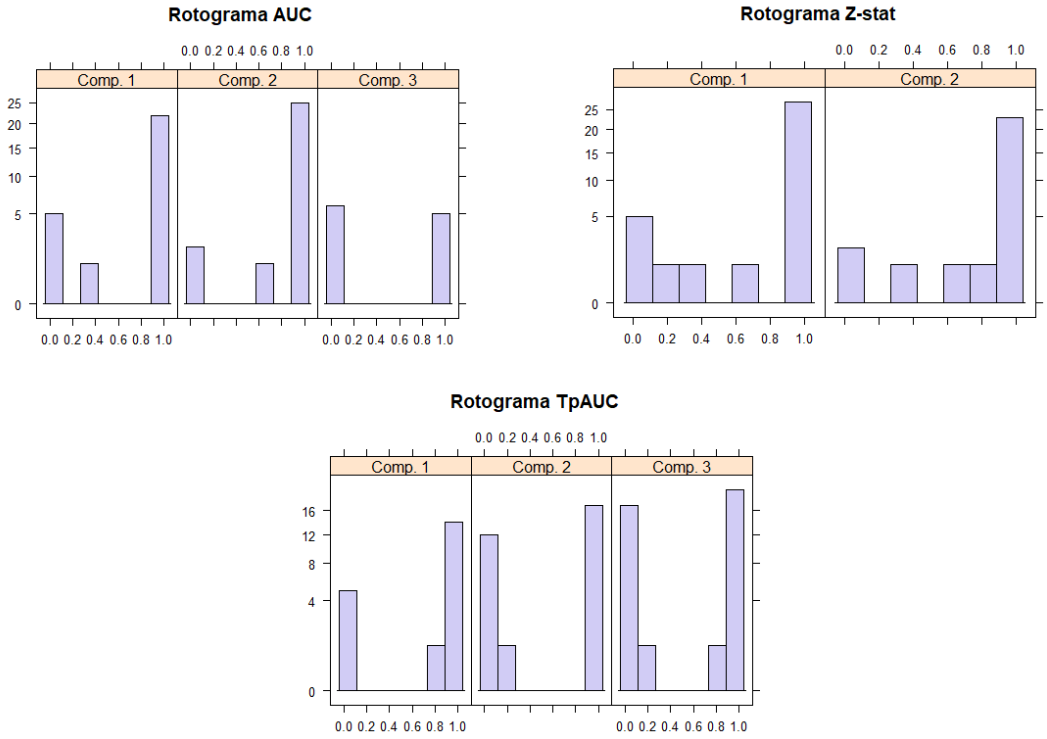

Figura 3.4: Rotogramas grupo AUC, Z-stat y TpAUC.

que indican que para el modelo de genes AUC las componentes 1 y 2 están bien separadas pero la 3 no; para el Z-stat ambas componentes están relativamente bien separadas y para el TpAUC la componente 1 está bien separada y existe cierto solapamiento entre las otras dos.

3.2.3. Clasificación y ajuste de los modelos obtenidos

En este apartado estudiaremos la proporción de verdaderos negativos o especificidad y de verdaderos positivos o sensibilidad de los modelos obtenidos en el apartado anterior. También compararemos la clasificación de la muestra realizada por estos modelos comparándola con la verdadera clasificación, que venía determinada en los datos originales por la columna "d", con valor 1 si el tejido es canceroso y 0 si no tiene cáncer.

En el caso de los genes seleccionados por el índice pAUC, AUC y TpAUC hemos obtenido modelos de mixturas de tres componentes. Vamos a estudiar cuáles se corresponden con los tejidos sanos y cuáles con los enfermos. En este tipo de datos es razonable que exista mayor variabilidad de la clasificación de los tejidos sin cáncer ovárico dado que, a pesar de referirnos a ellos como sanos, lo único que podemos garantizar es que no padecen este tipo concreto de cáncer pero no que sean muestras sanas realmente.

Con las siguientes tablas se representan las clasificaciones realizadas por cada modelo frente a la clasificación real de los datos.

	Comp. 1	Comp. 2	Comp. 3
Sanos: 0	1	3	19
Enfermos: 1	23	3	4

Tabla 3.11: Tejidos sanos y enfermos para grupo pAUC.

	Comp. 1	Comp. 2	Comp. 3
Sanos: 0	1	21	1
Enfermos: 1	21	5	4

Tabla 3.12: Tejidos sanos y enfermos para grupo AUC.

	Comp. 1	Comp. 2	Comp. 3
Sanos: 0	10	0	13
Enfermos: 1	5	17	8

Tabla 3.13: Tejidos sanos y enfermos para grupo TpAUC.

Observando las clasificaciones obtenidas en la Tabla 3.11, para el grupo pAUC la componente 3 se corresponde con los tejidos sanos, la 1 con los enfermos y la 2 clasifica el mismo número de tejidos enfermos que sanos. Se considerará que se corresponde con los tejidos sanos, por poder existir mayor variabilidad en este grupo ya que el sujeto puede padecer

otra patología ajena al cáncer ovárico que altere su expresión genética. Además, ya se ha comentado que se buscan valores bajos del FPR, es decir, valores altos de la especificidad, por lo que es preferible obtener más falsos negativos que falsos positivos.

Según la Tabla 3.12, para el grupo AUC la componente 2 es la que designa los tejidos sanos y la 1 y 3 los enfermos. Para el TpAUC (Tabla 3.13) la componente 2 se corresponde con los tejidos enfermos y la 1 y la 3 los sanos.

Los otros dos modelos (los obtenidos por los grupos ROC y Z-stat) tienen dos componentes: una se corresponderá con los tejidos sanos y la otra con los enfermos.

Observando la clasificación efectuada para cada modelo, se obtiene:

	Comp. 1	Comp. 2
Sanos: 0	20	3
Enfermos: 1	4	26

Tabla 3.14: Tejidos sanos y enfermos para grupo ROC.

	Comp. 1	Comp. 2
Sanos: 0	2	21
Enfermos: 1	26	4

Tabla 3.15: Tejidos sanos y enfermos para grupo Z-stat.

Podemos concluir que para el grupo ROC la componente 2 se corresponde con los tejidos enfermos y la 1 con los sanos. Para el grupo Zstat la componente 1 se corresponde con los tejidos enfermos y la 2 con los sanos.

Se realizan los cambios pertinentes en cada caso para mantener la notación original del estado de cada individuo 0 para el sano y 1 para el enfermo, como se muestra en el Apéndice A.

De cada uno de las clasificaciones calcularemos la proporción de verdaderos positivos (sensibilidad) y verdaderos negativos (especificidad) del modelo y el valor predictivo de cada uno, cuyos resultados se muestran en la siguiente tabla.

	Sensibilidad	Especificidad	GR
ROC	0,8695652	0,8666667	0,8679245
pAUC	0,9565217	0,7666667	0,8490566
AUC	0,9130435	0,8333333	0,8679245
Zstat	0,9130435	0,8666667	0,8867925
TpAUC	1,0000000	0,5666667	0,7547170

Tabla 3.16: Sensibilidad, especificidad y valor predictivo global de cada modelo.

En la Tabla 3.16 se observa que los mayores valores de sensibilidad se obtienen para los modelos seleccionados con los grupos de genes TpAUC y posteriormente pAUC, AUC y Z-stat, por lo que estos tienen éxito a la hora de clasificar como cancerosos los tejidos

que lo son. Cabe destacar que el modelo TpAUC tiene una sensibilidad de 1 por lo que no falla nunca al clasificar un tejido enfermo como tal.

Los mayores valores de especificidad se obtienen para los modelos ROC, Z-stat, AUC y pAUC por lo que estos modelos clasifican bien los tejidos sanos. Sin embargo, el modelo TpAUC falla clasificando los tejidos sanos en casi un 44 % de los casos.

Por tanto, el modelo que mejor clasifica la muestra es el obtenido mediante el grupo Z-stat.

A continuación se representan las curvas ROC de las clasificaciones proporcionadas por cada uno de los modelos haciendo uso del paquete *pROC*.

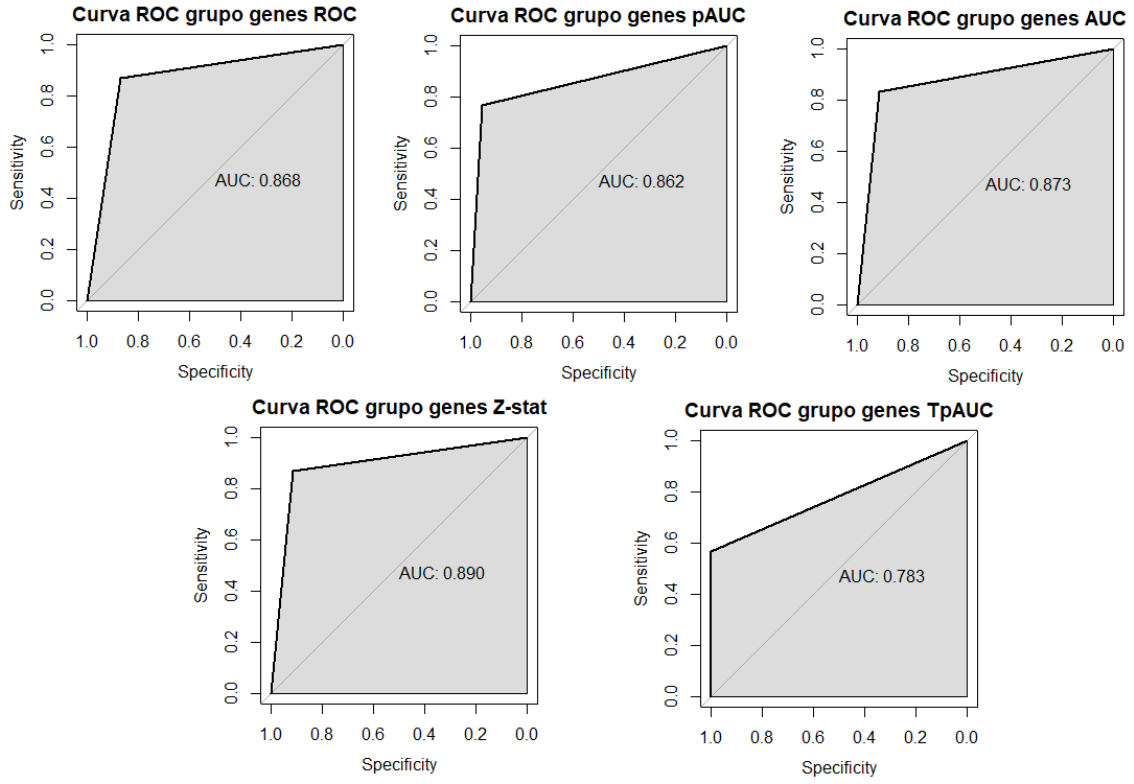

Figura 3.5: Curva ROC de cada modelo de clasificación.

A partir de las gráficas de la Figura 3.5, se constata que el área bajo la curva es mayor para el grupo Z-stat, por lo que se trata del mejor método clasificador de la muestra.

3.2.4. Estabilidad de los modelos: técnica Bootstrap

Aplicamos la técnica de remuestro bootstrap sobre los modelos de mixturas obtenidos para cada grupo de genes. El objetivo es estudiar la estabilidad de los modelos, es decir, estudiaremos, en qué medida estos modelos son capaces de clasificar correctamente una muestra aleatoria generada a partir de ellos. Se usará el comando *rflexmix* implementado en el paquete, que genera números aleatorios a partir del modelo de mixturas dado. Se

puede acceder a la clasificación que proporciona usando el comando $\$class$ (Grün and Leisch, 2010) y obtenemos las siguientes tablas de clasificación de los datos

	Comp. 1	Comp. 2
Sanos: 0	14	9
Enfermos: 1	16	14

Tabla 3.17: Tejidos sanos y enfermos para el remuestreo del grupo ROC.

	Comp. 1	Comp. 2	Comp. 3
Sanos: 0	8	2	13
Enfermos: 1	14	3	13

Tabla 3.18: Tejidos sanos y enfermos para el remuestreo del grupo pAUC.

	Comp. 1	Comp. 2	Comp. 3
Sanos: 0	13	9	1
Enfermos: 1	15	14	1

Tabla 3.19: Tejidos sanos y enfermos para el remuestreo del grupo AUC.

	Comp. 1	Comp. 2
Sanos: 0	12	11
Enfermos: 1	17	13

Tabla 3.20: Tejidos sanos y enfermos para el remuestreo del grupo Z-stat.

	Comp. 1	Comp. 2	Comp. 3
Sanos: 0	5	8	10
Enfermos: 1	9	12	9

Tabla 3.21: Tejidos sanos y enfermos para el remuestreo del grupo TpAUC.

A partir de las Tablas 3.17-3.21, se observa que existe un mayor solapamiento de las componentes y ya no se distingue de forma sencilla qué componentes designan los tejidos enfermos y cuáles los sanos.

Suponiendo las mismas clasificaciones previas, es decir, que para el grupo ROC la componente 1 identifica los tejidos sanos y la 2 los enfermos, para el grupo pAUC la componente 2 y 3 se corresponden con los tejidos sanos, y así sucesivamente, se obtienen los siguientes valores de sensibilidad, especificidad y valor predictivo global para cada modelo:

	Sensibilidad	Especificidad	GR
ROC	0,6086957	0,4666667	0,5283019
pAUC	0,6521739	0,4666667	0,5471698
AUC	0,08695652	0,90000000	0,5471698
Z-stat	0,4782609	0,5666667	0,5283019
TpAUC	0,6521739	0,4000000	0,5094340

Tabla 3.22: Sensibilidad, especificidad y valor predictivo global para el remuestreo de los modelos.

Los mayores valores de sensibilidad se obtienen con los modelos pAUC y AUC. El valor más alto de especificidad se obtiene para el modelo AUC, pero también el más bajo de sensibilidad, por lo que el mejor modelo de clasificación es el pAUC.

Los valores predictivos globales de todos los modelos son bastante bajos por lo que se considera que ninguno de ellos es suficientemente estable para clasificar correctamente una muestra generada de forma aleatoria.

Capítulo 4

Discusión y conclusiones

Uno de los principales problemas del uso del algoritmo EM para estimar los parámetros de un modelo de mixturas es la necesidad de introducir un número fijo de componentes, en general desconocido. Otro de los obstáculos que presenta es que la convergencia a la solución óptima depende de los valores de inicio utilizados en el algoritmo. Ambos son problemas característicos del algoritmo EM para los que todavía no existen estrategias que permitan abordarlos de forma definitiva. Para el primero, se propone considerar modelos con distinto número de componentes y usar criterios de selección para determinar el modelo que mejor se ajusta a los datos. Para el segundo problema, se proponen distintos métodos de inicialización y algunas variantes del algoritmo. Sin embargo, no existe consenso en qué criterio de selección se debe usar ni cuál es el mejor método de inicio. Los resultados varían según el tamaño de la muestra, el número de componentes de la mixtura, la familia de distribución paramétrica y el tipo de datos.

En este trabajo además de haberse estudiado estas dos problemáticas aplicadas a los modelos de mixturas gamma y normales, se ha planteado como principal objetivo obtener un modelo que clasifique bien cualquier muestra de tejido ovárico sin necesidad de saber si se trata de un tejido canceroso o no, y así poder estudiar qué grupo de genes resultan ser mejores biomarcadores de esta enfermedad. A continuación se comentan brevemente los resultados obtenidos.

► Con respecto a los distintos criterios usados para seleccionar el mejor modelo, es decir, determinar el número óptimo de componentes de la mixtura en cada caso, se constata que el criterio Akaike sobreestima el número de componentes para todos los grupos de genes y las dos familias parámetricas consideradas, como sucede en Baudry et al. (2015), Hu (2015) y Olivier (1999), ya que se obtienen modelos con un número de componentes muy superior a la verdadera clasificación. Por otro lado, en nuestro estudio, los modelos seleccionados por los criterios BIC e ICL coinciden en todos los casos. Se comentaba en el segundo capítulo que el criterio ICL es más robusto para modelos con componentes no gausianas, motivo por el que se decidió incluir en el estudio. Sin embargo, para nuestra base de datos, ambos criterios son igualmente eficaces en la selección del número de componentes de la mixtura.

Los valores de los tres criterios son superiores para los modelos GMM que los obtenidos para los modelos de mixturas gamma, por lo que se considera que los segundos ajustan

mejor la muestra.

Además, los menores valores de los tres criterios de selección se obtienen para el grupo pAUC, por lo que, con este criterio, se puede considerar el modelo que mejor se ajusta a los datos. Posteriormente se encuentran el Z-stat, el ROC, el TpAUC y por último el AUC.

► En lo referente a métodos de inicio, se han llevado a la práctica las inicializaciónes de tipo aleatorio, ejecuciones cortas del algoritmo y dos de las variantes del algoritmo EM: el CEM y el SEM. Se han escogidos estas cuatro estrategias de inicialización por ser ampliamente usadas, producir buenos resultados en lo referente al coste computacional y eficacia de la clasificación realizada, que además están implementadas en el paquete estadístico.

A continuación se detallan los resultados y conclusiones obtenidos sobre los distintos métodos de inicio.

Para los modelos de mixturas gamma se tiene que, de los 15 modelos seleccionados (uno por cada grupo de genes y cada criterio de selección usado), 3 han sido obtenidos mediante una inicialización de tipo aleatorio, 4 de tipo emEM, sólo uno de tipo CEM y 7 de tipo SEM.

Para los GMM, sólo uno de ellos ha sido obtenido mediante una inicialización de tipo aleatorio, 2 de tipo emEM, 9 de tipo CEM y 3 de tipo SEM.

Los modelos obtenidos mediante los criterios BIC e ICL coinciden y se tiene que entre los modelos de mixturas gamma seleccionados uno ha sido obtenido mediante una inicialización aleatoria, 2 de tipo emEM y 2 de tipo SEM. Para los GMM se obtienen 4 modelos mediante una inicialización CEM y sólo uno con inicialización de tipo SEM.

Por tanto, se observa que, en general, los mejores modelos se obtienen, para el caso de los modelos gamma, con las estrategias de inicialización SEM y emEM, como se proponía en Scharl et al. (2009) y Baudry and Celeux (2015); y para los GMM, con el método CEM.

► En relación a la superposición de los clusters de los modelos gamma se han obtenido dos modelos (grupo ROC y Z-stat) con dos componentes bien separadas, por lo que una se corresponde con los tejidos cancerosos y la otra con los sanos; y tres modelos (grupo pAUC, AUC y TpAUC) de tres componentes con cierto solapamiento entre sus componentes, por lo que resulta más complicado realizar la correspondencia de cada componente con la clasificación enfermo-sano. Resulta lógico que, en estos tres últimos casos, alguna de las componentes estén solapadas dado que el modelo teórico de clasificación de los datos está formado por dos componentes, una por cada tipo de tejido.

► Se ha realizado el estudio haciendo corresponder cada componente con el tipo de tejido predominante que la constituye. Por ejemplo, en el caso del grupo AUC, la primera componente está compuesta por una muestra sana y 21 enfermas, por lo que se considera que la componente 1 designa los tejidos enfermos. La componente 2 está formada por 21 muestras sanas y 5 enfermas, entonces se corresponde con los tejidos sanos; y la tercera componente por 4 tejidos enfermos y sólo uno sano, por lo que también se ha hecho corresponder con el grupo de tejidos enfermos.

Para el grupo pAUC, se ha obtenido que una de las componentes estaba formada por

tantos tejidos enfermos como sanos. En este caso, se ha tenido en cuenta que se trata de una prueba que pretende detectar los tejidos enfermos de un cáncer muy concreto, pero que el sujeto puede padecer otras patologías distintas que influyan en la expresión genética que se está considerando, por lo que puede existir mayor variabilidad para los tejidos correspondientes al grupo de los sanos. Además, se trata de una enfermedad con un tratamiento agresivo para el sujeto, por lo que en Pepe et al. (2003) se sugiere considerar valores altos de la especificidad (o equivalentemente bajos de FPR) a pesar de poder obtener más falsos negativos. Por todo esto, se ha considerado que dicha componente se corresponde con los tejidos sanos en lugar de con los enfermos.

► En relación al estudio de la eficacia de los modelos de clasificación de los datos se ha estudiado sobre cada uno el valor predictivo global y el área bajo la curva. La mejor clasificación se obtiene con el modelo del grupo de genes Z-stat. Posteriormente y en orden decreciente, el AUC, el ROC, el pAUC y por último el TpAUC.

En este caso como estamos interesados en una prueba con valores altos de especificidad, la mejor prueba-diagnóstico del cáncer ovárico sería el modelo de mixturas obtenido con el grupo de genes Z-stat, el ROC o el AUC.

► Tras estudiar la estabilidad de los modelos de clasificación realizando un remuestreo aleatorio, se obtiene que el mejor modelo de clasificación es el grupo pAUC y el AUC.

Ordenando las pruebas-diagnóstico según la especificidad, se obtiene que el mejor modelo es el AUC, y posteriormente el pAUC, Z-stat, ROC y TpAUC. Sin embargo cabe destacar que el modelo AUC proporciona el valor más bajo de sensibilidad respecto a los otros cuatro grupos de genes. Por ello, se considera que el pAUC es el grupo de genes que proporciona una mejor clasificación de la muestra aleatoria, siendo así el modelo de clustering más estable.

En conclusión, en este trabajo se han estudiado varios modelos de clasificación de una muestra de tejidos ováricos con y sin cáncer con el objetivo de determinar cuáles son los mejores biomarcadores genéticos de la enfermedad y así contribuir a la detección temprana del cáncer ovárico.

Para ello, se ha partido de 4 grupos de diez genes considerados buenos biomarcadores según los criterios ROC, pAUC, AUC y Z-stat, y que habían sido propuestos en Pepe et al. (2003). En este trabajo hemos incluido en el estudio los diez genes seleccionados por el índice TpAUC, propuesto en Vivo et al. (2017).

Finalmente, se ha obtenido que la mejor clasificación para esta muestra se produce con el grupo de genes seleccionado por el índice Z-stat con un porcentaje de éxito de casi el 89 %, no habiendo excesiva diferencia con la clasificación obtenida por los criterios ROC, AUC y pAUC. El grupo de genes TpAUC ha resultado ser algo menos eficaz que los otros cuatro, con un portentaje de éxito de aproximadamente el 75 %.

Por otro lado, el modelo más estable, es decir, aquel que produce una mejor clasificación de una muestra aleatoria generada a partir de la base de datos original, ha resultado ser el pAUC con una eficacia del 55 %. Cabe destacar que ninguno de los modelos parece proporcionar una buena clasificación de una muestra aleatoria dado que se producen valores

predictivos globales muy bajos.

Bibliografía

Aitkin, M. and Rubin, D. B. (1985). Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):67–75.

Atienza, N. (2003). *Mixturas de distribuciones: Modelización de experiencias con asimetría en los datos*. PhD thesis, Universidad de Sevilla.

Baudry, J.-P. and Celeux, G. (2015). EM for mixtures. Statistics and Computing. 25(1):713–726.

Baudry, J.-P. et al. (2015). Estimation and model selection for model-based clustering with the conditional classification likelihood. *Electronic Journal of Statistics*, 9:4–6.

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3-4):561–575.

Celeux, G., Chauveau, D., and Diebolt, J. (1995). On stochastic versions of the EM algorithm. Technical report, HAL INRIA. https://hal.inria.fr/inria-00074164/document.

Celeux, G. and Diebolt, J. (1989). Une version de type recuit simulé de l'algorithme EM. Technical report, HAL INRIA. https://hal.inria.fr/inria-00075436/document.

Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis*, 14:315–332.

Couvreur, C. (1997). The EM algorithm: a guided tour. In *Computer Intensive Methods in Control and Signal Processing*, pages 209–222. Springer.

Delmar, P., Robin, S., Roux, T.-L., Daudin, J. J., et al. (2005). Mixture model on the variance for the differential analysis of gene expression data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54:31–50.

Ferreira, E. and Garín, M. (2010). Estadística Actuarial. Modelos Estocásticos. Universidad del País Vasco.

Freien Universität Berlin (2010). Mixture Models for the Analysis of Gene Expression, Chapter 2. https://refubium.fu-berlin.de/bitstream/handle/fub188/64/02_chapter2.pdf?sequence=3&isAllowed=y.

Gómez, Á. (2014). Modelos de mixturas finitas para la caracterización y mejora de las redes de monitorización de la calidad del aire. Master's thesis, Universidad de Granada.

Grün, B. and Leisch, F. (2007). Flexmix: An R package for finite mixture modelling. *R News*, 7(1):8–13.

Grün, B. and Leisch, F. (2010). Finite mixture model diagnostics using resampling methods. Technical report. https://cran.r-project.org/web/packages/flexmix/vignettes/bootstrapping.pdf.

Gupta, M. R., Chen, Y., et al. (2011). Theory and use of the EM algorithm. *Foundations and Trends® in Signal Processing*, 4:223–296.

Hu, Z. (2015). *Initializing the EM Algorithm for Data Clustering and Sub-population Detection*. PhD thesis, The Ohio State University, USA.

Leisch, F. (2004). Flexmix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, 11:1–18.

López, M., Mallorquín, P., and Vega, M. (2006). Aplicaciones de los microarrays y biochips en salud humana. Technical report, Fundación Española para el Desarrollo de la Investigación en Genómica y Proteómica, Universidad Autónoma de Madrid.

Marín, J. M. (2018). Análisis de dos muestras. http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/Disenno/tema1de.pdf.

McNicholas, P. D. (2016). Model-based clustering. *Journal of Classification*, 33(3):331–373.

Michael, S. and Melnykov, V. (2016). An effective strategy for initializing the EM algorithm in finite mixture models. *Advances in Data Analysis and Classification*, 10:563–583.

Ng, S. K., Krishnan, T., and McLachlan, G. J. (2012). The EM algorithm. In *Handbook of computational statistics*. pp. 139-172. Springer.

Olivier, C. (1999). Choice of the number of component clusters in mixture models by information criteria. *Proceedings of Vision Interface*. pp. 74-81.

Pepe, M. S., Longton, G., Anderson, G. L., and Schummer, M. (2003). Selecting differentially expressed genes from microarray experiments. *Biometrics*, 59:133–142.

Picard, F. (2007). An introduction to mixture models. Technical report, Statistics for Systems Biology Group.

Scharl, T., Grün, B., and Leisch, F. (2009). Mixtures of regression models for time course gene expression data: evaluation of initialization and random effects. *Bioinformatics*, 26:370–377.

Silva, C. and Molina, M. (2017). Likelihood ratio (razón de verosimilitud): definición y aplicación en radiología. *Revista Argentina de Radiología*, 81:204–208.

STHDA (2017). Model based clustering essentials. http://www.sthda.com/english/articles/30-advanced-clustering/104-model-based-clustering-essentials/.

Valle-Benavides, A. R. (2017). Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones. Master's thesis, Universidad de Sevilla.

Vijverberg, A. (2007). Clustering microarray data. Master's thesis, Pomona College, USA.

Vivo, J.-M., Franco, M., and Vicari, D. (2017). Rethinking an ROC partial area index for evaluating the classification performance at a high specificity range. *Advances in Data Analysis and Classification*. https://doi.org/10.1007/s11634-017-0295-9.

Apéndice A

Código R

En este anexo se encuentra el código R que se ha usado en el tercer capítulo.

Descripción de los datos y obtención de los modelos correspondientes

Los datos que tenemos se recogen en el documento *genes.txt*.

```
datos_orig<-read.csv("genes.txt", header=T, sep=",", dec=".")
```

Dado que hay algunas entradas de los datos que no han sido recogidas, substituímos los valores NaN por ceros.

```
haz.cero<-function(x){
              ifelse(is.na(x),0,x)
              }

datos<-data.frame(sapply(datos_orig,haz.cero.na))
```

Los genes seleccionados son:

```
genes_ROC<-datos[,c("g93","g76","g65","g42","g5","g16","g39",
                 "g35","g23","g52")]

genes_pAUC<-datos[,c("g93","g65","g5","g23","g42","g51","g52"
                 ,"g35","g73","g76")]

genes_AUC<-datos[,c("g93","g42","g76","g65","g16","g5","g52",
                 "g97","g39","g75")]

genes_Zstat<-datos[,c("g93","g65","g42","g97","g39","g23",
                  "g35","g76","g63","g5")]

genes_TpAUC<-datos[,c("g93","g34","g25","g10","g23","g51",
```

```
                    "g73","g84","g52","g5")]
```

Cargamos los siguientes paquetes y definimos los siguientes modelos de distribuciones gamma.

```
library(lattice)
library(flexmix)
```

```
set.seed(400)

M1<-list(FLXMRglm(genes_ROC[,1]~., family="Gamma"),FLXMRglm(
        genes_ROC[,2]~.,family="Gamma"),
        FLXMRglm(genes_ROC[,3]~., family="Gamma"),FLXMRglm(
        genes_ROC[,4]~.,family="Gamma"),
        FLXMRglm(genes_ROC[,5]~., family="Gamma"),FLXMRglm(
        genes_ROC[,6]~.,family="Gamma"),
        FLXMRglm(genes_ROC[,7]~., family="Gamma"),FLXMRglm(
        genes_ROC[,8]~.,family="Gamma"),
        FLXMRglm(genes_ROC[,9]~., family="Gamma"),FLXMRglm(
        genes_ROC[,10]~.,family="Gamma"))

M2<-list(FLXMRglm(genes_pAUC[,1]~., family="Gamma"),FLXMRglm(
        genes_pAUC[,2]~.,family="Gamma"),
        FLXMRglm(genes_pAUC[,3]~.,family="Gamma"),FLXMRglm(
        genes_pAUC[,4]~.,family="Gamma"),
        FLXMRglm(genes_pAUC[,5]~.,family="Gamma"),FLXMRglm(
        genes_pAUC[,6]~.,family="Gamma"),
        FLXMRglm(genes_pAUC[,7]~.,family="Gamma"),FLXMRglm(
        genes_pAUC[,8]~.,family="Gamma"),
        FLXMRglm(genes_pAUC[,9]~.,family="Gamma"),FLXMRglm(
        genes_pAUC[,10]~.,family="Gamma"))

M3<-list(FLXMRglm(genes_AUC[,1]~., family="Gamma"),FLXMRglm(
        genes_AUC[,2]~.,family="Gamma"),
        FLXMRglm(genes_AUC[,3]~.,family="Gamma"),FLXMRglm(
        genes_AUC[,4]~.,family="Gamma"),
        FLXMRglm(genes_AUC[,5]~.,family="Gamma"),FLXMRglm(
        genes_AUC[,6]~.,family="Gamma"),
        FLXMRglm(genes_AUC[,7]~.,family="Gamma"),FLXMRglm(
        genes_AUC[,8]~.,family="Gamma"),
        FLXMRglm(genes_AUC[,9]~.,family="Gamma"),FLXMRglm(
        genes_AUC[,10]~.,family="Gamma"))

M4<-list(FLXMRglm(genes_Zstat[,1]~., family="Gamma"),FLXMRglm(
        genes_Zstat[,2]~.,family="Gamma"),
        FLXMRglm(genes_Zstat[,3]~.,family="Gamma"),FLXMRglm(
```

```
        genes_Zstat[,4]~.,family="Gamma"),
        FLXMRglm(genes_Zstat[,5]~.,family="Gamma"),FLXMRglm(
        genes_Zstat[,6]~.,family="Gamma"),
        FLXMRglm(genes_Zstat[,7]~.,family="Gamma"),FLXMRglm(
        genes_Zstat[,8]~.,family="Gamma"),
        FLXMRglm(genes_Zstat[,9]~., family="Gamma"),FLXMRglm(
        genes_Zstat[,10]~.,family="Gamma"))

M5<-list(FLXMRglm(genes_TpAUC[,1]~.,family="Gamma"),FLXMRglm(
        genes_TpAUC[,2]~.,family="Gamma"),
        FLXMRglm(genes_TpAUC[,3]~.,family="Gamma"),FLXMRglm(
        genes_TpAUC[,4]~., family="Gamma"),
        FLXMRglm(genes_TpAUC[,5]~.,family="Gamma"),FLXMRglm(
        genes_TpAUC[,6]~.,family="Gamma"),
        FLXMRglm(genes_TpAUC[,7]~.,family="Gamma"),FLXMRglm(
        genes_TpAUC[,8]~.,family="Gamma"),
        FLXMRglm(genes_TpAUC[,9]~.,family="Gamma"),FLXMRglm(
        genes_TpAUC[,10]~.,family="Gamma"))
```

Para obtener los modelos de cada grupo y cada método de inicio se usa el siguiente código:

```
mixt_ROC_rnd<-stepFlexmix(~1,data=genes_ROC,k=1:5,model=M1,
                verbose=FALSE)

mixt_ROC_rndA<-getModel(mixt_ROC_rnd, which="AIC")
mixt_ROC_rndB<-getModel(mixt_ROC_rnd, which="BIC")
mixt_ROC_rndI<-getModel(mixt_ROC_rnd, which="ICL")
```

```
mixt_pAUC_rnd<-stepFlexmix(~1,data=genes_pAUC,k=1:5,model=M2,
        verbose=FALSE)

mixt_pAUC_rndA<-getModel(mixt_pAUC_rnd, which="AIC")
mixt_pAUC_rndB<-getModel(mixt_pAUC_rnd, which="BIC")
mixt_pAUC_rndI<-getModel(mixt_pAUC_rnd, which="ICL")
```

```
mixt_AUC_rnd<-stepFlexmix(~1,data=genes_AUC,k=1:5,model=M3,
        verbose=FALSE)

mixt_AUC_rndA<-getModel(mixt_AUC_rnd, which="AIC")
mixt_AUC_rndB<-getModel(mixt_AUC_rnd, which="BIC")
mixt_AUC_rndI<-getModel(mixt_AUC_rnd, which="ICL")
```

```
mixt_Zstat_rnd<-stepFlexmix(~1,data=genes_Zstat,k=1:5,
        model=M4,verbose=FALSE)
```

```
mixt_Zstat_rndA<-getModel(mixt_Zstat_rnd, which="AIC")
mixt_Zstat_rndB<-getModel(mixt_Zstat_rnd, which="BIC")
mixt_Zstat_rndI<-getModel(mixt_Zstat_rnd, which="ICL")
```

```
mixt_TpAUC_rnd<-stepFlexmix(~1,data=genes_TpAUC,k=1:5,
       model=M5,verbose=FALSE)

mixt_TpAUC_rndA<-getModel(mixt_TpAUC_rnd, which="AIC")
mixt_TpAUC_rndB<-getModel(mixt_TpAUC_rnd, which="BIC")
mixt_TpAUC_rndI<-getModel(mixt_TpAUC_rnd, which="ICL")
```

```
mixt_ROC_em<-initFlexmix(~1,data=genes_ROC,k=1:5,model=M1,
       init="tol.em",verbose=FALSE)

mixt_ROC_emA<-getModel(mixt_ROC_em,which="AIC")
mixt_ROC_emB<-getModel(mixt_ROC_em,which="BIC")
mixt_ROC_emI<-getModel(mixt_ROC_em,which="ICL")
```

```
mixt_pAUC_em<-initFlexmix(~1,data=genes_pAUC,k=1:5,model=M2,
       init="tol.em",verbose=FALSE)

mixt_pAUC_emA<-getModel(mixt_pAUC_em,which="AIC")
mixt_pAUC_emB<-getModel(mixt_pAUC_em,which="BIC")
mixt_pAUC_emI<-getModel(mixt_pAUC_em,which="ICL")
```

```
mixt_AUC_em<-initFlexmix(~1,data=genes_AUC,k=1:5,model=M3,
       init="tol.em",verbose=FALSE)

mixt_AUC_emA<-getModel(mixt_AUC_em,which="AIC")
mixt_AUC_emB<-getModel(mixt_AUC_em,which="BIC")
mixt_AUC_emI<-getModel(mixt_AUC_em,which="ICL")
```

```
mixt_Zstat_em<-initFlexmix(~1,data=genes_Zstat,k=1:5,
       model=M4,init="tol.em",verbose=FALSE)

mixt_Zstat_emA<-getModel(mixt_Zstat_em,which="AIC")
mixt_Zstat_emB<-getModel(mixt_Zstat_em,which="BIC")
mixt_Zstat_emI<-getModel(mixt_Zstat_em,which="ICL")
```

```
mixt_TpAUC_em<-initFlexmix(~1,data=genes_TpAUC,k=1:5,
       model=M5,init="tol.em",verbose=FALSE)

mixt_TpAUC_emA<-getModel(mixt_TpAUC_em,which="AIC")
mixt_TpAUC_emB<-getModel(mixt_TpAUC_em,which="BIC")
mixt_TpAUC_emI<-getModel(mixt_TpAUC_em,which="ICL")
```

```
mixt_ROC_cem<-initFlexmix(~1,data=genes_ROC,k=1:5,model=M1,
       init="cem",verbose=FALSE)

mixt_ROC_cemA<-getModel(mixt_ROC_cem,which="AIC")
mixt_ROC_cemB<-getModel(mixt_ROC_cem,which="BIC")
mixt_ROC_cemI<-getModel(mixt_ROC_cem,which="ICL")
```

```
mixt_pAUC_cem<-initFlexmix(~1,data=genes_pAUC,k=1:5,model=M2,
       init="cem",verbose=FALSE)

mixt_pAUC_cemA<-getModel(mixt_pAUC_cem,which="AIC")
mixt_pAUC_cemB<-getModel(mixt_pAUC_cem,which="BIC")
mixt_pAUC_cemI<-getModel(mixt_pAUC_cem,which="ICL")
```

```
mixt_AUC_cem<-initFlexmix(~1,data=genes_AUC,k=1:5,model=M3,
       init="cem",verbose=FALSE)

mixt_AUC_cemA<-getModel(mixt_AUC_cem,which="AIC")
mixt_AUC_cemB<-getModel(mixt_AUC_cem,which="BIC")
mixt_AUC_cemI<-getModel(mixt_AUC_cem,which="ICL")
```

```
mixt_Zstat_cem<-initFlexmix(~1,data=genes_Zstat,k=1:5,
       model=M4,init="cem",verbose=FALSE)

mixt_Zstat_cemA<-getModel(mixt_Zstat_cem,which="AIC")
mixt_Zstat_cemB<-getModel(mixt_Zstat_cem,which="BIC")
mixt_Zstat_cemI<-getModel(mixt_Zstat_cem,which="ICL")
```

```
mixt_TpAUC_cem<-initFlexmix(~1,data=genes_TpAUC,k=1:5,
       model=M5,init="cem",verbose=FALSE)

mixt_TpAUC_cemA<-getModel(mixt_TpAUC_cem,which="AIC")
mixt_TpAUC_cemB<-getModel(mixt_TpAUC_cem,which="BIC")
mixt_TpAUC_cemI<-getModel(mixt_TpAUC_cem,which="ICL")
```

```
mixt_ROC_sem<-initFlexmix(~1,data=genes_ROC,k=1:5,model=M1,
       init="sem",verbose=FALSE)

mixt_ROC_semA<-getModel(mixt_ROC_sem,which="AIC")
mixt_ROC_semB<-getModel(mixt_ROC_sem,which="BIC")
mixt_ROC_semI<-getModel(mixt_ROC_sem,which="ICL")
```

```
mixt_pAUC_sem<-initFlexmix(~1,data=genes_pAUC,k=1:5,model=M2,
       init="sem",verbose=FALSE)

mixt_pAUC_semA<-getModel(mixt_pAUC_sem,which="AIC")
```

```
mixt_pAUC_semB<-getModel(mixt_pAUC_sem,which="BIC")
mixt_pAUC_semI<-getModel(mixt_pAUC_sem,which="ICL")
```

```
mixt_AUC_sem<-initFlexmix(~1,data=genes_AUC,k=1:5,model=M3,
       init="sem",verbose=FALSE)

mixt_AUC_semA<-getModel(mixt_AUC_sem,which="AIC")
mixt_AUC_semB<-getModel(mixt_AUC_sem,which="BIC")
mixt_AUC_semI<-getModel(mixt_AUC_sem,which="ICL")
```

```
mixt_Zstat_sem<-initFlexmix(~1,data=genes_Zstat,k=1:5,
       model=M4,init="sem",verbose=FALSE)

mixt_Zstat_semA<-getModel(mixt_Zstat_sem,which="AIC")
mixt_Zstat_semB<-getModel(mixt_Zstat_sem,which="BIC")
mixt_Zstat_semI<-getModel(mixt_Zstat_sem,which="ICL")
```

```
mixt_TpAUC_sem<-initFlexmix(~1,data=genes_TpAUC,k=1:5,
       model=M5,init="sem",verbose=FALSE)

mixt_TpAUC_semA<-getModel(mixt_TpAUC_sem,which="AIC")
mixt_TpAUC_semB<-getModel(mixt_TpAUC_sem,which="BIC")
mixt_TpAUC_semI<-getModel(mixt_TpAUC_sem,which="ICL")
```

Escogemos un modelo para cada criterio y cada grupo de genes. Como los modelos del BIC e ICL coinciden bastará con uno de ellos.

```
ROC_A<-mixt_ROC_cemA
ROC<-mixt_ROC_cemI

pAUC_A<-mixt_pAUC_semA
pAUC<-mixt_pAUC_emI

AUC_A<-mixt_AUC_rndA
AUC<-mixt_AUC_rndI

Zstat_A<-mixt_Zstat_semA
Zstat<-mixt_Zstat_semI

TpAUC_A<-mixt_TpAUC_semA
TpAUC<-mixt_TpAUC_emI
```

Ajuste de los modelos obtenidos. Estudio de la sensibilidad y especificidad

Finalmente nos centraremos en los modelos seleccionados por el criterio BIC o ICL ya que el AIC sobreestima el orden de los modelos. Con el siguiente comando se obtienen las tablas que ayudan a determinar qué componente se corresponde con los tejidos enfermos, que en los datos reales se corresponde con el valor 1, y los sanos, con valor 0.

```
table(datos[,"d"],clusters(pAUC))
```

```
    1  2  3
0   1  3 19
1  23  3  4
```

Cambiamos la notación de la clasificación de los datos según consideremos qué componentes se corresponden con un tipo de tejido u otro.

```
c_pAUC<-clusters(pAUC)
for (i in seq(1,53)){
        if(clusters(pAUC)[i]==1){
                c_pAUC[i]<-1
        } else{
                c_pAUC[i]<-0
        }
}
```

```
c_AUC<-clusters(AUC)
for (i in seq(1,53)){
        if(clusters(AUC)[i]==2){
                c_AUC[i]<-0
        } else{
                c_AUC[i]<-1
        }
}
```

```
c_TpAUC<-clusters(TpAUC)
for (i in seq(1,53)){
        if(clusters(TpAUC)[i]==2){
                c_TpAUC[i]<-1
        } else{
                c_TpAUC[i]<-0
        }
}
```

Con la siguiente función se puede determinar la clasificación correspondiente para los otros dos grupos (que tienen sólo dos componentes).

```
cambio<-function(clusters_orig) {
```

```
        c_model<-clusters_orig
        a<-0
        b<-0
        for (i in seq(1,53)){
            if (datos[i,"d"] ==clusters_orig[i]){
               a<-a+1
            } else if (clusters_orig[i]==2 &datos[i,"d"]==1){
               b<-b+1
            }
         }

         if(a>b){
             print("Tejido enfermo Componente 1")
             for(i in seq(1,53)){
                if (clusters_orig[i]==2){
                   c_model[i]<-0
                }
              }
         } else if (a<b){
             print("Tejido enfermo Componente 2")
             for(i in seq(1,53)){
                if (clusters_orig[i]==2){
                   c_model[i]<-1
                } else if (clusters_orig[i]==1) {
                   c_model[i]<-0
                }
             }
         }
         return(c_model)
         }
```

La aplicamos en los dos casos restantes.

```
c_ROC<-cambio(clusters(ROC))
c_Zstat<-cambio(clusters(Zstat))
```

Finalmente se obtiene la clasificación de cada modelo manteniendo la notación original: 0 para el tejido sano y 1 para el enfermo.

A continuación se calcula la sensibilidad, especificidad y valor predictivo global de cada una de las clasificaciones obtenidas previamente, usando la siguiente función:

```
Sensibilidad_Especificidad_Global<-function(c_modelo){
       d<-datos[,"d"]
       vn<-0
       vp<-0
       fn<-0
```

```
        fp<-0
        for (i in seq(1,53)){
                if (c_modelo[i]==1 & d[i]==1){
                        vn<-vn+1
                } else if (c_modelo[i]==0 & d[i]==0){
                        vp<-vp+1
                } else if (c_modelo[i]==1 & d[i]==0){
                        fn<-fn+1
                } else if (c_modelo[i]==0 & d[i]==1){
                        fp<-fp+1
                }
        }
        return(c(vp/(vp+fn),vn/(vn+fp),(vp+vn)/(vp+vn+fp+fn)))
}
```

Para representar las curvas ROC de cada modelo se usa el paquete *pROC*.

```
library(pROC)

Curva_R<-roc(datos[,c("d")],c_ROC, smooth=FALSE,auc = TRUE)
Curva_p<-roc(datos[,c("d")],c_pAUC, smooth=FALSE,auc = TRUE)
Curva_A<-roc(datos[,c("d")],c_AUC, smooth=FALSE,auc = TRUE)
Curva_Z<-roc(datos[,c("d")],c_Zstat, smooth=FALSE,auc = TRUE)
Curva_T<-roc(datos[,c("d")],c_TpAUC, smooth=FALSE,auc = TRUE)

par(pty="s")
plot.roc(Curva_R,auc.polygon = TRUE,print.auc=TRUE,
        main="Curva ROC grupo genes ROC")
plot.roc(Curva_p,auc.polygon = TRUE,print.auc=TRUE,
        main="Curva ROC grupo genes pAUC")
plot.roc(Curva_A,auc.polygon = TRUE,print.auc=TRUE,
        main="Curva ROC grupo genes AUC")
plot.roc(Curva_Z,auc.polygon = TRUE,print.auc=TRUE,
        main="Curva ROC grupo genes Z-stat")
plot.roc(Curva_T,auc.polygon = TRUE,print.auc=TRUE,
        main="Curva ROC grupo genes TpAUC")
```

Estabilidad de los modelos: técnica Bootstrap

Con el comando *rflexmix* se realiza un remuestreo aleatorio de los modelos seleccionados.

```
ROC_boot<-rflexmix(ROC)
pAUC_boot<-rflexmix(pAUC)
AUC_boot<-rflexmix(AUC)
Zstat_boot<-rflexmix(Zstat)
```

```
TpAUC_boot<-rflexmix(TpAUC)
```

Adaptamos las clasificaciones del remuetreo de los modelos a la notación inicial, considerando que las componentes designan el mismo tipo de tejido que antes.

```
c_pAUC_boot<-pAUC_boot$class
for (i in seq(1,53)){
        if(pAUC_boot$class[i]==3){
                c_pAUC_boot[i]<-0
        } else{
                c_pAUC_boot[i]<-1
        }
}
```

```
c_AUC_boot<-AUC_boot$class
for (i in seq(1,53)){
        if(pAUC_boot$class[i]==2){
                c_AUC_boot[i]<-0
        } else{
                c_AUC_boot[i]<-1
        }
}
```

```
c_TpAUC_boot<-TpAUC_boot$class
for (i in seq(1,53)){
        if(TpAUC_boot$class[i]==2){
                c_TpAUC_boot[i]<-1
        } else{
                c_TpAUC_boot[i]<-0
        }
}
```

```
c_ROC_boot<-ROC_boot$class
for (i in seq(1,53)){
        if(ROC_boot$class[i]==2){
                c_ROC_boot[i]<-0
        }
}
```

```
c_Zstat_boot<-Zstat_boot$class
for (i in seq(1,53)){
        if(Zstat_boot$class[i]==2){
                c_Zstat_boot[i]<-0
        }
}
```

De estas clasificaciones y con la función *Sensibilidad_Especificidad_Global* se obtienen los valores de sensibilidad, especificidad y predicción global de cada modelo.

Apéndice B

Código para GMM

En este anexo se encuentra parte del código y resultados obtenidos para los modelos de mixturas normales.

Se definen los modelos correspondientes sin definir una familia paramétrica, ya que por defecto se implementa una distribución normal.

```
M1N<-list(FLXMRglm(genes_ROC[,1]~.),FLXMRglm(genes_ROC[,2]~.),
     FLXMRglm(genes_ROC[,3]~.),FLXMRglm(genes_ROC[,4]~.),
     FLXMRglm(genes_ROC[,5]~.),FLXMRglm(genes_ROC[,6]~.),
     FLXMRglm(genes_ROC[,7]~.),FLXMRglm(genes_ROC[,8]~.),
     FLXMRglm(genes_ROC[,9]~.),FLXMRglm(genes_ROC[,10]~.))

M2N<-list(FLXMRglm(genes_pAUC[,1]~.),FLXMRglm(genes_pAUC
     [,2]~.),FLXMRglm(genes_pAUC[,3]~.),FLXMRglm(genes_pAUC
     [,4]~.),FLXMRglm(genes_pAUC[,5]~.),FLXMRglm(genes_pAUC
     [,6]~.),FLXMRglm(genes_pAUC[,7]~.),FLXMRglm(genes_pAUC
     [,8]~.),FLXMRglm(genes_pAUC[,9]~.),FLXMRglm(genes_pAUC
     [,10]~.))

M3N<-list(FLXMRglm(genes_AUC[,1]~.),FLXMRglm(genes_AUC[,2]~.),
     FLXMRglm(genes_AUC[,3]~.),FLXMRglm(genes_AUC[,4]~.),
     FLXMRglm(genes_AUC[,5]~.),FLXMRglm(genes_AUC[,6]~.),
     FLXMRglm(genes_AUC[,7]~.),FLXMRglm(genes_AUC[,8]~.),
     FLXMRglm(genes_AUC[,9]~.),FLXMRglm(genes_AUC[,10]~.))

M4N<-list(FLXMRglm(genes_Zstat[,1]~.),FLXMRglm(genes_Zstat
     [,2]~.),FLXMRglm(genes_Zstat[,3]~.),FLXMRglm(genes_Zstat
     [,4]~.),FLXMRglm(genes_Zstat[,5]~.),FLXMRglm(genes_Zstat
     [,6]~.),FLXMRglm(genes_Zstat[,7]~.),FLXMRglm(genes_Zstat
     [,8]~.),FLXMRglm(genes_Zstat[,9]~.),FLXMRglm(genes_Zstat
     [,10]~.))

M5N<-list(FLXMRglm(genes_TpAUC[,1]~.),FLXMRglm(genes_TpAUC
```

```
      [,2]~.),FLXMRglm(genes_TpAUC[,3]~.),FLXMRglm(genes_TpAUC
      [,4]~.),FLXMRglm(genes_TpAUC[,5]~.),FLXMRglm(genes_TpAUC
      [,6]~.),FLXMRglm(genes_TpAUC[,7]~.),FLXMRglm(genes_TpAUC
      [,8]~.),FLXMRglm(genes_TpAUC[,9]~.),FLXMRglm(genes_TpAUC
      [,10]~.))
```

Se definen los modelos de mixturas obtenidos con el algoritmo EM y una inicialización de tipo aleatorio de la siguiente forma:

```
mixt_ROC_rndN<-stepFlexmix(~1,data=genes_ROC,k=1:5,model=M1N,
       verbose=FALSE)

mixt_ROC_rndNA<-getModel(mixt_ROC_rndN, which="AIC")
mixt_ROC_rndNB<-getModel(mixt_ROC_rndN, which="BIC")
mixt_ROC_rndNI<-getModel(mixt_ROC_rndN, which="ICL")
```

```
mixt_pAUC_rndN<-stepFlexmix(~1,data=genes_pAUC,k=1:5,
                model=M2N,verbose=FALSE)

mixt_pAUC_rndNA<-getModel(mixt_pAUC_rndN, which="AIC")
mixt_pAUC_rndNB<-getModel(mixt_pAUC_rndN, which="BIC")
mixt_pAUC_rndNI<-getModel(mixt_pAUC_rndN, which="ICL")
```

```
mixt_AUC_rndN<-stepFlexmix(~1,data=genes_AUC,k=1:5,model=M3N,
       verbose=FALSE)

mixt_AUC_rndNA<-getModel(mixt_AUC_rndN, which="AIC")
mixt_AUC_rndNB<-getModel(mixt_AUC_rndN, which="BIC")
mixt_AUC_rndNI<-getModel(mixt_AUC_rndN, which="ICL")
```

```
mixt_Zstat_rndN<-stepFlexmix(~1,data=genes_Zstat,k=1:5,
                 model=M4N, verbose=FALSE)

mixt_Zstat_rndNA<-getModel(mixt_Zstat_rndN, which="AIC")
mixt_Zstat_rndNB<-getModel(mixt_Zstat_rndN, which="BIC")
mixt_Zstat_rndNI<-getModel(mixt_Zstat_rndN, which="ICL")
```

```
mixt_TpAUC_rndN<-stepFlexmix(~1,data=genes_TpAUC,k=1:5,
                 model=M5N, verbose=FALSE)

mixt_TpAUC_rndNA<-getModel(mixt_TpAUC_rndN, which="AIC")
mixt_TpAUC_rndNB<-getModel(mixt_TpAUC_rndN, which="BIC")
mixt_TpAUC_rndNI<-getModel(mixt_TpAUC_rndN, which="ICL")
```

Realizando este proceso con los otros 3 métodos de inicio, se seleccionan los siguientes modelos de mixturas gausianas para los tres criterios usados.

```
mixt_ROC_emN<-initFlexmix(~1,data=genes_ROC,k=1:5,model=M1N,
       init="tol.em", verbose=FALSE)

mixt_ROC_emNA<-getModel(mixt_ROC_emN,which="AIC")
mixt_ROC_emNB<-getModel(mixt_ROC_emN,which="BIC")
mixt_ROC_emNI<-getModel(mixt_ROC_emN,which="ICL")


mixt_pAUC_emN<-initFlexmix(~1,data=genes_pAUC,k=1:5,model=M2N,
       init="tol.em", verbose=FALSE)

mixt_pAUC_emNA<-getModel(mixt_pAUC_emN,which="AIC")
mixt_pAUC_emNB<-getModel(mixt_pAUC_emN,which="BIC")
mixt_pAUC_emNI<-getModel(mixt_pAUC_emN,which="ICL")


mixt_AUC_emN<-initFlexmix(~1,data=genes_AUC,k=1:5,model=M3N,
       init="tol.em", verbose=FALSE)

mixt_AUC_emNA<-getModel(mixt_AUC_emN,which="AIC")
mixt_AUC_emNB<-getModel(mixt_AUC_emN,which="BIC")
mixt_AUC_emNI<-getModel(mixt_AUC_emN,which="ICL")


mixt_Zstat_emN<-initFlexmix(~1,data=genes_Zstat,k=1:5,
       model=M4N,init="tol.em", verbose=FALSE)

mixt_Zstat_emNA<-getModel(mixt_Zstat_emN,which="AIC")
mixt_Zstat_emNB<-getModel(mixt_Zstat_emN,which="BIC")
mixt_Zstat_emNI<-getModel(mixt_Zstat_emN,which="ICL")


mixt_TpAUC_emN<-initFlexmix(~1,data=genes_TpAUC,k=1:5,
       model=M5N,init="tol.em", verbose=FALSE)

mixt_TpAUC_emNA<-getModel(mixt_TpAUC_emN,which="AIC")
mixt_TpAUC_emNB<-getModel(mixt_TpAUC_emN,which="BIC")
mixt_TpAUC_emNI<-getModel(mixt_TpAUC_emN,which="ICL")


mixt_ROC_cemN<-initFlexmix(~1,data=genes_ROC,k=1:5,model=M1N,
       init="cem", verbose=FALSE)

mixt_ROC_cemNA<-getModel(mixt_ROC_cemN,which="AIC")
```

```
mixt_ROC_cemNB<-getModel(mixt_ROC_cemN,which="BIC")
mixt_ROC_cemNI<-getModel(mixt_ROC_cemN,which="ICL")


mixt_pAUC_cemN<-initFlexmix(~1,data=genes_pAUC,k=1:5,
       model=M2N,init="cem", verbose=FALSE)

mixt_pAUC_cemNA<-getModel(mixt_pAUC_cemN,which="AIC")
mixt_pAUC_cemNB<-getModel(mixt_pAUC_cemN,which="BIC")
mixt_pAUC_cemNI<-getModel(mixt_pAUC_cemN,which="ICL")


mixt_AUC_cemN<-initFlexmix(~1,data=genes_AUC,k=1:5,model=M3N,
       init="cem", verbose=FALSE)

mixt_AUC_cemNA<-getModel(mixt_AUC_cemN,which="AIC")
mixt_AUC_cemNB<-getModel(mixt_AUC_cemN,which="BIC")
mixt_AUC_cemNI<-getModel(mixt_AUC_cemN,which="ICL")


mixt_Zstat_cemN<-initFlexmix(~1,data=genes_Zstat,k=1:5,
       model=M4N,init="cem", verbose=FALSE)

mixt_Zstat_cemNA<-getModel(mixt_Zstat_cemN,which="AIC")
mixt_Zstat_cemNB<-getModel(mixt_Zstat_cemN,which="BIC")
mixt_Zstat_cemNI<-getModel(mixt_Zstat_cemN,which="ICL")


mixt_TpAUC_cemN<-initFlexmix(~1,data=genes_TpAUC,k=1:5,
       model=M5N,init="cem", verbose=FALSE)

mixt_TpAUC_cemNA<-getModel(mixt_TpAUC_cemN,which="AIC")
mixt_TpAUC_cemNB<-getModel(mixt_TpAUC_cemN,which="BIC")
mixt_TpAUC_cemNI<-getModel(mixt_TpAUC_cemN,which="ICL")


mixt_ROC_semN<-initFlexmix(~1,data=genes_ROC,k=1:5,model=M1N,
       init="sem", verbose=FALSE)

mixt_ROC_semNA<-getModel(mixt_ROC_semN,which="AIC")
mixt_ROC_semNB<-getModel(mixt_ROC_semN,which="BIC")
mixt_ROC_semNI<-getModel(mixt_ROC_semN,which="ICL")
```

```
mixt_pAUC_semN<-initFlexmix(~1,data=genes_pAUC,k=1:5,
       model=M2N,init="sem", verbose=FALSE)

mixt_pAUC_semNA<-getModel(mixt_pAUC_semN,which="AIC")
mixt_pAUC_semNB<-getModel(mixt_pAUC_semN,which="BIC")
mixt_pAUC_semNI<-getModel(mixt_pAUC_semN,which="ICL")


mixt_AUC_semN<-initFlexmix(~1,data=genes_AUC,k=1:5,model=M3N,
       init="sem", verbose=FALSE)

mixt_AUC_semNA<-getModel(mixt_AUC_semN,which="AIC")
mixt_AUC_semNB<-getModel(mixt_AUC_semN,which="BIC")
mixt_AUC_semNI<-getModel(mixt_AUC_semN,which="ICL")


mixt_Zstat_semN<-initFlexmix(~1,data=genes_Zstat,k=1:5,
       model=M4N,init="sem",   verbose=FALSE)

mixt_Zstat_semNA<-getModel(mixt_Zstat_semN,which="AIC")
mixt_Zstat_semNB<-getModel(mixt_Zstat_semN,which="BIC")
mixt_Zstat_semNI<-getModel(mixt_Zstat_semN,which="ICL")


mixt_TpAUC_semN<-initFlexmix(~1,data=genes_TpAUC,k=1:5,
       model=M5N,init="sem",   verbose=FALSE)

mixt_TpAUC_semNA<-getModel(mixt_TpAUC_semN,which="AIC")
mixt_TpAUC_semNB<-getModel(mixt_TpAUC_semN,which="BIC")
mixt_TpAUC_semNI<-getModel(mixt_TpAUC_semN,which="ICL")
```

Los modelos obtenidos son:

	AIC	BIC	ICL
ROC	prior size post>0 ratio Comp.1 0.0956 5 11 0.455 Comp.2 0.4527 24 25 0.960 Comp.3 0.1320 7 23 0.304 Comp.4 0.3197 17 24 0.708 'log Lik.' 13.58702 (df=83) AIC: 138.826 BIC: 302.3602	prior size post>0 ratio Comp.1 0.453 24 25 0.960 Comp.2 0.190 10 24 0.417 Comp.3 0.357 19 25 0.760 'log Lik.' -12.81486 (df=62) AIC: 149.6297 BIC: 271.7878	prior size post>0 ratio Comp.1 0.453 24 25 0.960 Comp.2 0.190 10 24 0.417 Comp.3 0.357 19 25 0.760 'log Lik.' -12.81486 (df=62) AIC: 149.6297 BIC: 271.7878 ICL=272,0061
pAUC	prior size post>0 ratio Comp.1 0.3457 18 26 0.692 Comp.2 0.4329 23 24 0.958 Comp.3 0.0755 4 7 0.571 Comp.4 0.1459 8 22 0.364 'log Lik.' 95.60501 (df=83) AIC: -25.21003 BIC: 138.3242	prior size post>0 ratio Comp.1 0.4160 22 28 0.786 Comp.2 0.0941 5 10 0.500 Comp.3 0.4899 26 26 1.000 'log Lik.' 55.49867 (df=62) AIC: 13.00266 BIC: 135.1608	prior size post>0 ratio Comp.1 0.4160 22 28 0.786 Comp.2 0.0941 5 10 0.500 Comp.3 0.4899 26 26 1.000 'log Lik.' 55.49867 (df=62) AIC: 13.00266 BIC: 135.1608 ICL=135,2681
AUC	prior size post>0 ratio Comp.1 0.0771 4 8 0.500 Comp.2 0.4291 23 24 0.958 Comp.3 0.2089 11 18 0.611 Comp.4 0.0931 5 8 0.625 Comp.5 0.1918 10 13 0.769 'log Lik.' -45.00545 (df=104) AIC: 298.0109 BIC: 502.9213	prior size post>0 ratio Comp.1 0.416 22 26 0.846 Comp.2 0.151 8 20 0.400 Comp.3 0.433 23 25 0.920 'log Lik.' -106.6145 (df=62) AIC: 337.2289 BIC: 459.387	prior size post>0 ratio Comp.1 0.416 22 26 0.846 Comp.2 0.151 8 20 0.400 Comp.3 0.433 23 25 0.920 'log Lik.' -106.6145 (df=62) AIC: 337.2289 BIC: 459.387 ICL=459,6653
Z-stat	prior size post>0 ratio Comp.1 0.0566 3 4 0.750 Comp.2 0.1877 10 15 0.667 Comp.3 0.2276 12 22 0.545 Comp.4 0.4714 25 26 0.962 Comp.5 0.0566 3 3 1.000 'log Lik.' 82.79976 (df=104) AIC: 42.40049 BIC: 247.3108	prior size post>0 ratio Comp.1 0.4291 23 30 0.767 Comp.2 0.0999 5 19 0.263 Comp.3 0.4710 25 27 0.926 'log Lik.' 21.98182 (df=62) AIC: 80.03637 BIC: 202.1945	prior size post>0 ratio Comp.1 0.4291 23 30 0.767 Comp.2 0.0999 5 19 0.263 Comp.3 0.4710 25 27 0.926 'log Lik.' 21.98182 (df=62) AIC: 80.03637 BIC: 202.1945 ICL=202,8936
TpAUC	prior size post>0 ratio Comp.1 0.132 7 19 0.368 Comp.2 0.111 6 17 0.353 Comp.3 0.151 8 17 0.471 Comp.4 0.378 20 21 0.952 Comp.5 0.228 12 26 0.462 'log Lik.' 26.24903 (df=104) AIC: 155.5019 BIC: 360.4123	prior size post>0 ratio Comp.1 0.129 7 11 0.636 Comp.2 0.294 15 34 0.441 Comp.3 0.242 13 24 0.542 Comp.4 0.335 18 21 0.857 'log Lik.' -1.849145 (df=83) AIC: 169.6983 BIC: 333.2325	prior size post>0 ratio Comp.1 0.129 7 11 0.636 Comp.2 0.294 15 34 0.441 Comp.3 0.242 13 24 0.542 Comp.4 0.335 18 21 0.857 'log Lik.' -1.849145 (df=83) AIC: 169.6983 BIC: 333.2325 ICL=335,1078

► Todos los modelos anteriores salvo el grupo TpAUC (para los tres criterios), y el ROC, pAUC y AUC (para el criterio AIC), han sido obtenidos mediante una inicialización de tipo CEM.

► Los modelos seleccionados por los criterios BIC e ICL son los mismos en todos los casos.

► Se observa que los valores del AIC, BIC e ICL son mucho más altos en comparación con los obtenidos para los modelos de mixturas con familia parámetrica gamma. Por tanto, los modelos de mixturas con distribuciones gamma se ajustan mejor a los datos.