Máster Universitario en Estadística Aplicada
Departamento de Estadística e Investigación Operativa

Modelos de mixturas finitas para la caracterización y mejora de la redes de monitorización de la calidad del aire

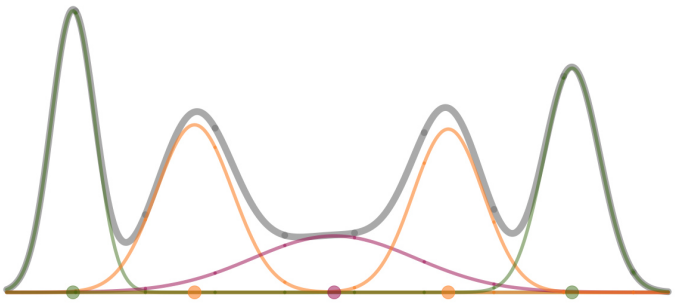
Tutores:

Dr. Rafael Pino Mejías
Universidad de Sevilla

Dra. Yolanda Román Montoya
Universidad de Granada

Alumno:

Álvaro Gómez Losada



Facultad de Ciencias
Granada, junio 2014

Modelos de mixturas finitas para la caracterización y mejora de la redes de monitorización de la calidad del aire

Memoria realizada por Álvaro Gómez Losada bajo la dirección de D. Rafael Pino Mejías, profesor del Departamento de Estadística e Investigación Operativa de la Universidad de Sevilla, y de D.^{a} Yolanda Roldán Montoya, profesora del Departamento de Estadística e Investigación Operativa de la Universidad de Granada.

Granada, junio 2014

Álvaro Gómez Losada

V^{o}B^o directores del trabajo

Fdo. D. Rafael Pino Mejías Fdo. D.^{a} Yolanda Román Montoya

El presente Trabajo Fin de Máster está avalado hasta la fecha de su lectura por la siguiente publicación:

Gómez-Losada, A., **Lozano-García, A.**, **Pino-Mejías, R.**, **Contreras-González, J**. 2014. Finite mixture models to characterize and refine air quality monitoring networks. *Science of the Total Environment*, 485-486: 292-9. DOI: 10.1016/j.scitotenv.2014.03.091.

Disponible en `http://dx.doi.org/10.1016/j.scitotenv.2014.03.091`

A mi padre

Abstract

Background Existing air quality monitoring programs are, on occasion, not updated according to local, varying conditions and as such the monitoring programs become non-informative over time, under-detecting new sources of pollutants or duplicating information. Furthermore, inadequate maintenance may cause the monitoring equipment to be utterly deficient in providing information. To deal with these issues, a combination of formal statistical methods is used to optimize resources for monitoring and to characterize the monitoring networks, introducing new criteria for their refinement.

Methods Monitoring data were obtained on key pollutants such as carbon monoxide (CO), nitrogen dioxide (NO_2), ozone (O_3), particulate matter (PM_{10}) and sulfur dioxide (SO_2) from 12 air quality monitoring sites in Seville (Spain) during 2012. A total of 49 data sets were fit to mixture models of Gaussian distribution using the expectation-maximization (EM) algorithm. To summarize these 49 models, the mean (μ_m) and coefficient of variation (cv_m) were calculated for each mixture and carried out a hierarchical clustering analysis (HCA) to study the grouping of the sites according to these statistics. To handle the lack of observational data from the sites with unmonitored pollutants, the missing statistical values were imputed by applying the random forests technique and then later, a principal component analysis (PCA) was carried out to better understand the relationship between the level of pollution and the classification of monitoring sites. All of the techniques were applied using free, open-source, statistical software R.

Results and conclusions One example of source attribution and contribution is analyzed using mixture models and the potential for mixture models is posed in characterizing pollution trends. The mixture statistics μ_m and cv_m have proven to be a fingerprint for every model and this work presents a novel use of it and represents a promising approach to characterizing mixture models in the air quality management discipline. The imputation technique used is allowed for estimating the missing information from key unmonitored pollutants to gather information about unknown pollution levels and to suggest new possible monitoring configurations for this network. Posterior PCA confirmed the misclassification of one site detected with HCA.

Resumen

Antecedentes Los planes de monitorización de la calidad del aire, en ocasiones, no son convenientemente actualizados en concordancia con las cambiantes condiciones locales, repercutiendo en la información atmosférica que proporcionan, bien dejando de detectar nuevas fuentes de contaminación o duplicando cierta información. Además, posibles mantenimientos deficientes del equipamiento de las redes de monitorización suponen a aquel un inconveniente añadido. Para abodar estos aspectos, se ha recurrido a una combinación de métodos estadísticos para la optimización de los recursos empleados en la monitorización, introduciendo nuevos criterios para su mejora.

Métodos Datos de monitorización de contaminantes clave como el monóxido de carbono (CO), dióxido de nitrógeno (NO_2), ozono (O_3), material particulado (PM_{10}) y dióxido de azufre (SO_2) fueron obtenidos de 12 estaciones de monitorización de la calidad del aire en Sevilla (España). Un total de 49 conjuntos de datos fueron modelizados mediante mixturas finitas gaussianas utilizando el algoritmo de esperanza-maximización (EM). Para resumir estos 49 modelos, la media (μ_m) y coeficiente de variación (cv_m) de cada mixtura fueron calculados, y a partir de ellos, se realizó un análisis clúster jerárquico (ACJ) para estudiar el agrupamiento de las estaciones de acuerdo con estos estadísticos. El valor de los parámetros no monitorizados en las estaciones de medición fueron imputados aplicando un algoritmo basado en bosques aleatorios, utilizando los valores de μ_m y cv_m conocidos. Posteriormente, el análisis de componentes principales (ACP) permitió comprender la relación intrínseca entre las estaciones de la red, así como la concordancia en su clasificación. Todas las técnicas fueron aplicadas utilizando el software estadístico gratuito y de código abierto R.

Resultados y conclusiones Se ha analizado un ejemplo de atribución y contribución de fuentes utilizando la modelización mediante mixturas finitas, y el potencial de estos modelos es propuesto para caracterizar tendencias de contaminación. Los estadísticos de la mixturas μ_m y cv_m representan su huella dactilar, y su empleo es nuevo en la caracterización de los modelos mixtos en el área de la gestión de la calidad del aire. La técnica de imputación empleada ha permitido la estimación de valores de concentración de parámetros no monitorizados y el planteamiento de nuevos esquemas de monitorización para esta red. El empleo posterior del ACP ha confirmado una clasificación errónea de una estación detectada inicialmente mediante el ACJ.

Abreviaturas y Símbolos

EMV	estimadores de máxima verosimilitud
log	ln
y	muestra observada
$\phi(\cdot\|\mu,\sigma)$	función de densidad normal de parámetros μ y σ
$L(\theta\|y)$	función de verosimilitud de una muestra y
$L(\Psi\|y)$	función de verosimilitud de la mixtura
$\ell(\Psi\|y)$	función de log-verosimilitud de la mixtura
z	vector de variables latentes
x	vector de datos completos
μ_m	media de la mixtura
σ_m^2	varianza de la mixtura
cv_m	coeficiente de variación de la mixtura
$L(\Psi\|y,z)$	función de verosimilitud de los datos completos
$\ell(\Psi\|y,z)$	función de log-verosimilitud de los datos completos
$\mathsf{E}(\cdot)$	operador Esperanza
$Q(\Psi\|\Psi^{(h)})$	función Q
λ	multiplicador de Lagrange
I_{oc}	matriz de información de los datos completos
$\widehat{se_B}$	estimador *bootstrap* del error estándar
e.e. SEM	error estándar calculado mediante método SEM
CO	monóxido de carbono
NO_2	dióxido de nitrógeno
O_3	ozono
PM_{10}	material particulado de diámetro 10μm o inferior
SO_2	dióxido de azufre
EM	esperanza-maximización
ACJ	análisis clúster jerárquico
BA	bosques aleatorios
ACP	análisis de componentes principales

Índice

1

Introducción

Las distribuciones de mixturas finitas se han empleado para la modelización de datos heterogéneos en muchas areas de conocimiento y numerosos ejemplos son citados en MacLachlan y Peel (2000). Más recientemente, destaca su aplicación en bioinformática y genética (Delmar et.al, 2005). No obstante, su utilización en la disciplina ambiental ha sido limitada (Li et al, 2013) y, en particular, el potencial de los modelos mixtos no ha sido explotado en el área de la investigación atmosférica. En este estudio, se aplica parte de su potencial a las redes de monitorización atmosférica.

Como en otras tantas áreas de estudio, con frecuencia, no es suficiente explicar la distribución de unos datos mediante una única distribución estadística. En particular, si estos datos pueden ser agrupados en subpoblaciones o asociados a distintos *procesos generadores*, es necesaria la utilización de una composición de distribuciones. Tales composiciones suelen ser descritas mediante los modelos de mixturas, los cuales se definen por los parámetros de cada componente y las proporciones en las que cada una de ellas contribuye a la distribución general. Este concepto conduce al agrupamiento de partes del conjunto de observaciones basado en alguna característica común desconocida. El conjunto de parámetros que definen a estos modelos pueden resumirse generalmente mediante sus *momentos*, constituyendo su huella dactilar. Las distribuciones mixtas pueden ser estimadas mediante muchas técnicas, tales como métodos gráficos, el método de los momentos, de máxima verosimilitud, aproximaciones bayesianas y otras. El algoritmo de Esperanza-Maximización (EM) es una herramienta habitual iterativa para la estimación de máxima verosimilitud de las distribuciones mixtas (McLachlan y Basford, 1998). La idea es introducir una variable indicadora multinomial que identifica la pertenencia a un clúster de cada observación del conjunto de datos. Esto representa una aproximación conveniente para la obtención de los parámetros de las mixturas cuando no existe una solución analítica (ver Wilks [2006] para una descripción del algoritmo EM en un contexto general).

1.1. Las redes de inmisión

La calidad del aire viene determinada por la presencia en la atmósfera de sustancias contaminantes, que pueden ser gases o aerosoles. La protección de la atmósfera y de la calidad del aire en cualquier territorio pasa por la prevención, vigilancia y reducción de los efectos nocivos de dichas sustancias contaminantes sobre la salud y el medio ambiente en su conjunto. Para ello, las normativas vigentes en materia de calidad del aire establecen unos objetivos de calidad del aire o niveles de concentración de contaminantes en la atmósfera que no deben sobrepasarse. También ha de cumplirse con el requisito imprescindible de informar a la población y a las organizaciones interesadas.

En España, la Ley 34/2007, de 15 de noviembre, de calidad del aire y protección de la atmósfera, define la evaluación de la calidad del aire como el resultado de aplicar cualquier método que permita medir, calcular, predecir o estimar las emisiones, los niveles o los efectos de la contaminación atmosférica.

Conforme al Real Decreto 102/2011, de 28 de enero, la evaluación de la calidad del aire ambiente se realizará, dependiendo del nivel de los contaminantes con respecto a los umbrales/valores objetivo, mediante mediciones fijas, técnicas de modelización, campañas de mediciones representativas, mediciones indicativas o investigacio-

nes, o una combinación de todos o algunos de estos métodos. Por tanto, los diferentes métodos de evaluación pueden ser los siguientes:

- Mediciones fijas: mediciones efectuadas en emplazamientos fijos, bien de forma continua, bien mediante un muestreo aleatorio, con el propósito de determinar los niveles de conformidad con los objetivos de calidad de los datos establecidos por la legislación (incertidumbre, recogida mínima de datos y cobertura mínima temporal).
- Mediciones indicativas: mediciones cuyos objetivos de calidad de los datos en cuanto a cobertura temporal mínima son menos estrictos que los exigidos para las mediciones fijas (esto es, se efectúan con una menor frecuencia), pero satisfacen todos los demás objetivos de calidad de los datos establecidos por la legislación.
- Modelizaciones: herramientas matemáticas que simulan el comportamiento de la atmósfera para determinar los niveles de un determinado contaminante en ella.

La red de monitorización de calidad del aire de Andalucía incluye 89 estaciones de inmisión, siendo monitorizados simultánemante los contaminantes CO, NO_2, O_3, PM_{10} y SO_2 en, aproximadamente, unas 43 estaciones fijas de medida. Esta red es gestionada por la Agencia de Medio Ambiente y Agua de Andalucía, agencia pública empresarial adscrita a la administración ambiental de la Junta de Andalucía (en la fecha de redacción de esta memoria, denominada como Consejería de Medio Ambiente y Ordenación del Territorio).

1.1.1. Marco normativo

La normativa europea sobre calidad del aire actualmente en vigor (junio 2014) viene representada por las siguientes Directivas:

- Directiva 2008/50/CE del Parlamento Europeo y del Consejo, de 21 de mayo de 2008, relativa a la calidad del aire ambiente y a una atmósfera más limpia en Europa, transpuesta en España mediante el Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire.
- Directiva 2004/107/CE del Parlamento Europeo y del Consejo, de 15 de diciembre de 2004, relativa al arsénico, el cadmio, el mercurio, el níquel y los hidrocarburos aromáticos policíclicos en el aire ambiente, transpuesta en España mediante el Real Decreto 812/2007, de 22 de junio, sobre evaluación y gestión de la calidad del aire ambiente en relación con el arsénico, el cadmio, el mercurio, el níquel y los hidrocarburos aromáticos policíclicos, norma a su vez derogada por el Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire.

La Ley 34/2007, de 15 de noviembre, de calidad del aire y protección de la atmósfera, actualiza la base legal para los desarrollos relacionados con la evaluación y la gestión de la calidad del aire en España, y tiene como fin último el de alcanzar unos niveles óptimos de calidad del aire para evitar, prevenir o reducir riesgos o efectos negativos sobre la salud humana, el medio ambiente y demás bienes de cualquier naturaleza.

1.2. Motivación

El diseño de una red de monitorización de la calidad del aire básicamente conlleva determinar el número de estaciones y su emplazamiento, clase y número de contaminantes a monitorizar, sin dejar de considerar sus objetivos, costes y recursos disponibles. En la mayoría de los casos, las redes de monitorización en áreas metropolitanas son diseñadas para medir contaminantes de importancia sanitaria, como el CO, NO_2, O_3, PM_{10} y SO_2 (Chang y Tseng, 1999). Si no se reevalúa la representatividad de estos contaminantes para la detección de nuevas fuentes o niveles de contaminación, pueden llegar a no satisfacer la demanda informativa que la sociedad requiere al respecto. En ocasiones, algunos de estos contaminantes son monitorizados en estaciones vecinas, lo que conlleva una duplicidad de la información obtenida o la detección de similares niveles de contaminación

(la redundancia en el equipamiento fue introducida de forma analítica por Pires et al., 2008). Además, un problema inherente en los equipamientos de monitorización es que están sometidos a rigurosos programas de mantenimiento que, en caso de no cumplirse, conducen a que las estaciones no operen a un nivel satisfactorio.

El propósito de este trabajo es realizar una aproximación integrada para la optimización de la información suministrada por las redes de monitorización de calidad del aire y obtener criterios para su mejora. Esta mejora consiste esencialmente en replantear la monitorización de parámetros contaminantes en estaciones de inmisión, detectar posibles duplicidades, reclasificar los tipos de estaciones y, finalmente, ayudar al gestor de redes a efectuar consecuentes y progresivas modificaciones. Si se considerara que estas mejoras no son necesarias, la información obtenida mediante estos métodos estadísticos constituye una valiosa fuente para la caracterización y conocimiento de las redes de monitorización.

1.3. Información analizada

El presente estudio se concentró en el análisis de los contaminantes arriba citados durante el año 2012. Los datos observacionales provienen de 12 estaciones de monitorización, 10 de ellas situadas en el área metropolitana de Sevilla y 2 en la provincia, en un entorno rural, por lo que se dispuso de un diverso rango de concentraciones, localizaciones y ejemplos de atribuciones de fuentes.

Estas estaciones se han clasificado de acuerdo con el tipo de área donde se localizan (R-Rural, S-Suburbana, U-Urbana) y su fuente de emisión predominante (F-Fondo, I-Industrial, T-Tráfico). (Las principales características de las estaciones y contaminantes analizados en cada una de ellas se indican en la Tabla 1.1).

Dado que los datos monitorizados se obtienen a intervalos de tiempo diezminutales, dichas concentraciones fueron promediadas para obtener un valor único diario, asegurando de esta forma la independencia estadística de las observaciones. Estos valores medios se han calculado únicamente cuando se ha dispuesto de, al menos, el 80 % de todas las observaciones horarias durante el día (19 de 24 horas). A lo largo de este trabajo, todas las unidades de concentración se representan en $\mu g/m^3$.

Tabla 1.1: Contaminantes analizados y clasificación de las estaciones de monitorización de donde los datos fueron obtenidos (los emplazamientos son expresados en coordenadas X,Y ETRS89-UTM, zona 30). Los puntos representan contaminantes no monitorizados (11 casos).

Estación y abreviatura	Clasificación	Emplazamiento		Contaminantes estudiados				
Alcalá de Guadaíra (Alc)	U-F	248974	4136631	CO	NO_2	O_3	PM_{10}	SO_2
Aljarafe (Alj)	S-F	230473	4137017	·	NO_2	O_3	PM_{10}	SO_2
Bermejales (Ber)	U-F	236063	4137554	CO	NO_2	O_3	PM_{10}	SO_2
Centro (Cen)	U-F	235156	4142125	CO	NO_2	O_3	·	SO_2
Cobre Las Cruces (Cob)	R-I	231798	4160779	CO	NO_2	O_3	PM_{10}	SO_2
Dos Hermanas (Dos)	U-F	241677	4130413	CO	NO_2	O_3	·	SO_2
Príncipes (Pri)	U-F	233863	4140741	CO	NO_2	·	PM_{10}	SO_2
Ranilla (Ran)	U-T	237965	4141611	CO	NO_2	·	·	SO_2
San Jerónimo (Saj)	S-I	236286	4146731	·	NO_2	O_3	·	·
Santa Clara (Sac)	S-F	238720	4143149	CO	NO_2	O_3	PM_{10}	·
Sierra Norte (Sie)	R-F	265817	4208544	·	NO_2	O_3	PM_{10}	SO_2
Torneo (Tor)	U-T	234151	4142873	CO	NO_2	O_3	PM_{10}	SO_2

La red de monitorización estudiada se encuentra sujeta a un exhaustivo programa de mantenimiento que asegura la obtención de valores correctos. Dichos valores son validados por la administración ambiental autonómica previo a cualquier uso.

1.4. Orientación y estructura del trabajo

Se ha realizado la aproximación objeto de este trabajo mediante funciones específicamente diseñadas para la ocasión (algoritmo EM) o ya implementadas (ACJ, BA, ACP) a través del software libre R (R Core Team, 2013). La aplicación metodológica es secuencial y responde a los siguientes objetivos: 1) obtener mediante cada modelo de mixtura asociado a cada contaminate (Tabla 1.1) información de su nivel de impacto (μ_m) y variabilidad (σ_m) en las áreas de influencia de cada estación, además de evaluar la atribución de fuentes de contaminación; 2) identificar el agrupamiento de las estaciones de inmisión en base a μ_m y cv_m empleando el ACJ; 3) estimar mediante la técnica de imputación basada en BA (Breiman, 2001) los valores de μ_m y cv_m para contaminantes no monitorizados; y 4) estudiar la reclasificación de las estaciones a través del ACP, añadiendo la nueva información obtenida mediante la imputación a la ya conocida por modelización mediante mixturas.

El procedimiento descrito ha sido aplicado en la red de monitorización de calidad del aire de Sevilla, tanto por el mayor número de estaciones que la componen como por su centralización. La provincia de Sevilla se localiza en el sur de España y cubre una superficie de 14 036 km^2, y durante el año de estudio tenía una población total de 1 935 364 habitantes (IECA, 2012). El área metroplitana de Sevilla es la principal aglomeración urbana de la región de Andalucía y tenía, para el mismo periodo, una población de 1 217 811 habitantes (SG, 2012).

El apartado metodológico en este trabajo comprende la descripción de las mixturas finitas, el algoritmo EM y las técnicas estadísticas complementarias utilizadas para la consecución del propósito descrito más arriba. En el Capítulo 2 se definen los tipos de mixturas empleadas para describir los datos experimentales analizados, así como el algoritmo EM estándar, utilizado para estimar la parametrización de estas mixturas. En este capítulo, además, se especifican algunos aspectos concretos de la codificación empleada de este algoritmo, como la definición del criterio de parada empleado o la tolerancia permitida. También se aborda la obtención de la mejor modelización utilizando los criterios de información. Los algoritmos empleados para la estimación de errores de los estimadores EM de las mixturas se tratan al final del capítulo: el método *SEM* (Supplemented EM algorithm) y el replicativo *bootstrap*. La implementación computacional en R del algoritmo EM se recoge en el Anexo II, en donde se muestran todas las funciones específicas diseñadas y un ejemplo de su uso.

El Capítulo 3 muestra una breve definición de estos métodos complementarios empleados: el ACJ, la imputación mediante BA y el ACP. Estos han sido implementados en R utilizando libros ya disponibles en este entorno. Esta implementación se muestra en el Anexo III.

Los resultados obtenidos de la parámetrización de las mixturas se muestran en el Anexo I, y son resumidos y discutidos en el Capítulo 4, junto con los resultados de los métodos complementarios. Finalmente, en el Capítulo 5 se resume este trabajo y se obtienen sus principales conclusiones.

2

Modelos de mixturas y Algoritmo EM

El presente capítulo comienza con las definiciones básicas relacionadas con los modelos de mixturas finitas y la estimación de máxima verosimilitud (apartado 2.1). A continuación, se presenta el algoritmo EM como un método general de obtención de soluciones para las ecuaciones de verosimilitud en el caso de que no exista una analítica para ellas (apartado 2.2). Seguidamente, se expone el problema de los valores iniciales, un aspecto al que el algoritmo se muestra especialmente sensible. La selección del mejor modelo se trata en el apartado 2.4. Finalmente, el capítulo acaba con los dos métodos utilizados en el trabajo para aproximar el valor los errores de los estimadores EM: el método SEM y el replicativo bootstrap.

2.1. Modelos de mixturas finitas

Las distribuciones mixtas son utilizadas para la modelización de datos heterogéneos en multitud de situaciones experimentales, en donde aquellos pueden interpretarse como procedentes de dos o más subpoblaciones (componentes[1]). La obtención de estas componentes conduce a la estimación de los parámetros de la mixtura. Este problema de estimación tiene una larga historia y se remonta a Pearson (1894), quien trabajó con una mixtura de dos componentes con varianzas iguales usando el método de los momentos. Trabajos posteriores que utilizan esta aproximación son los de Charlier (1906), Charlier y Wicksell (1924), Cohen (1967), y Tan y Chang (1972). Más tarde, Rao (1948) y Hasselblad (1966, 1969) utilizaron la estimación de máxima verosimilitud en este contexto.

Un amplio rango de aplicaciones prácticas y un análisis estadístico detallado de las mixturas finitas, considerando diferentes métodos de estimación, fueron presentados por Everitt y Hand (1981), y Titterington et al. (1985). Descripciones más generales fueron publicadas por McLachlan y Basford (1988), McLachlan y Jones (1988), McLachlan y Krishnan (1997), y McLachlan y Peel (2000).

Una buena revisión histórica de estos modelos puede encontrarse en Holgersson y Jorner (1978), Redner y Homer (1984), y Everitt (1996). Algunas aplicaciones en un contexto médico fueron presentadas por Schlattmann (2009) y Frühwirth-Schnatter (2010). El trabajo reciente de Mengerse et al. (2011) muestra la relevancia de los modelos mixtos considerando un esquema bayesiano.

Para el desarrollo conceptual del algoritmo EM, que se verá más adelante, es conveniente proporcionar una formulación paramétrica para la representación del modelo, y se adoptará aquí, en general, la notación de McLachlan y Peel (2000). En lo sucesivo, mediante $Y = (Y_1, Y_2, \ldots, Y_n)$, se denotará a una muestra aleatoria de tamaño n, donde Y_j es un vector aleatorio q-dimensional con función de densidad de probabilidad $f(y_j)$ en $\mathbb{R}^q$. Así, $y = (y_1, y_2, \ldots, y_n)$ representa a una muestra observada o realización de Y, donde y_j constituye un valor observado del vector aleatorio Y_j.

[1] La asociación de cada componente de la mixtura a un clúster es objeto de discusión. En la portada de este trabajo, se ha querido representar gráficamente la posibilidad de que una mixtura de 4 componentes aparentes esté descrita por 5 subpoblaciones o procesos generadores.

Definición 2.1 La distribución de una variable aleatoria Y_j cuya función de densidad se escribe

$$f(y_j|\Psi) = \sum_{i=1}^{g} \pi_i \; f_i(y_j|\theta_i), \quad y_j \in \mathbb{R}^q, \tag{2.1}$$

se denomina *distribución de mixtura finita* de g componentes, con un vector de parámetros del modelo

$$\Psi = (\pi_1, \ldots, \pi_g, \theta_1, \ldots, \theta_g).$$

Así, $f_i(y_j|\theta_i)_{i=1,\ldots,g}$ denotan las *densidades de las componentes* de la mixtura con parámetros θ_i, y $\pi_1, \ldots, \pi_g$, las *proporciones o pesos*. Mediante la notación $f_i(\cdot|\theta_i)$, se asume que estas densidades pueden pertenecer a diferentes familias paramétricas, agrupándose cada uno de sus vectores parámetricos en θ_i.

Las proporciones de la mixtura representan las probabilidades de que la realización y_j de la variable aleatoria haya sido generada por las g diferentes densidades y, como probabilidades que son, están sujetas a las restricciones

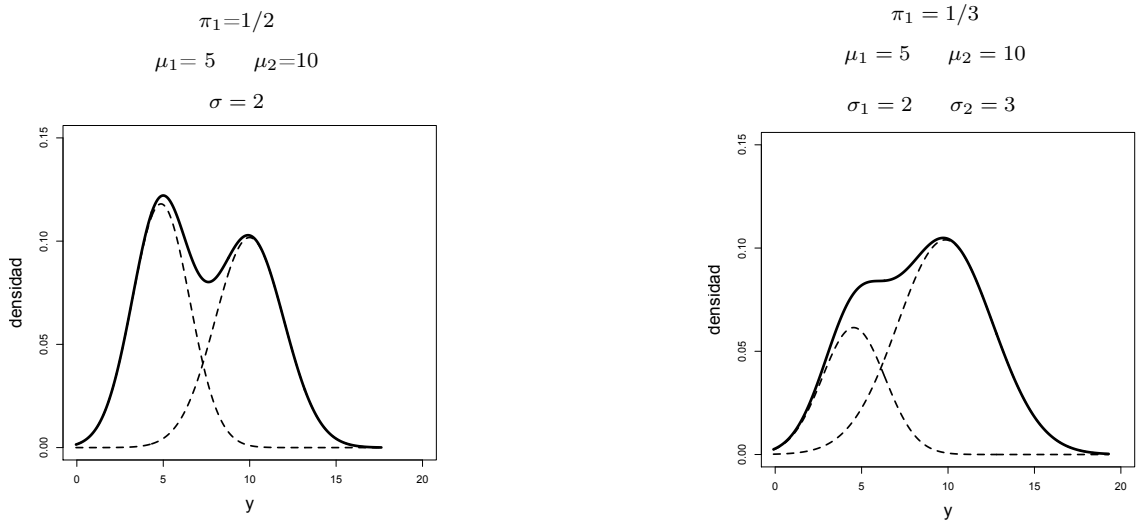
$$0 \leqslant \pi_i \leqslant 1 \qquad i = 1, \ldots, g \tag{2.2}$$

y

$$\sum_{i=1}^{g} \pi_i = 1, \tag{2.3}$$

por lo que uno de los pesos resulta redundante. La Figura 2.1 muestra un ejemplo de mixturas de distribuciones gaussianas con dos componentes y diferente parametrización, creadas a partir de muestras sintéticas generadas mediante una misma semilla aleatoria.

Figura 2.1: Las lineas discontinuas muestran la densidad de cada componente. A la izquierda, mixtura homocedástica.



Observación 2.1 Para el desarrollo experimental de este trabajo, las densidades de las componentes han sido del tipo gaussianas univariantes heterocedásticas, por lo que en (2.1) éstas podrían haberse representado mediante $f(y_j|\theta_i)$, con $\theta_i = (\mu_i, \sigma_i)$, o bien $\phi(y_j|\mu_i, \sigma_i)$. Para mixturas cuyas componentes pertenecen a otras familias de densidades, puede consultarse Simar (1976) y Moharir (1992) para mixturas de distribuciones Poisson; Falls (1970), para mixturas Weibull; o Blischke (1962) y Medgyessy (1961), para mixturas binomiales.

Como se decía, dado que las distribuciones empleadas en este trabajo han sido las gaussianas, procede a continuación indicar su función de densidad.

Definición 2.2 Se dice que una variable aleatoria Y sigue una distribución normal o gaussiana si su función de densidad puede escribirse como

$$f(y|\theta) = \phi(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \qquad -\infty < y < \infty \tag{2.4}$$

con $-\infty < \mu < \infty$, $\sigma^2 > 0$ y $\theta = (\mu, \sigma^2)$ los parámetros de la distribución.

2.1.1. Identificabilidad

Para un conjunto de observaciones $y_1, \ldots, y_n$ se persigue su ajuste a una distribución mixta mediante la estimación de todos los parámetros Ψ, como se definió en (2.1). La estimación de Ψ en función de las observaciones y_j sólo tiene sentido si Ψ es identificable, lo que hace referencia a la existencia de un única caracterización para un modelo de mixtura. En general, una familia paramétrica de densidades $f(y_j|\Psi)$ es identificable si valores diferentes del parámetro Ψ determinan miembros distintos de la familia de densidades. Esto es

$$f(y_j|\Psi) = f(y_j|\Psi^*), \tag{2.5}$$

Si y solo si

$$\Psi = \Psi^*.$$

En las mixturas de distribuciones la identificabilidad es algo diferente. Si $f(y_j|\Psi)$ posee dos componentes con densidades $f_i(y_j|\theta_i)$ y $f_h(y_j|\theta_h)$ que pertenecen a la misma familia paramétrica, entonces (2.5) solo es cierta si los índices de las componentes i y h se intercambian en Ψ. Aunque esta clase de mixturas pueda ser identificable, Ψ no lo es. Por tanto, la identificabilidad en la caso de mixturas finitas se interpreta como sigue:

Definición 2.3 Sean

$$f(y_j|\Psi) = \sum_{i=1}^{g} \pi_i \ f_i(y_j|\theta_i) \qquad \text{y} \qquad f(y_j|\Psi^*) = \sum_{i=1}^{g^*} \pi_i^* \ f_i(y_j|\theta_i^*)$$

dos miembros cualesquiera de una familia paramétrica de mixturas. Esta clase de mixturas finitas se dice que es identificable para Ψ si

$$f(y_j|\Psi) \equiv f(y_j|\Psi^*)$$

Si y solo si $g = g^*$ y las etiquetas de las componentes pueden permutarse tal que

$$\pi_i = \pi_i^* \qquad \text{y} \qquad f(y_j|\theta_i) \equiv f(y_j|\theta_i^*) \qquad i = 1, \ldots, g.$$

Con esta definición, las mixturas de distribuciones normales son identificables si pueden permutarse los índices de las componentes. Para solventar este intercambio, pueden imponerse restricciones a la solución, como, por ejemplo, $\mu_1 < \mu_1 < \ldots < \mu_k$ (Aitkin y Rubin, 1985). Por tanto, en la práctica, todos los parámetros pueden ser determinados con exactitud. Para una descripción detallada del concepto de identificabilidad, puede consultarse Titterington et al. (1985), Teicher (1961, 1963), Frühwirth-Schnatter (2010), y Yakowitz y Spragins (1968). La pérdida de identificabilidad no supone un inconveniente en el ajuste mediante mixturas de distribuciones

por el método de la máxima verosimilitud, como es el caso en el empleo del algoritmo EM (McLachlan y Peel, 2000).

2.1.2. Mixtura de 3 componentes gaussianas

Como podrá comprobarse en el Capítulo 4 (Tabla 4.1), las mixturas finitas obtenidas para la modelización de las distribuciones de datos experimentales no presentaron en ningún caso más de tres componentes ($g \leq 3$). Por ello cobra sentido el presente apartado, que contiene la definición de este tipo de mixtura y donde se expone la utilidad del cálculo de sus momentos.

Definición 2.4 Una distribución cuya función de densidad es

$$g(y|\Psi) = \pi_1 \ \phi(y|\mu_1, \sigma_1^2) + \pi_2 \ \phi(y|\mu_2, \sigma_2^2) + \pi_3 \ \phi(y|\mu_3, \sigma_3^2) \tag{2.6}$$

se dice que sigue una mixtura de tres componentes gaussianas, donde $\phi(\cdot)$ es la función de densidad gaussiana, como se definió en (2.4) y

$$\Psi = (\pi_1, \ \pi_2, \ \mu_1, \ \mu_2, \ \mu_3, \ \sigma_1^2, \ \sigma_2^2, \ \sigma_3^3).$$

Para asegurar la identificabilidad de Ψ, se asume que las medias de las componentes se encuentran en orden ascendente,

$$\mu_1 \ < \mu_2 \ < \mu_3.$$

En la Figura 2.2 se muestran algunos ejemplos de mixturas de distribuciones con 3 componentes gaussianas con diferentes parametrizaciones. Como en la Figura 2.1, para su representación se crearon muestras artificiales de 3 componentes, a partir de una misma semilla, estimándose posteriormente los parámetros mediante el algoritmo EM especialmente diseñado para este trabajo.

Observación 2.2 El solapamiento de las componentes en mixturas gaussianas puede determinarse cuantitativamente a través de la diferencia de cada una de las medias de las componentes. No obstante, es necesario suponer una condición de homocedasticidad entre las componentes ($\sigma = \sigma_i \, , i = 1, \ldots, g$). Dado que no es el caso en este trabajo, su desarrollo se omite, aunque se remite a McLachlan y Peel (2000, cap.2) para su consulta.

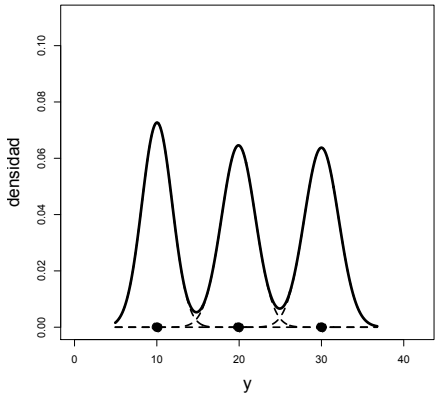
Lema 2.1 Suponiendo que Y es una variable aleatoria que sigue una distribución mixta de tres componentes gaussianas definida como en (2.6), y que los momentos de primer y segundo orden de las mismas existan, entonces, la media μ_m y varianza σ_m^2 de la mixtura son:

$$\mu_m = \pi_1\mu_1 + \pi_2\mu_2 + \pi_3\mu_3.$$
$$\sigma_m^2 = \pi_1(\sigma_1^2 + \mu_1^2) + \pi_2(\sigma_2^2 + \mu_2^2) + \pi_3(\sigma_3^2 + \mu_3^2) - \mu_m^2.$$

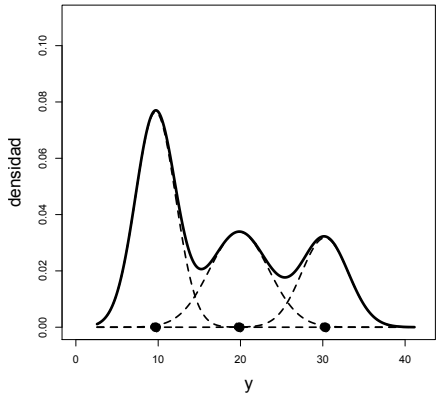
Demostración 2.1 Utilizando el operador E como esperanza, la media y la varianza de la mixtura se calculan

Figura 2.2: Ejemplos de mixturas con 3 componentes gaussianas, homocedásticas en la fila superior. Los puntos en el eje de abcisas representan la media de cada componente.

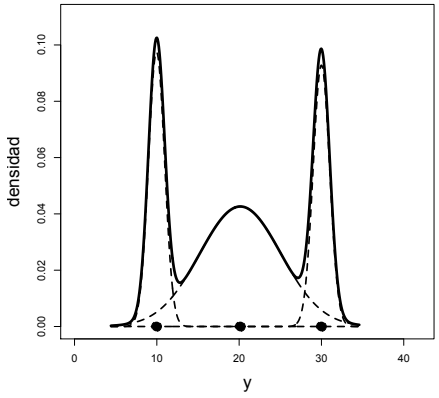
$\pi_1 = 1/3 \quad \pi_2 = 1/3$
μ_1= 10 μ_2=20 μ_3=30
$\sigma = 3$



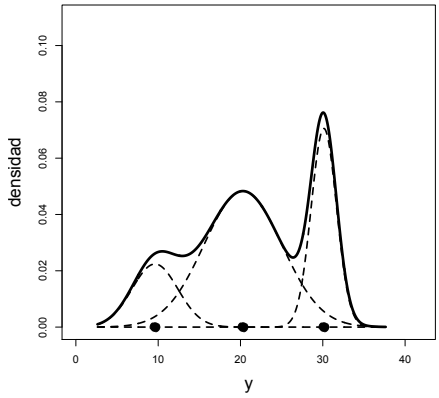
$\pi_1 = 1/2 \quad \pi_2 = 1/4$
μ_1= 10 μ_2=20 μ_3=30
$\sigma = 3$



$\pi_1 = 1/4 \quad \pi_2 = 1/2$
μ_1= 10 μ_2=20 μ_3=30
σ_1= 1 σ_2=5 σ_3=1



$\pi_1 = 1/7 \quad \pi_2 = 4/7$
μ_1= 10 μ_2=20 μ_3=30
σ_1= 3 σ_2=5 σ_3=1,5



$$\mu_m = \mathsf{E}[Y|\Psi] =$$

$$= \int_{-\infty}^{\infty} y\ g(y|\Psi)\ dy$$

$$= \int_{-\infty}^{\infty} y\ \pi_1\ f(y|\theta_1)\ dy + \int_{-\infty}^{\infty} y\ \pi_2\ f(y|\theta_2)\ dy + \int_{-\infty}^{\infty} y\ \pi_3\ f(y|\theta_3)\ dy$$

$$= \pi_1\mu_1 + \pi_2\mu_2 + \pi_3\mu_3.$$

$$\sigma_m^2 = \mathsf{E}[Y^2|\Psi] - \mathsf{E}[Y|\Psi]^2 =$$

$$= \int_{-\infty}^{\infty} y^2\ g(y|\Psi)\ dy\ - \mu_m^2$$

$$= \pi_1 \int_{-\infty}^{\infty} y^2\ f(y|\theta_1)\ dy + \pi_2 \int_{-\infty}^{\infty} y^2\ f(y|\theta_2)\ dy + \pi_3 \int_{-\infty}^{\infty} y^2\ f(y|\theta_3)\ dy - \mu_m^2$$

$$= \pi_1(\sigma_1^2 + \mu_1^2) + \pi_2(\sigma_2^2 + \mu_2^2) + \pi_3(\sigma_3^2 + \mu_3^2) - \mu_m^2.$$

Expresiones que pueden generalizarse mediante:

$$\mu_m = \sum_{i=1}^{g} \pi_i \mu_i \qquad \sigma_m^2 = \sum_{i=1}^{g} \pi_i(\mu_i^2 + \sigma_i^2) - \mu_m^2. \tag{2.7}$$

La ventaja de la utilización de estos momentos reside en que permite resumir la parametrización completa de una mixtura mediante dos simples números, μ_m y σ_m^2. Por simplicidad, para describir la variabilidad de la mixtura, se utilizará $\sigma_m = \sqrt{\sigma_m^2}$.

2.1.3. Estimación mediante máxima verosimilitud

Existen numerosos métodos de estimación puntuales, entre los que se incluyen el método de los momentos, procedimientos gráficos, bayesiana, mínimo cuadrática, mínimo χ^2 o máxima verosimilitud. La elección del método más adecuado para la estimación de los parámetros en las mixturas ha sido motivo de discusión/controversia y una respuesta parcial puede encontrarse en Tan y Chang (1972). Estos autores realizaron una comparación entre el método de los momentos y el de máxima verosimilitud, demostrando que el segundo es superior. Holgerson y Jorner (1978) compararon igualmente varios métodos de estimación, llegando igualmente a semejante conclusión.

Finalmente, Day (1969) mencionó las ventajas de estimación mediante máxima verosimilitud sobre el mínimo χ^2 y los estimadores bayesianos. Por tanto, la estimación mediante máxima verosimilitud es la utilizada en este trabajo, lo que conlleva la maximización de la función de verosimilitud, o equivalentemente, la maximización de la función de log-verosimilitud, la cual es descrita, por ejemplo, en Little y Rubin (2002).

Definición 2.5 Sea $y = (y_1, y_2, \ldots, y_n)$ observaciones independientes de una variable aleatoria Y con función de densidad $f(y|\theta)$, donde θ es el vector de parámetros desconocidos que queremos estimar; entonces, la función de densidad conjunta de y se escribe

$$f(y|\theta) = \prod_{j=1}^{n} f(y_j|\theta) = L(\theta|y) \tag{2.8}$$

donde $L(\theta|y)$ representa la función de verosimilitud y se considera una función de θ.

Observación 2.3 Para resaltar el hecho de que la función de verosimilitud es una función de θ, se denota también como f_θ. El proceso de estimación está basado en la asignación de un valor al parámetro desconocido θ que caracteriza una población, y que es observada a través de la muestra aleatoria Y. La idea que subyace en el método de máxima verosimilitud es dar como estimación del parámetro aquel valor, de entre los posibles, que haga máxima la probabilidad de la muestra observada. Así, se considera que es preciso ajustar el valor de θ, permaneciendo fijos los valores de la muestra.

Como el máximo de una función y el de su logaritmo se alcanzan en el mismo valor de θ, habitualmente resulta más simple emplear su transformación logarítmica:

$$\ell(\theta|y) = \log L(\theta|y) = \log \prod_{j=1}^{n} f(y_j|\theta) = \sum_{j=1}^{n} \log f(y_j|\theta)$$

Definición 2.6 Un estimador de máxima verosimilitud $\hat{\theta}$ de θ es un valor que maximiza $L(\theta|y)$, es decir,

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L(\theta).$$

donde Θ representa el espacio paramétrico.

Esta definición permite la posibilidad de obtener más de un estimador de máxima verosimilitud, como puede ocurrir en aproximaciones experimentales donde se detectan múltiples máximos. No obstante, para mucho modelos destacables "el estimador de máxima verosimilitud es único, y más aún, la función de verosimilitud es diferenciable y acotada superiormente" (Little y Rubin, 2002, p. 81). En tales casos, puede encontrarse una solución mediante la resolución de las correspondientes ecuaciones normales.

Definición 2.7 Las ecuaciones normales o de verosimilitud vienen dadas por

$$S(y|\theta) = \frac{\partial \ell(\theta|y)}{\partial \theta_j} = 0, \qquad j = 1, \ldots, k.$$

en el supuesto de que $\theta = (\theta_1, \theta_2, \ldots, \theta_k)$ sea un parámetro k dimensional.

Definición 2.8 Sea

$$I(\theta|y) = -\frac{\partial^2}{\partial\theta\partial\theta^T}\ \ell(\theta|y)$$

las derivadas segundas parciales negativas de la función de log-verosimilitud con respecto a θ, donde el θ^T denota el vector transpuesto de θ. Entonces, $I(\theta|y)$ se denomina la *matriz de información observada*. La matriz de información *esperada* viene dada, bajo condiciones de regularidad, por

$$\mathcal{I}(\theta|y) = \mathsf{E}[S(y|\theta)\, S^T(y|\theta)] = \mathsf{E}[I(\theta|y)].$$

Observación 2.4 Esta definición es aplicable en el caso de la mixtura sustituyendo θ por Ψ.

Ejemplo 2.1 Sea $(Y_1, Y_2, \ldots, Y_n)$ una muestra aleatoria simple de una población $N(\mu, \sigma^2)$, como se definió en (2.4), siendo $\theta = (\mu, \sigma^2)$ dos parámetros desconocidos. Entonces, las funciones de verosimilitud y log-verosimilitud son

$$L(\theta|y) = f_\theta(y_1, y_2, \ldots, y_n) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{n}(y_j - \mu)^2\right\}.$$

$$\begin{aligned}\ell(\theta|y) &= -\frac{n}{2}\ \log\ \sigma^2 + \log\left(\frac{1}{\sqrt{2\pi}}\right)^n - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(y_j - \mu)^2 \\ &= -\frac{n}{2}\ \log\ \sigma^2 - \frac{n}{2}\ \log(2\pi) - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(y_j - \mu)^2.\end{aligned}$$

La derivada de $\ell(\theta|y)$ respecto a cada uno de los parámetros μ y σ^2 permite obtener las siguientes ecuaciones de verosimilitud:

$$\frac{\partial}{\partial\mu}\ \ell(\theta|y) = \frac{1}{2\sigma^2}\sum_{t=1}^{n} 2\ (y_i - \mu) = \frac{1}{\sigma^2}\sum_{j=1}^{n}(y_i - \mu) = 0,$$

$$\frac{\partial}{\partial\sigma^2}\ \ell(\theta|y) = -\frac{n}{2}\frac{1}{\sigma^2} + \frac{2\ \sum_{j=1}^{n}(y_j - \mu)^2}{4(\sigma^2)^2} = -\frac{1}{2\sigma^2}\left(-n + \frac{\sum_{j=1}^{n}(y_j - \mu)^2}{\sigma^2}\right) = 0,$$

representando un sistema de dos ecuaciones con dos incógnitas, que tiene como soluciones la media y la varianza muestral:

$$\hat{\mu} = \bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j \qquad\qquad \hat{\sigma}^2 = s^2 = \frac{1}{n}\sum_{j=1}^{n}(y_j - \bar{y})^2$$

Para la comprobación de que se trata de un máximo, se realizan las segundas derivadas en $\ell(\theta|y)$:

$$\frac{\partial^2}{\partial\theta\partial\theta^T}\ \ell(\theta|y) = -\begin{pmatrix} \dfrac{n}{\sigma^2} & \dfrac{1}{\sigma^4}\displaystyle\sum_{j=1}^{n}(y_j-\mu) \\ \dfrac{1}{\sigma^4}\displaystyle\sum_{i=1}^{n}(y_i-\mu) & \dfrac{1}{\sigma^6}\displaystyle\sum_{j=1}^{n}(y_j-\mu)^2 - \dfrac{n}{2\sigma^4} \end{pmatrix} = I(\theta|y),$$

matriz que es definida negativa cuando $\hat{\mu} = \bar{y}$ y $\hat{\sigma}^2 = s^2$.

La matriz de información esperada, entonces:

$$\mathcal{I}(\theta|y) = \mathsf{E}[I(\theta|y)] = \begin{pmatrix} \dfrac{n}{\sigma^2} & 0 \\ 0 & \dfrac{n\sigma^2}{\sigma^6} - \dfrac{n}{2\sigma^4} \end{pmatrix} = \begin{pmatrix} \dfrac{n}{\sigma^2} & 0 \\ 0 & \dfrac{n}{2\sigma^4} \end{pmatrix},$$

que tiene por inversa:

$$\mathcal{I}(\theta|y)^{-1} = \begin{pmatrix} \dfrac{\sigma^2}{n} & 0 \\ 0 & \dfrac{2\sigma^4}{n} \end{pmatrix}.$$

Definición 2.9 Sea $y = (y_1, y_2, \ldots, y_n)$ observaciones independientes de una variable aleatoria Y con función de densidad $f(y|\Psi)$, donde Ψ es el vector de parámetros desconocidos que queremos estimar, entonces,

$$L(\Psi|y) = \prod_{j=1}^{n}\ f(y_j|\Psi) = \prod_{j=1}^{n}\ \sum_{i=1}^{g}\pi_i\ f_i(y_j|\theta_i), \tag{2.9}$$

recibe el nombre de *función de verosimilitud de la mixtura*, que, tomando logaritmos en $L(\Psi|y)$, conduce a su función de *log-verosimilitud*:

$$\ell(\Psi|y) = \log\ L(\Psi|y) = \log\ \prod_{j=1}^{n}\left\{\sum_{i=1}^{g}\pi_i\ f_i(y_j|\theta_i)\right\} = \sum_{j=1}^{n}\log\left\{\sum_{i=1}^{g}\pi_i\ f_i(y_j|\theta_i)\right\}, \tag{2.10}$$

cuya correspondiente ecuación de verosimilitud es

$$\frac{\partial}{\partial\Psi}\sum_{j=1}^{n}\log\left\{\sum_{i=1}^{g}\pi_i\ f_i(y_j|\theta_i)\right\} = 0. \tag{2.11}$$

Esta expresión, debido a la presencia del logaritmo dentro de una suma, es de difícil resolución y requiere, en el mejor de los casos, procedimientos iterativos para su resolución. En el caso de que $f(\cdot)$ presente una forma

compleja, puede no llegar a tener una solución analítica (McLachlan y Basford, 1988, cap.2).

Entre estos procedimientos, los más ampliamente utilizados son el método de Newton-Rapson (NR) y *scoring* de Fisher. El algoritmo EM se ha convertido en otra técnica estándar para el cálculo de los EMV. Este algoritmo representa en sí mismo un esquema de trabajo para la obtención de estimaciones en problemas de datos faltantes. La idea que en él subyace es resolver problemas de datos incompletos de cierta complejidad abordando de forma repetida una situación de datos completos de resolución más asequible. Para ello es necesario considerar que la población en estudio, aunque sin datos faltantes, los posee. Así, se asigna a cada dato observado una etiqueta que indique su pertenencia a una u otra subpoblación en la muestra de partida, asumiendo que estas existen. Así, entonces, la asignación de un conjunto de datos a una u otra componente de la mixtura conduce de forma natural al agrupamiento de estos o a la formación de *clústers*.

El valor de estas etiquetas o indicadores de pertenencia a un clúster es, a priori, desconocido. Así, bajo una distribución mixta, la estimación de su valor puede considerarse como un problema de datos faltantes, y el algoritmo EM puede utilizarse. A diferencia del método scoring de Fisher, EM no requiere del cálculo de una matriz Hessiana en cada iteración, en particular la inversa de la matriz de información, lo que supone una ventaja si esta es compleja de calcular. Esta matriz Hessiana puede ser aproximada numéricamente mediante simulación empleando métodos de Montecarlo, aunque el coste computacional puede ser elevado si se requieren numerosas iteraciones para su obtención.

2.2. El algoritmo EM

2.2.1. Introducción al algoritmo

El algoritmo EM fue descrito y presentado por Dempster et al. (1977) para la obtención de un máximo de la función de verosimilitud cuando su cálculo no es posible. Además de problemas con datos censurados o perdidos, como se adelantó, el algoritmo puede utilizarse también en otras situaciones en donde no se considera aparentemente la ausencia de observaciones, como es el caso de las mixturas finitas. Bajo este último enfoque, se plantea la necesidad de formular el conjunto de datos observados como un caso de datos incompletos. El extenso ámbito de aplicación de este algoritmo en diferentes campos de aplicación lo ha hecho ciertamente popular. En 1992, Meng y Pedlow publicaron una relación bibliográfica con más de 1 000 artículos relacionados con el algoritmo EM. McLachlan y Khirshan (1997) estimaron al menos 1 700 publicaciones.

La idea básica del algoritmo fue anterior a Dempster et al. (1977). La primera referencia en la literatura sobre un algoritmo tipo EM pertenece a Newcomb (1886), quien consideró la estimación de los parámetros de una mixtura gaussiana de dos componentes. Este trabajo fue seguido por muchos otros, como McKendrick (1926), quien presentó una aplicación en un contexto médico, y Healy y Westmacott (1956), quienes propusieron un ejemplo del algoritmo EM en un diseño por bloques completamente aleatorizado. Baum et al. (1970) utilizaron este algoritmo conjuntamente con modelos de Markov, y Orchad y Woodbuty (1972) trabajaron en un algoritmo similar denominado "principio de información perdida". Un resumen estructurado de la historia del algoritmo EM puede encontrarse en McLachlan y Krishnan (1997), o en Redner y Hoomer (1984). Otro resumen interesante al respecto puede encontrarse en McLachlan y Jones (1988), y en Little y Rubin (2002).

El algoritmo EM posee ciertas ventajas atractivas comparado con otros algoritmos iterativos, como el algoritmo de NR o método scoring de Fisher. Por ejemplo, la economía de almacenamiento, la facilidad de implementación y la estabilidad numérica. Más aún, en la mayoría de situaciones prácticas, el algoritmo EM converge en condiciones de regularidad a un máximo local, concepto que se desarrolla al final de esta sección.

No obstante, la utilización de este algoritmo en muchas situaciones estadísticas revela su limitación. Por ello, se han desarrollado numerosas modificaciones y extensiones. Especialmente, el hecho de que en ciertas situaciones el algoritmo converge muy lentamente ha llevado al desarrollo de modificaciones de aceleración del mismo. Por ejemplo, Redner y Homer (1984) recomendaron utilizar el algoritmo EM de forma conjunta con un algoritmo de tipo Newton, donde las buenas propiedades de convergencia del algoritmo EM se sumarán

a la rápida convergencia local del método de Newton. Otro algoritmo híbrido, denominado EM/GN, fue introducido por Aitkin y Aitkin (1996), donde el algoritmo EM es combinado con el algoritmo Gauss-Newton (GN). Igualmente, Du (2002) propuso la combinación del algoritmo EM con el NR. Para más detalles sobre algoritmos EM modificados y algoritmos híbridos, se recomienda dirigirse a McLachlan y Krishnan (1997), y Little y Rubin (2002).

Datos incompletos y completos

Definición 2.10 Sea $y = (y_1, y_2, \ldots, y_n)$ una muestra observada de tamaño n, a la que denominaremos vector de datos *incompleto*, correspondiente a una realización de Y, con función de densidad $f(y|\Psi)$, donde Ψ es el vector de parámetros que se desea estimar. Sea igualmente una variable $Z = (Z_1, Z_2, \ldots, Z_n)$, que denominaremos *latente*, que representará a los datos no observados, y cuya realización es $z = (z_1, z_2, \ldots, z_n)$. El vector aleatorio $X = (Y, Z)$ recibe el nombre de vector de datos *completos* y sus realizaciones serán $x_1 = (y_1, z_1),\ x_2 = (y_2, z_2), \ldots,\ x_n = (y_n, z_n)$, de tal forma que a una realización y_j le corresponde siempre otra z_j.

Observación 2.5 La presencia de estas nuevas variables asociadas a unas observaciones, ahora consideradas incompletas, se entiende como un *aumento* de los datos, denominación que surge naturalmente en aplicaciones que presentan datos faltantes (Tanner, 1987). Las funciones de verosimilitud y log-verosimilitud de la mixtura dadas en (2.9) y (2.10) son denominadas entonces, cada una de ellas, *incompletas*.

En este contexto, Z_j representa a una variable indicadora binaria g-dimensional cuyo elemento i-ésimo, Z_{ij}, indica la pertenencia de la observación y_j a la componente i-ésima de la mixtura ($i = 1, \ldots, g\ ;\ j = 1, \ldots, n$). Es decir, $z_{ij} \in \{0, 1\}$ y:

$$Z_{ij} = \begin{cases} 1 & \text{si la observación } y_j \text{ proviene de la componente } i\text{-ésima} \\ 0 & c.c. \end{cases}$$

La representación matricial del vector de datos incompletos (y) y completos (x) es:

$$y' = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad x' = \begin{pmatrix} y' & z' \end{pmatrix} = \begin{pmatrix} y_1 & z_1 \\ y_2 & z_2 \\ \vdots & \vdots \\ y_n & z_n \end{pmatrix} = \begin{pmatrix} y_1 & z_{11} & \cdots & z_{1g} \\ y_2 & z_{21} & \cdots & z_{2g} \\ \vdots & \vdots & \ddots & \vdots \\ y_n & z_{n1} & \cdots & z_{ng} \end{pmatrix}.$$

Observación 2.6 Cada variable Z_j toma $(g-1)$ valores 0 y un único valor 1. Por su parte, las variables Z_{ij} son referidas en la literatura como $I_{\{Z_j=i\}}$, $\mathbf{1}_{\{Z_j=i\}}$ u otra representación adecuada de variable indicadora.

Dada la naturaleza categórica de las variables Z_{ij} al indicar la pertenencia ("*labelling*") de los puntos muestrales a una componente u otra de la mixtura, puede asumirse que Z_j sigue una distribución multinomial de solo una realización sobre g categorías con probabilidades $\pi = (\pi_1, \ldots, \pi_g)$, es decir, la función de probabilidad de Z_j será:

$$Z_1, Z_2, \ldots, Z_n \overset{iid}{\sim} \text{Mult}_g(1, \pi),$$

$$P(Z_j = z_j) = \begin{pmatrix} 1 \\ z_{j1}\ z_{j2}\ \cdots\ z_{jg} \end{pmatrix} \pi_1^{z_{j1}}\ \pi_2^{z_{j2}}\ \cdots\ \pi_g^{z_{jg}} = \prod_{i=1}^{g} \pi_i^{z_{ji}}, \tag{2.12}$$

verificándose de nuevo las restricciones (2.2) y (2.3), y además :

$$\sum_{i=1}^{g} z_{ij} = 1 \qquad \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} = n.$$

Observación 2.7 Titterington (1990) denomina a este modelo multinomial oculto.

Función de log-verosimilitud de los datos completos

Teniendo en cuenta la relación entre la densidad marginal y condicional de una observación, la función de densidad conjunta de una observación, que denominamos *completa*, es:

$$f(x_j) = f(y_j, z_j) = f(y_j|z_j)\ p(z_j). \tag{2.13}$$

Las variables $Z_1, \ldots, Z_n$, con $Z_j = \{Z_{j1}, \ldots, Z_{jg}\}$, están relacionadas con la observación muestral Y_j condicionalmente, siendo esta la única información de la que se dispone de la distribución de Z_j:

$$f_i(Y_j|Z_{ji} = 1) \sim f_i(y_j|\theta_i). \tag{2.14}$$

Desarrollando (2.13), la distribución conjunta de Y_j y todos los estados posibles de Z_j es:

$$\begin{aligned} f_i(Y_j, Z_j) &= f_i(Y_j = y_j, Z_{j1} = z_{j1}, \ldots, Z_{jg} = z_{jg}) = \\ &= f_i(Y_j = y_j \mid Z_{j1} = z_{j1}, \ldots, Z_{jg} = z_{jg}) \times P(Z_{j1} = z_{j1}, \ldots, Z_{jg} = z_{jg}) \\ &= \left\{\prod_{i=1}^{g}\left[f_i(y_j|\theta_i)^{z_{ji}}\right]\right\} \times \left\{\prod_{i=1}^{g}\pi_i^{z_{ji}}\right\} = \prod_{i=1}^{g}\left[\pi_i f_i(y_j|\theta_i)\right]^{z_{ji}}. \end{aligned}$$

La función de verosimilitud conjunta para todos los valores observados y y para el vector z de todas las no observadas z_{ij} será, por tanto:

$$L(\Psi|y, z) = \prod_{j=1}^{n}\prod_{i=1}^{g}\left[\pi_i f_i(y_j|\theta_i)\right]^{z_{ij}},$$

que tras aplicar la función logaritmo, se obtiene la función de log-verosimilitud de los datos completos:

$$\ell(\Psi|y, z) = \log\ L(\Psi|y, z) = \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log\ [\,\pi_i f_i(y_j|\theta_i)\,] = \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ [\log\ \pi_i\ +\ \log\ f_i(y_j|\theta_i)\,] =$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log\ \pi_i\ + \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log\ f_i(y_j|\theta_i) \qquad \Psi = (\pi_i, \theta_i)_{i=1,\ldots,g}. \tag{2.15}$$

Observación 2.8 Si se compara esta expresión con la (2.10), se observa que la función *log* no precede a una suma, sino que actúa directamente sobre una función de densidad, que, al ser miembro de la familia exponencial, favorece la operatoria de maximización.

La función de log-verosimilitud de una mixtura gaussiana, con $\theta_i = (\mu_i, \sigma_i^2)$, considerando la forma de (2.15) es:

$$\begin{aligned}
\ell(\Psi|y,z) &= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log \pi_i \ + \ \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log\ \phi(y_j|\mu_i,\sigma_i^2) \\
&= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log \pi_i \ + \ \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \left[-\frac{1}{2}\log 2\pi - \log\sigma_i - \frac{(y_j-\mu_i)^2}{2\sigma_i^2} \right] \\
&= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log \pi_i \ - \ \frac{1}{2}\, n\, \log 2\pi \ - \ \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\left[2\log\sigma_i + \frac{(y_j-\mu_i)^2}{\sigma_i^2} \right] \\
&= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log \pi_i \ - \ \frac{n}{2}\ \log 2\pi \ - \ \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\left[\log\sigma_i^2 + \frac{(y_j-\mu_i)^2}{\sigma_i^2} \right]. \qquad (2.16)
\end{aligned}$$

Agrupamiento (clustering) mediante modelos de mixturas

Una vez definidas las variables Z_{ij}, puede introducirse ahora el concepto de agrupamiento sobre los datos observados. Uno de los propósitos de los modelos mixtos es el de proporcionar una partición de los datos en g grupos, siendo g un número previamente establecido. La i-ésima proporción de la mixtura ($\pi_i,\ i = 1,\ldots,g$) puede interpretarse como la probabilidad *a priori* de que una observación muestral pertenezca a la población g, así:

$$P(Z_{ij} = 1) = \pi_i \qquad i = 1,\ldots,g$$

Bajo la especificación de los datos completos, el procedimiento de agrupamiento tiene como objetivo asociar cada una de las variables $z_1,\ldots,z_n$ con los datos observados $y_1,\ldots,y_n$. Una vez que el modelo de mixtura ha sido ajustado y su parámetro Ψ estimado, se proporciona un agrupamiento probabilístico de las observaciones en términos de las probabilidades posteriores (Teorema de Bayes) de pertenencia a uno u otro clúster:

$$\hat{\tau}_{ij} = P\{Z_{ij} = 1|Y_j = y_j\} = \frac{\hat{\pi}_i f_i(y_j|\hat{\theta}_i)}{\sum_{\ell=1}^{g}\hat{\pi}_\ell f_\ell(y_j|\hat{\theta}_\ell)} \qquad \ell = 1,\ldots,g \qquad j = 1,\ldots,n \qquad (2.17)$$

Por tanto, $\hat{\tau}_{1j}, \hat{\tau}_{2j}, \ldots, \hat{\tau}_{gj}$, representan las probabilidades (posteriores) de que el punto y_j pertenezca a la g-ésima componente de la mixtura. Finalmente, la asignación de una observación a uno u otro clúster se decide mediante la mayor de estas probabilidades:

$$\hat{z}_{ij} = \begin{cases} 1 & \text{si} \quad i = \arg\max_{\ell}\ \hat{\tau}_{\ell j} \\ 0 & c.c. \end{cases} \qquad i = 1,\ldots,g \qquad j = 1,\ldots,n \qquad (2.18)$$

2.2.2. Formulación del algoritmo

A continuación, se presenta la formulación del algoritmo como en McLachlan y Krishnan (1997). Cada iteración del algoritmo consta de dos etapas, la etapa de esperanza y la de maximización, nombres que son abreviados con la denominación paso E y paso M, debiendo esta terminología a Dempster et al. (1977). Como ya se dijo, la idea básica es asociar a un problema de datos incompletos dado otro de datos completos para el

cual la estimación de máxima verosimilitud es más manejable.

De forma resumida, el paso E adecúa los datos completos, lo cual incluye el cálculo de la función de verosimilitud del conjunto de datos completos. Como esta función de verosimilitud está basada parcialmente en datos no observados, es reemplazada por su esperanza condicional, dados los datos observados, utilizando el valor $\Psi^{(t)}$. Finalmente, el paso M maximiza esta función de verosimilitud de datos completos sobre Ψ. Comenzando con un conjunto de valores de parámetros iniciales adecuados, el algoritmo itera hasta la convergencia:

$$\Psi^{(0)} \longrightarrow \Psi^{(1)} \longrightarrow \cdots \longrightarrow \Psi^{(t)} \longrightarrow \Psi^{(t+1)} \longrightarrow \cdots \longrightarrow \Psi^{(\infty)} = \hat{\Psi}$$

Observación 2.9 A continuación se describe el paso E y M para la primera iteración, siendo válido para cualquier par de iteraciones consecutivas t y $t+1$.

Paso E

Sea $\Psi^{(0)}$ algún valor inicial de Ψ. Entonces, en la primera iteración, el paso E requiere el cálculo de la esperanza condicional de la función de log-verosimilitud de los datos completos, dado el dato observado y, y empleando el valor inicial $\Psi^{(0)}$, que se representa:

$$\mathsf{E}\left[\ \ell(\Psi|y,Z) \mid Y=y,\ \Psi^{(0)}\ \right] := Q(\ \Psi \mid \Psi^{(0)}\ ) \tag{2.19}$$

Observación 2.10 Esta esperanza, que utiliza el valor de Ψ, se denota habitualmente como E_{Ψ}.

Desarrollando (2.19), y teniendo en cuenta la linealidad de $\mathsf{E}(\cdot)$ sobre los datos no observados Z_{ij}:

$$\begin{aligned} Q(\ \Psi \mid \Psi^{(0)}\ ) &= \mathsf{E}\ \left[\ \ell(\Psi|y,Z) \mid Y=y,\ \Psi^{(0)}\ \right] \\ &= \mathsf{E}\ \left[\ \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}\ \log\left[\pi_i f_i(y_j|\theta_i)\right] \mid Y=y,\ \Psi^{(0)}\ \right] \\ &= \sum_{j=1}^{n}\sum_{i=1}^{g} \mathsf{E}\left[\ z_{ij} \mid Y_j=y_j,\Psi^{(0)}\right]\left[\ \log\pi_j\ + \log f_i(y_j|\theta_i)\ \right] \end{aligned} \tag{2.20}$$

Por lo tanto, el paso E solo requiere el cálculo del primer factor en (2.20):

$$\begin{aligned} \mathsf{E}\left[\ Z_{ij} \mid Y_j=y_j,\Psi^{(0)}\right] &= P(Z_{ij}=1|Y_j=y_j,\Psi^{(0)}) \\ &= \left.\frac{f_i(Y_j=y_j|Z_{ij}=1)P(Z_{ij}=1)}{\sum_{\ell=1}^{g} f_i(Y_j=y_j|Z_{\ell j}=1)P(Z_{\ell j}=1)}\right|_{\Psi^{(0)}} \\ &= \left.\frac{\hat{\pi}_i\ f_i(y_j|\hat{\theta}_i)}{\sum_{i=1}^{g}\hat{\pi}_i\ f_i(y_j|\hat{\theta}_i)}\right|_{\Psi^{(0)}} := \ \hat{\tau}_{ij}^{(0)} \end{aligned} \tag{2.21}$$

Por lo que la expresión (2.20) puede reescribirse:

$$Q(\ \Psi \mid \Psi^{(0)}\ ) = \sum_{j=1}^{n}\sum_{i=1}^{g} \hat{\tau}_{ij}^{(0)}\ [\ \log \pi_i\ +\ \log f_i(y_j|\theta_i)\ ]$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{g} \hat{\tau}_{ij}^{(0)} \log \pi_i\ + \sum_{j=1}^{n}\sum_{i=1}^{g} \hat{\tau}_{ij}^{(0)} \log f_i(y_j|\theta_i) \tag{2.22}$$

Así, $\hat{\tau}_{ij}$ en (2.17) es reemplazado por $\hat{\tau}_{ij}^{(0)}$. Estas $\hat{\tau}_{ij}^{(0)}$ representan probabilidades estimadas de que el punto y_j pertenezca a las componentes $i = 1, \ldots, g$ de la mixtura, siendo la mayor probabilidad entre ellas la que asigne el mismo a una u otra componente, es decir:

$$\hat{z}_{ij}^{(0)} = \begin{cases} 1 & \text{si} \quad i = \arg\max_j \hat{\tau}_{ij}^{(0)}(y_j|\Psi^{(0)}) \\ 0 & c.c. \end{cases} \qquad i = 1, \ldots, g \qquad j = 1, \ldots, n$$

Observación 2.11 Estas son las probabilidades posteriores ("responsabilities"). Con frecuencia, a los valores de estas probabilidades se les denomina asignaciones *soft*, dado que toman valores en [0,1], en contraste con las probabilidades de asignación a un clúster que se obtienen, por ejemplo, en el algoritmo *k-means*, que son referidas como *hard* al tomar valores en {0,1} o en {1,...,g}.

Paso M

El paso M requiere a continuación la maximización de la función Q con respecto a Ψ. Dado que π_i aparece únicamente en el primer término de (2.22) y que θ_i lo hace en el segundo, esta maximización puede realizarse independientemente. Comenzando con la maximización del primer término, es necesario resolver:

$$\frac{\partial}{\partial \pi_i}\left(\sum_{j=1}^{n}\sum_{i=1}^{g} \hat{\tau}_{ij}^{(0)}\ \log \pi_i\ + \lambda \left[\sum_{i=1}^{g} \pi_i - 1 \right] \right) = 0$$

en donde se ha introducido una restricción con un multiplicador de Lagrange (λ). Así:

$$\sum_{j=1}^{n} \hat{\tau}_{ij}^{(0)} \frac{1}{\pi_i} + \lambda = 0 \tag{2.23}$$

$$\sum_{j=1}^{n} \hat{\tau}_{ij}^{(0)} = -\lambda\, \pi_i \tag{2.24}$$

Sumando sobre g, en ambos términos de (2.24) obtenemos:

$$\sum_{i=1}^{g}\sum_{j=1}^{n} \hat{\tau}_{ij}^{(0)} = n$$

$$\sum_{i=1}^{g} -\lambda\, \pi_i = -\lambda \sum_{i=1}^{g} \pi_i = -\lambda$$

Por lo que:

$$-\lambda\ =\ n$$

Sustituyendo en (2.23) obtenemos un estimador iterativo para π_i :

$$\hat{\pi}_i^{(1)} = \frac{1}{n}\sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}. \tag{2.25}$$

La maximización de (2.22) respecto a θ_i depende de la función de densidad $f_i(y_j|\theta_i)$, extremo que se desarrolla a continuación para el caso particular de las distribuciones gaussianas:

$$\begin{aligned}\log f_i(y_j|\theta_i) &= \log \phi(y_j|\mu_i,\sigma_i^2)\\ &= -\frac{1}{2}\log(2\pi\sigma^2) - \frac{\frac{1}{2}(y_j-\mu_i)^2}{\sigma_i^2} = -\frac{1}{2}\log(2\pi) - \log\,\sigma_i\, - \frac{(y_j-\mu_i)^2}{2\,\sigma_i^2}.\end{aligned}$$

Comenzando con μ_i, la maximización incluye el cálculo de:

$$\frac{\partial}{\partial\mu_i}\sum_{j=1}^{n}\sum_{i=1}^{g}\hat{\tau}_{ij}^{(0)}\left(-\frac{1}{2}\log(2\pi) - \log\,\sigma_i\, - \frac{(y_j-\mu_i)^2}{2\,\sigma_i^2}\right) = 0, \tag{2.26}$$

Obteniendo:

$$2\sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}\left(\frac{y_j-\mu_i}{2\,\sigma_i^2}\right) = 0 \quad \Longrightarrow \sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}\,y_j = \sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}\,\mu_i \tag{2.27}$$

Que da como resultado un estimador iterativo para μ_i:

$$\hat{\mu}_i^{(1)} = \frac{\sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}\,y_j}{\sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}}. \tag{2.28}$$

Para obtener el estimador de σ_i^2, procedemos como con μ_i, derivando con respecto a σ_i^2 en (2.26), en lugar de μ_i:

$$-\sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}\frac{1}{\sigma_i} + \sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}(y_j-\mu_i)^2\,\frac{1}{\sigma_i^3} = 0$$

Obteniendo:

$$\hat{\sigma}_i^{2\,(1)} = \frac{\displaystyle\sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}(y_j-\hat{\mu}_i^{(1)})^2}{\displaystyle\sum_{j=1}^{n}\hat{\tau}_{ij}^{(0)}}. \tag{2.29}$$

2.2.3. Criterio de parada

Los pasos E y M son repetidos alternativamente hasta que se satisface un criterio de parada adecuado. En este proceso se va generando una secuencia de valores de la función de verosimilitud observada, ya definida en (2.10). Para detener la iteración pueden considerarse la diferencia absoluta

$$|\,\ell(\,\Psi^{(t+1)}\,|\,y) - \ell(\,\Psi^{(t)}\,|\,y)|,$$

o la diferencia relativa

$$\frac{|\,\ell(\,\Psi^{(t+1)}\,|\,y) - \ell(\,\Psi^{(t)}\,|\,y)|}{|\,\ell(\,\Psi^{(t)}\,|\,y)|} \tag{2.30}$$

en la mencionada secuencia, aunque también puede utilizarse la magnitud del cambio entre los parámetros estimados en cada iteración

$$|\,\Psi^{(t+1)} - \Psi^{(t)}|.$$

Si la diferencia utilizada es menor que un valor ϵ escogido previamente, el algoritmo finaliza. Si esto sucede en la iteración $(t+1)$, la estimación de Ψ es $\hat{\Psi} = \Psi^{(t+1)}$.

Seidel et al. (2000) demostraron que los resultados de la estimación dependen fuertemente de esta implementación y de la selección de los parámetros iniciales. Aunque no existe un consenso extendido sobre qué criterio de parada utilizar, los más frecuentemente usados son los basados en la verosimilitud. En este trabajo se utilizó el de la diferencia relativa, por su adimensionalidad, y en cuanto al valor máximo de tal diferencia, $\epsilon = 1 \cdot 10^{-6}$. En general, este valor puede hacerse más pequeño aún, si bien se encuentra limitado por la precisión computacional y generalmente solo redunda en un aumento del número de iteraciones obtenidas hasta la convergencia.

2.2.4. Propiedades de convergencia

Una vez elegido el criterio de parada apropiado, la convergencia del algoritmo es especialmente relevante y, en definitiva, sobre la que descansa la esencia del mismo. Una descripción detallada puede encontrarse en Wu (1983) o Dempster et al. (1977). Este último muestra la monotonía de la secuencia de valores de log-verosimilitud.

Lema 2.2 La secuencia de log-verosimilitud no decrece tras una iteración EM, esto es

$$\ell(\Psi^{t+1}) \geq \ell(\Psi^{t}) \qquad \forall t$$

Demostración 2.2 Ver Dempster et al. (1977).

De acuerdo con este resultado, y si los valores de log-verosimilitud se encuentran acotados superiormente, la secuencia converge de forma monótona a algún $L^* = L(\theta^*)$ (ver Wu, 1983). Este autor formuló condiciones de regularidad; entre otras, la compacidad del espacio parámetrico, bajo la cual cualquier secuencia de verosimilitud se encuentra acotada. Más aun, bajo condiciones débiles, por ejemplo si $Q(\Psi, \Psi^{(t)})$ es continua en Ψ y $\Psi^{(t)}$, L^* representa un punto estacionario.

Como la verosimilitud puede tener varios puntos estacionarios, la convergencia a un máximo, como se adelantó, depende de los valores iniciales. No obstante, no puede garantizarse la convergencia a un máximo local o global. Únicamente en el caso de que la verosimilitud sea unimodal, cualquier secuencia EM converge a un único estimador de máxima verosimilitud, independientemente del valor inicial (Wu, 1983).

Además, Wu (1983) demostró que si la secuencia de verosimilitud no queda atrapada en un *punto de silla*, el punto estacionario representa un máximo local. No obstante, en la práctica, esta condición es limitada y de difícil verificación. Por tanto, este autor recomendó probar con varios valores iniciales en la implementación del algoritmo, ya que una pequeña perturbación en los mencionados puntos silla provoca que el algoritmo diverja fuera del mismo (McLachlan y Basford, 1988).

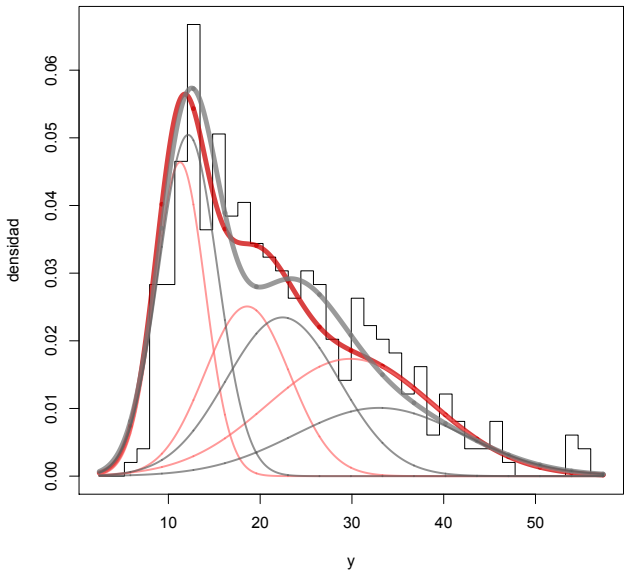
No obstante, pueden encontrarse ejemplos en donde el algoritmo EM converge a un punto de silla y no a un máximo local, como se recoge en McLachlan y Krishnan (1997), donde se discute un ejemplo adaptado de Murray (1977). En consecuencia, dado que no puede garantizarse esta convergencia a un máximo global, la recomendación de Wu sobre probar con diferentes valores iniciales cobra sentido.

2.3. El problema de los valores iniciales

El algoritmo EM es altamente dependiente del valor escogido para los parámetros iniciales, ya que influye sobre la velocidad de convergencia y la capacidad para alcanzar un máximo global (Karlis y Xekalaki, 2003). Aunque han sido propuestas numerosas estrategias para su inicialización, no existe un consenso extendido al respecto. En general, como se mencionaba en el apartado anterior, se recomienda probar varias estrategias de inicialización y seleccionar aquella que mejor resultados ofrezca.

La Figura 2.3 muestra los diferentes resultados del algoritmo EM mediante dos estrategias de inicialización sobre unos datos experimentales.

Figura 2.3: Efecto del valor de inicialización de $\hat{\Psi}$ sobre la estimación de la densidad final (línea gruesa) de la mixtura empleando el algoritmo EM. La densidad estimada de la mixtura mediante 3 componentes aparece sobreimpuesta al histograma de las observaciones. La línea gris representa la inicialización mediante el algoritmo *k-medias* (en gris) y la utilizada en este trabajo (en rojo).



El problema de los valores iniciales fue considerado por McLachlan (1988) para el caso de datos multivariantes. Este autor propuso la utilización de un diagrama de puntos en dos dimensiones como una exploración previa para detectar la presencia de clústers en la distribución en estudio. Un procedimiento similar puede ser utilizado para el caso univariante.

Estrategias para este último caso, como el que aquí se trata, pueden encontrarse en Karlis y Xekalaki (2003), donde además se incluye una revisión de trabajos previos al respecto. La utilizada en este trabajo reproduce la de Finch et al. (1989), solo para el cálculo de las medias de cada componente, por la cual se divide la muestra en g particiones y, sobre cada una de ellas, se calcula la media de las observaciones que contiene. Estos valores representan $\mu_1^{(0)}, \ldots, \mu_g^{(0)}$, es decir, la media de cada componente sobre las que el algoritmo comienza a iterar. Para las varianzas, Finch et al. (1989) obtuvieron una común, $\sigma^{2\,(0)}$, ponderada según $\mu_1^{(0)}, \ldots, \mu_g^{(0)}$; y en cuanto a las proporciones iniciales, utilizaron g números aleatorios extraidos de una distribución $U(0, 1)$. En este trabajo, para el cálculo de la desviación típica se procedió como con las medias, calculando la correspondiente

para cada partición, y respecto a las proporciones, todas semejantes, según $\pi_1^{(0)}, \ldots, \pi_g^{(0)} = 1/g$.

Ejemplo 2.2 Para una mixtura de dos componentes ($g = 2$) gaussianas de tamaño n_1 y n_2 sobre una muestra de tamaño $n = n_1 + n_2$, el cálculo de los valores iniciales se obtendría:

$$\begin{aligned}
\pi_1^{(0)} &= \pi_2^{(0)} = 1/2 \\
\mu_1^{(0)} &= \frac{1}{n_1} \sum_{s=1}^{n_1} y_s \\
\mu_2^{(0)} &= \frac{1}{n_2} \sum_{s=n_1+1}^{n} y_s \\
\sigma_1^{(0)} &= \left(\frac{1}{n_1 - 1} \sum_{s=1}^{n_1} (y_s - \mu_1^{(0)})^2 \right)^{1/2} \\
\sigma_2^{(0)} &= \left(\frac{1}{n_2 - 1} \sum_{s=n_1+1}^{n} (y_s - \mu_2^{(0)})^2 \right)^{1/2}.
\end{aligned}$$

2.4. Selección del mejor modelo

La elección del número de componentes que constituyen un modelo de mixtura no es un problema trivial y ha sido objeto de estudio en las últimas decadas. Esta elección depende de varios factores, aunque son relevantes la distribución de los datos que son modelizados y la forma de las componentes. En un contexto de discriminación, puede obtenerse dicho número empleando diferentes técnicas de agrupamiento, cada una utilizando un diferente número de componentes, y consensuar, a través de un criterio adecuado, cuál es la cantidad de componentes que mejor estructura el conjunto de datos inicial.

Otra alternativa es el empleo de los denominados "criterios de información". Existen varios de estos criterios y un estudio comparativo de ellos puede encontrarse en Bozdogan (1993), quien utilizó muestras artificiales compuestas por diferentes clústers, más o menos solapados, con diferentes formas y compacidad.

En este trabajo se utilizaron cuatro de estos criterios para consensuar el número "apropiado" de componentes. El primero de ellos, de muy extendida utilización durante las últimas decadas, ha sido el debido a Akaike (AIC) (Akaike, 1974, 1978), dado por la expresión

$$AIC = -2 \log L(\Psi) + 2\,k,$$

donde $L(\Psi)$ es la función de verosimilitud del modelo, como en la ecuación (2.9), y k, el número de parámetros independientes de la mixtura según el número de componentes propuesto ($3g - 1$). Cuando se selecciona entre los diferentes modelos, cada uno definido por un número de componentes distintos, se selecciona aquel que menor AIC presenta. El AIC penaliza el sobreajuste que se presenta con modelos grandes (elevado g) a través de k (principio de parsimonia), siendo esta característica extensible a todos los criterios aquí expuestos. No obstante, se considera que el AIC sobrestima el número de componentes (McLachlan y Peel, 2000).

Otro criterio muy utilizado es el bayesiano (BIC), propuesto por Schwarz (1978), que responde a la expresión

$$BIC = -2 \log L(\Psi) + k \log(n),$$

semejante a AIC excepto por el término de penalización, que incluye ahora el número de observaciones independientes de la muestra univariante (n). Para $n \geq 8$ este término de penalización es mayor que el de AIC, por lo que BIC es menos susceptible de sobrestimar el número de componentes.

Con el fin de comparar los resultados de estos dos criterios, también se utilizaron para la determinación del "mejor" g otros dos más recientes que los anteriores.

El criterio de verosimilitud completa integrada (ICL, *integrated complete likelihood*) fue propuesto por Biernacky et al. (2010) y es similar al BIC, excepto que añade un nuevo término de penalización. Este término, denominado entropía media, es

$$Ent(g) = -\sum_{i=1}^{g}\sum_{j=1}^{n} \hat{\tau}_{ij} \log \hat{\tau}_{ij} \geq 0$$

y representa la capacidad del modelo de mixtura para dar una partición relevante de la distribución, de tal forma que si las componentes están bien separadas, este término tiende a 0. Como en (2.17), $\hat{\tau}_{ij}$ denota la probabilidad condicional de que la observación y_j pertenezca a la g-ésima componente de la mixtura . Así, el número de clúster g' estimado por ICL es siempre menor o igual que el de BIC, debido a este término de penalización adicional que incorpora.

El cuarto criterio escogido se debe a Hurvich y Tsai (1989), el cual modifica a AIC mediante un nuevo término (AIC_c, AIC corregido), tomando la expresión

$$AIC_c = -2 \log\, L(\Psi) + 2\,k + \frac{2\,k(k+1)}{n-k-1}.$$

A medida que el tamaño de la muestra se incrementa, el último término se aproxima a 0, llegando a las mismas conclusiones que con AIC.

La elección del criterio que se utilizó en este trabajo fue finalmente el bayesiano (BIC). Para ello influyeron varios motivos. Según se observa en el Anexo I, en donde se incluyen los resultados del número de clúster tras utilizar cada uno de estos criterios, el AIC y AIC_c tienden a sobrestimar el número de componentes, mientras que el ICL parece subestimarlos. Esta apreciación, subjetiva, se corroboró mediante pruebas gráficas, enfrentando el histograma de la distribución de los datos observados con el número de componentes propuesto por estos criterios. El mayor respaldo en la literatura hacia el criterio BIC, junto con la confirmación de las pruebas gráficas, sugirieron su elección.

2.5. Obtención de errores en los estimadores

2.5.1. Método SEM

Este procedimiento, descrito por Meng y Rubin (1972), obtiene unos estimadores estables de la matriz de varianzas-covarianzas de los parámetros obtenidos mediante el algoritmo EM. Para su cálculo, emplea todas las iteraciones obtenidas hasta la obtención de los mismos, motivo por el que se le considera un producto derivado del algoritmo EM, y la matriz de información de los datos completos. Su uso no está extendido debido a ciertas peculiaridades:

1. Es susceptible de presentar inexactitudes numéricas e inestabilidad, especialmente en configuraciones de alta dimensionalidad (Baker, 1992; McCulloch, 1998; Segal et al., 1994).

2. Requiere obtener la matriz de información de datos completos, que en el caso de modelos complejos, no siempre es posible (Baker, 1992).

3. Puede ser mucho más costoso (computacionalmente) (Belin y Rubin, 1995).

No obstante, estos inconvenientes pueden ser parcialmente obviados para el caso de este trabajo, ya que, aplicado para mixturas normales univariantes, su asumible complejidad y reducida dimensión han permitido realizar su implementación sin demasiada dificultad. En particular, el coste computacional no ha sido tal, como lo demuestran los tiempos de proceso obtenido en los cálculos (resultados no mostrados) al compararlos con la aproximación replicativa (bootstrap) que en el siguiente apartado se trata.

A continuación, se interpreta el algoritmo EM como una sucesión de iteraciones, que alternan entre un paso E y otro M, y que constituye una aplicación tal que:

$$\begin{aligned} M &: \Psi \to \Psi \\ \Psi^{(t+1)} &= M(\Psi^{(t)}) \qquad t = 0, 1, \ldots \end{aligned}$$

Así, si $\Psi^{(t)}$ converge a algún punto Ψ^*, entonces Ψ^* satisface

$$\Psi^* = M(\Psi^*).$$

Mediante este planteamiento, se consigue obtener lo que los autores denominaron la tasa de convergencia del algoritmo EM, que viene dada por

$$DM = \left(\frac{\partial M_j(\Psi)}{\partial \Psi_i} \right) \Bigg|_{\Psi = \Psi^*}$$

que representa la matriz jacobiana $(d \times d)$ de $M(\Psi)$ evaluada en Ψ^*, siendo d el número de parámetros del modelo de mixtura.

Por su parte, el planteamiento de los datos completos introducido en el apartado 2.2.1 permite introducir a su matriz de información (I_{oc}), que viene dada según la expresión

$$I_{oc} = \mathsf{E}[I_o(\Psi|X)|Y, \Psi] \Big|_{\Psi = \Psi^*}$$

$$I_o(\Psi|X) = -\frac{\partial^2 \ell(\Psi|X)}{\partial \Psi^2}$$

siendo I_o la matriz de información de los datos observados.

La esencia del método SEM se debe a la siguiente expresión, debida a Meng y Rubin (1977):

$$V = I_{oc}^{-1} + I_{oc}^{-1} DM (I - DM)^{-1} \tag{2.31}$$

Siendo V la matriz de varianzas-covarianzas de los valores de los parámetros utilizando los datos completos, e $I_{d \times d}$ la matriz identidad. Los errores SEM se obtienen aplicando la raíz cuadrada a la diagonal de la matriz V, lo que equivale a los errores estándares de los estimadores EM.

El algoritmo SEM consta fundamentalmente de tres partes bien diferenciadas: (1) La evaluación de I_{oc}^{-1}, (2) la de DM y (3) la obtención final de la estimación de la matriz V.

En el caso de las mixturas gaussianas, para obtener I_{oc} se recurre a la obtención del hessiano de $Q(\Psi \mid \Psi^{(t)})$, lo que no es difícil analíticamente debido a la forma de esta distribución. Por su parte, la matriz DM requiere un cálculo numérico basado en todas las iteraciones del algoritmo EM hasta que este ha obtenido su convergencia.

Obtención de la matriz I_{oc}

Como se mencionaba, la obtención de I_{oc} implica la obtención del hessiano de $Q(\Psi \mid \Psi^{(t)})$. Con los mismos resultados, puede operarse en su lugar sobre la ecuación (2.16) y sustituir z_{ij} por $\hat{\tau}_{ij}$, multiplicando finalmente por (-1):

- Cálculo de $\frac{\partial \ell(\Psi|x)}{\partial \Psi}$:

$$\frac{\partial \ell(\Psi|y,z)}{\partial \pi_i} = \frac{1}{\pi_i} \sum_{j=1}^{n} z_{ij}$$

$$\frac{\partial \ell(\Psi|y,z)}{\partial \mu_i} = \frac{1}{\sigma_i^2} \sum_{j=1}^{n} z_{ij} \left(y_j - \mu_i\right)$$

$$\frac{\partial \ell(\Psi|y,z)}{\partial \sigma_i} = \sum_{j=1}^{n} z_{ij} \left(\frac{(y_j - \mu_i)^2}{\sigma_i^3} - \frac{1}{\sigma_i} \right)$$

- Cálculo de $\frac{\partial^2 \ell(\Psi|x)}{\partial \Psi \partial \Psi^T}$:

$$\frac{\partial^2 \ell(\Psi|y,z)}{\partial \pi_m \partial \pi_i} = \frac{\partial^2 \ell(\Psi|y,z)}{\partial \pi_i \partial \pi_m} = \begin{cases} -\frac{1}{\pi_i^2} \sum_{j=1}^{n} z_{ij} & \text{si} \quad i = m \\ 0 & \text{si} \quad i \neq m \end{cases}$$

$$\frac{\partial^2 \ell(\Psi|y,z)}{\partial \mu_m \partial \mu_i} = \frac{\partial^2 \ell(\Psi|y,z)}{\partial \mu_i \partial \mu_m} = \begin{cases} -\frac{1}{\sigma_i^2} \sum_{j=1}^{n} z_{ij} & \text{si} \quad i = m \\ 0 & \text{si} \quad i \neq m \end{cases}$$

$$\frac{\partial^2 \ell(\Psi|y,z)}{\partial \sigma_m \partial \sigma_i} = \frac{\partial^2 \ell(\Psi|y,z)}{\partial \sigma_i \partial \sigma_m} = \begin{cases} \sum_{j=1}^{n} z_{ij} \left[\frac{-3(y_j - \mu_i)^2}{\sigma_i^4} + \frac{1}{\sigma_i^2} \right] & \text{si} \quad i = m \\ 0 & \text{si} \quad i \neq m \end{cases}$$

$$\frac{\partial^2 \ell(\Psi|y,z)}{\partial \sigma_m \partial \mu_i} = \frac{\partial^2 \ell(\Psi|y,z)}{\partial \sigma_i \partial \mu_m} = \begin{cases} -\frac{2}{\sigma_i^3} \sum_{j=1}^{n} z_{ij}(y_j - \mu_i) & \text{si} \quad i = m \\ 0 & \text{si} \quad i \neq m \end{cases}$$

$$\frac{\partial^2 \ell(\Psi|y,z)}{\partial \pi_m \partial \mu_i} = \frac{\partial^2 \ell(\Psi|y,z)}{\partial \pi_i \partial \mu_m} = 0 \quad \forall j,m \qquad \frac{\partial^2 \ell(\Psi|y,z)}{\partial \pi_m \partial \sigma_i} = \frac{\partial^2 \ell(\Psi|y,z)}{\partial \pi_i \partial \sigma_m} = 0 \quad \forall j,m$$

Obtención de la matriz DM

Siendo r_{ij} el elemento (i,j)-ésimo de la matriz DM, se define

$$\Psi_{(i)}^{(k)} = (\theta_1^*, \ldots, \theta_{i-1}^*, \theta_i^{(k)}, \theta_{i+1}^*, \ldots, \theta_d^*) \tag{2.32}$$

como el conjunto de parámetros obtenidos en una iteración EM, (k), siendo d el número de parámetros del modelo. Sólo el parámetro $\theta_i^{(k)}$ es distinto al resto, ya que los demás coinciden con los valores de Ψ^*, es decir, con los valores de los estimadores EM, una vez que el algoritmo ha convergido. El cálculo de r_{ij} se obtiene

$$r_{ij} = \left(\frac{\partial M_j(\Psi)}{\partial \theta_i} \right) \Bigg|_{\Psi=\Psi^*} = \lim_{\theta_i \to \theta_i^*} \frac{M_j(\theta_i^*, \ldots, \theta_{i-1}^*, \theta_i, \theta_{i+1}^*, \ldots, \theta_d^*) - M_j(\Psi^*)}{\theta_i - \theta_i^*}$$

$$= \lim_{k \to \infty} \frac{M_j(\Psi_{(i)}^{(k)}) - \theta_j^*}{\theta_i^{(k)} - \theta_i^*} = \lim_{k \to \infty} r_{ij}^{(k)}. \tag{2.33}$$

Para llevar a cabo este procedimiento, es necesario asumir que el algoritmo EM converge en K iteraciones y que todas ellas son conocidas. Así, el algoritmo definido por Meng y Rubin comprende:

1. Fijar un $i = 1$ y obtener $\Psi_{(i)}^{(k)} = (\theta_1^*, \ldots, \theta_{i-1}^*, \theta_i^{(k)}, \theta_{i+1}^*, \ldots, \theta_d^*)$ y evaluar $M(\Psi_{(i)}^{(k)})$.

2. Calcular

$$\frac{M_j(\Psi_{(i)}^{(k)}) - \theta_j^*}{\theta_i^{(k)} - \theta_i^*}$$

   para $j = 1, \ldots, d$.

3. Repetir los pasos 1 y 2 para $i = 2, \ldots, d$.

Obteniendo la matriz DM, compuesta por r_{ij}, $i, j = 1, \ldots, d$. El elemento r_{ij} se obtiene cuando la secuencia $r_{ij}^{(k)}, r_{ij}^{(k+1)}, r_{ij}^{(k+2)}, \ldots$ es estable para algún k. El criterio de parada que se aplica suele ser más laxo que el que se aplica en las iteraciones EM y puede llegar a ser diferente según el parámetro del que se trate en Ψ, generalmente la raíz cuadrada del ϵ utilizado en el algoritmo EM (Jamshidian y Jennrich, 2000, p. 260):

$$| r_{ij}^{(k+1)} - r_{ij}^{(k)} | \approx \sqrt{\epsilon} \tag{2.34}$$

Tanner (1996) recomienda evitar los errores de redondeo que puedan cometerse debido al cálculo de esta diferencia. Igualmente, advierte que la matriz $\hat{V}$ puede no ser simétrica, por lo que sugiere en su lugar el cálculo de $\frac{1}{2}(\hat{V} + \hat{V}^T)$.

En este trabajo, en lugar de (2.34), se prefirió como criterio de parada a aquella iteración que presentara la mínima diferencia entre dos iteraciones sucesivas, representando una modificación al método habitual. Con esta modificación se evitó tener que considerar un ϵ distinto según el estimador del que se tratase.

El único inconveniente encontrado al obtener los resultados de este método está relacionado con la interrupción del algoritmo. Como describió Jamshidian y Jennrich (2000), el algoritmo se interrumpe al hacerse 0 la diferencia $\theta^{(k)} - \theta^*$ de algún componente.

2.5.2. Método Bootstrap

Para la estimación de los errores mediante este método replicativo, se ha seguido el procedimiento descrito en Basford et al. (1988) adaptado a mixturas univariantes. En esencia, consiste en generar a partir del conjunto de datos observados, y, B muestras bootstrap, $y_1^*, y_2^*, \ldots, y_B^*$, y obtener la parametrización de la mixtura mediante el algoritmo EM para cada una de ellas. Con posterioridad, se calcula el estimador bootstrap del error de muestreo ($\widehat{se_B}$) sobre cada uno de los estimadores EM de las B mixturas. Esquemáticamente:

1. A partir de la muestra de datos observados, y, generar muestras bootstrap, y_b^* $(b = 1, \ldots, B)$, de tamaño n, con B=1000.

2. Aplicar el algoritmo EM sobre cada y_i^* para obtener los estimadores EM de cada una de las mixturas obtenidas $\hat{\Psi}_1^*, \ldots, \hat{\Psi}_B^*$.

3. Sobre cada uno de los parámetros $\hat{\theta}_{bj}^*$ $(b = 1, \ldots, B\,;\, j = 1, \ldots, 3g-1)$ que componen $\hat{\Psi}_b^*$, calcular a continuación el estimador bootstrap del error estándar:

$$\widehat{se_B(\theta_j^*)} = \left(\frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}_j^*(b) - \overline{\hat{\theta}}_j) \right)^{1/2}$$

donde

$$\overline{\hat{\theta}}_j = \left(\frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_j^*(i) \right).$$

3

Otras técnicas estadísticas utilizadas

3.1. Análisis clúster jerárquico

Con está técnica multivariante se ha pretendido agrupar a las estaciones de medida (Tabla 1.1 - Estación) en función del grado de homogeneidad de la calidad del aire que en ellas se registra (Tabla 1.1 - Contaminantes estudiados). Así, estaciones con un alto grado de homogeneidad interna (niveles de calidad del aire asumiblemente similares) se situarán próximas en la representación gráfica del análisis (Figura 4.2). La métrica escogida fue la distancia euclídea, indicando las distancias más pequeñas una mayor similitud. Por ello, antes de realizar el análisis propiamente dicho, es necesario obtener una matriz de distancias, que fue calculada mediante la función dist del libro base de R.

El algoritmo empleado (función hclust, libro base) asigna inicialmente a cada objeto un clúster propio, e identifica las dos observaciones mas parecidas (cercanas) que no estén en el mismo clúster y las combina. Así, las iteraciones comienzan con cada observación en su propio clúster, combinando dos conglomerados a un tiempo, hasta que todas las observaciones se reúnen en un única *solución* clúster en función de su homogeneidad. El método aglomerativo utilizado fue el asociado al argumento single de hclust. La implementación del algoritmo se muestra en el Anexo III.

3.2. Imputación mediante bosques aleatorios

La función empleada (rfImpute, libro randomForest) comienza calculando la mediana de cada columna de la matriz de datos como valor inicial para la imputación de los datos faltantes en ellas, para posteriormente emplear el algoritmo Random Forest sobre la matriz de datos completada.

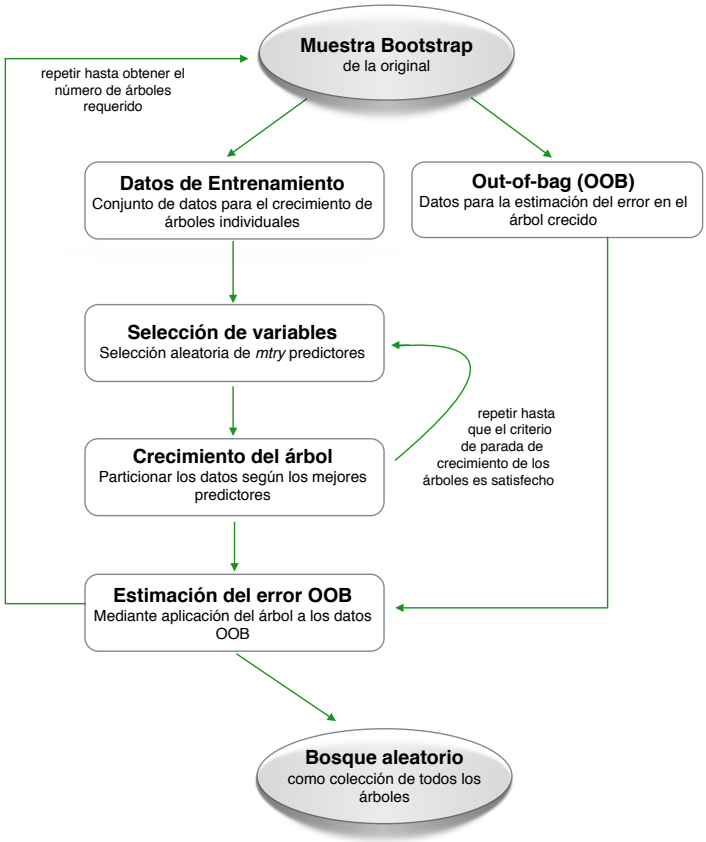
La denominación de bosques aleatorios responde a que, para la implementación del algoritmo, se utiliza un número determinado de árboles de decisión o clasificación. rfImpute precisa de los argumentos iter y ntree, con los que se especifica el número de iteraciones que se desea implementar y el número de árboles de decisión empleado en cada iteración. Estos valores fueron de 250 y 2 500, respectivamente (ver Anexo III).

El resultado de la implementación del algoritmo en R es una matriz de datos completa, en la que la variable respuesta viene representada por el tipo de estación en el que se han tomado las observaciones (Rural-R; S-Suburbana; U-Urbana), la cual es incluida en la primera columna de la matriz de datos en estudio. Las variables predictoras, continuas, se asocian a los contaminantes estudiados ($p = 5$: CO, NO_2, O_3, PM_{10}, SO_2). Al ser las variables predictoras continuas, rfImpute efectúa una regresión para la estimación de los valores imputados.

La aleatoriedad del algoritmo Random Forest se basa en dos aspectos: 1) cada árbol de decisión, entre los ntree, se crea a partir de un conjunto de observaciones de la muestra original (parte de una muestra bootstrap); y 2) cada nodo de división en cada uno de estos árboles de decisión se crea a partir de un número de variables predictoras candidatas ($mtry < p$), seleccionadas aleatoriamente. En Random Forest, el término OOB ("out of the bag") se refiere a la fracción de la muestra bootstrap que no ha sido seleccionada para la construcción

de cada árbol. Por tanto, cada iteración del algoritmo lleva asociada una cantidad OOB. Este término está asociado al error de los árboles generados en cada iteración. La predicción final con la que se obtienen los valores imputados finales se basa en el promedio de las predicciones realizadas por cada árbol. El algoritmo *bagging* es un caso especial de Random Forest cuando $mtry = p$. En la Figura 3.1 se muestra un esquema del algoritmo Random Forest.

Figura 3.1: Algoritmo Random Forest.



3.3. Análisis de componentes principales

El ACP es un método estadístico conocido desde principios de siglo (Pearson, 1901) y que consiste en describir la variación producida por la observación de p variables aleatorias, en términos de un conjunto de nuevas variables aleatorias incorreladas entre sí (denominadas *componentes principales*, -CP-), cada una de las cuales sea combinación lineal de las variables originales. Estas nuevas variables son obtenidas en orden de importancia, de manera que la primera CP incorpora la mayor variación debida a las variables originales; la segunda CP se elige de forma que explique la mayor cantidad posible de variacion que resta sin explicar por la primera CP, sujeta a la condición de ser incorrelada con la primera CP, y así sucesivamente. De esta manera, se reduce la dimensionalidad de los datos al considerar un número de variables $q\,(q < p)$ y sin perder apenas información relevante.

La aplicación perseguida del ACP en este trabajo, además de reducir la dimensionalidad, ha sido la de determinación de grupos homogéneos entre las estaciones estudiadas, a partir de los valores de μ_m y cv_m obtenidos (Tabla 4.1 y Figuras 4.3 y 4.4). Para su implementación mediante R se utilizó la función prcomp (libro base).

4

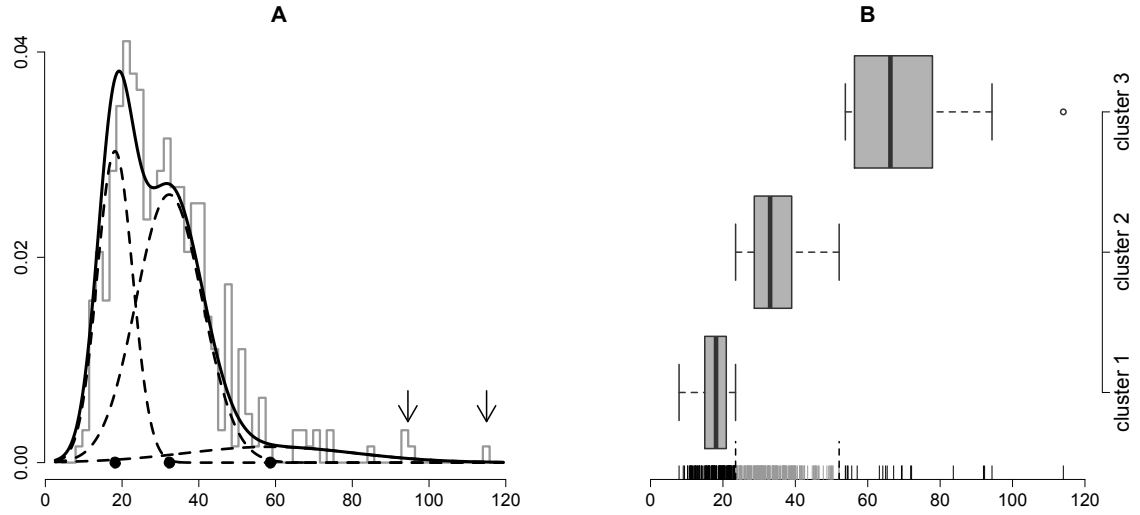
Resultados y discusión

Los modelos de distribuciones mixtas resultantes de la aplicación del algoritmo EM sobre cada conjunto de datos se muestran en el Anexo I, indicando además de la estimación de los parámetros y sus errores asociados, el tamaño de cada clúster (componente).

4.1. Análisis descriptivo y atribución de fuentes

En este apartado se explica una modelización mediante mixturas finitas entre todas los posibles (Anexo I) y su atribución de fuentes asociada. En este caso se ha elegido el modelo de PM_{10} en Alc (Alcalá de Guadaíra, Tabla A.1), representándose gráficamente en la Figura 4.1.

Figura 4.1: A. La densidad estimada de la mixtura (línea continua) y de las componentes (línea discontinua) se muestran sobreimpuestas al histograma de los datos observados (línea gris). Los puntos negros sobre el eje horizontal indican la media de cada componente y las flechas señalan las cuatro observaciones de mayor concentración. **B**. Diagramas de caja y bigote de cada clúster. Las observaciones que pertenecen a cada clúster se muestran en la base del gráfico en escala de grises y se muestran separadas unas de otras mediante una pequeña línea discontinua.



La primera componente (izquierda en la Figura 4.1A) está relacionada con la contaminación de fondo de PM_{10} en el área de Alc, y la concentración mínima obtenida fue de 7.88 $\mu g/m^3$. La segunda componente se extiende desde el rango de concentraciones de 23.50 a 52.10 $\mu g/m^3$ conteniendo el mayor número de observaciones ($n_2 = 193$). Esta segunda componente está relacionada con emisiones antropogénicas de PM_{10} de tipo primario, generadas por el tráfico viario y fundamentalmente por procesos industriales, representados por una industria de cemento presente en la localidad, cuya capacidad de producción anual es 775 641 t (E-PRTR, 2011), así como por pequeñas y dispersas canteras dedicadas a la extracción de albero. Esta presencia industrial podría llevar a pensar que esta segunda componente debería haber registrado un mayor valor que contribuyera a un aumento de la media del modelo mixto (μ_m). Sin embargo, ha de considerarse la moderada actividad industrial de la zona debida a la actual crisis económica, lo que también se refleja en un descenso del tráfico rodado. Desafortunadamente, no existen para el año de estudio datos relativos al consumo de energía eléctrica en esta localidad, si bien su consumo eléctrico industrial descendió en el periodo 2007 a 2010, de 920127 MWh (DP, 2009) a 766200 MWh (DP, 2012). La tercera componente en la mixtura (derecha) refleja las concentraciones más altas de PM_{10} y está relacionada con fenómenos meteorológicos especiales, como las intrusiones de polvo sahariano, las cuales invaden la región de Andalucía con frecuencia. Las flechas en la Figura 4.1A indican niveles de concentración de partículas de 91.96, 114.02, 92.27 y 94.32, que fueron registrados, respectivamente, el 27 y 28 de junio y el 10 y 22 de agosto. Todas estas fechas coinciden con intrusiones de polvo sahariano relevantes durante el verano y, en particular la última concentración señalada, con agresivos incendios en los países mediterráneos, incluyendo España (CIQSO, 2012). La concentración de 114.02 es clasificada como un dato anómalo en la Figura 4.1B (outlier), aunque es explicada por estos fenómenos.

La comparación de las diferentes componentes de las mixturas a través de diferentes años permite caracterizar la tendencia de la contaminación y de sus componentes en una escala de tiempo. Además, un mismo contaminante puede estudiarse a través del tiempo en una localidad, o bien ese mismo contaminante puede estudiarse para un mismo año en diferentes localidades, añadiendo una componente espacial al estudio.

4.2. Obtención de los momentos de las mixturas

Los momentos de las distribuciones de mixturas gaussianas pueden obtenerse fácilmente y representan de forma resumida la parametrización de estos modelos (Anexo I).

Con un propósito descriptivo, el valor de σ_m siempre debe entenderse en relación con el de μ_m, y el coeficiente de variación, cv_m, representa un estadístico adimensional y una medida normalizada de la variabilidad de la mixtura. Como sucede con los momentos, el uso de este último estadístico es desconocido en la literatura ambiental. El conjunto de valores μ_m y σ_m de los 49 modelos obtenidos se muestra en la Tabla 4.1. El valor del coeficiente de variación (cv_m) se refleja en la Tabla 4.3 junto con los resultados del segundo proceso de imputación.

Por tanto, μ_m y cv_m de cada modelo mixto nos permite obtener el nivel cuantitativo y la variabilidad de cada contaminante, respectivamente, y su significado cobra un mayor sentido cuando se analiza conjuntamente con la clasificación de las estaciones de monitorización, según su área y fuente de contaminación principal de donde provienen las observaciones (Tabla 1.1). En este estudio, la obtención de estos valores representa un punto de partida útil para la aplicación de posteriores análisis estadísticos para caracterizar la red de inmisión en estudio. Estos análisis se desarrollan en los siguientes apartados.

4.3. ACJ de las estaciones previo a la imputación

La información hasta ahora obtenida sugiere estudiar la relación de similitud entre las estaciones de monitorización con respecto a los valores μ_m y cv_m. Los resultados del análisis HC se muestran en la Figura 4.2.

La Figura 4.2A muestra cómo las estaciones son agrupadas de acuerdo con un nivel cuantitativo de contaminación (μ_m) con respecto a la contaminación en sus áreas de representatividad. Dos grupos de estaciones son claramente segregadas del resto: las urbanas de tráfico “Ran” y “Tor”, y las rurales “Cob” y “Sie”. Las estaciones

Tabla 4.1: Momentos de los 49 conjuntos de datos analizados mediante modelos mixtos, indicando el número de observaciones de cada uno (n) y las componentes detectadas (K), según el criterio de información BIC.

Estación	Contaminante	n	K	μ_m	σ_m
Alc	CO	361	3	308,57	51,00
	NO_2	360	2	21,06	10,22
	O_3	356	2	61,12	20,92
	PM_{10}	358	3	28,84	13,92
	SO_2	360	2	4,82	1,65
Alj	NO_2	364	3	18,44	9,39
	O_3	366	2	62,86	21,50
	PM_{10}	361	3	30,53	15,22
	SO_2	360	3	6,78	2,92
Ber	CO	365	2	480,55	148,03
	NO_2	342	3	21,64	14,37
	O_3	365	2	51,91	19,71
	PM_{10}	341	2	33,55	16,01
	SO_2	326	3	5,02	1,89
Cen	CO	340	2	677,75	276,24
	NO_2	345	2	21,46	9,03
	O_3	348	2	53,54	21,44
	SO_2	348	2	2,80	1,08
Cob	CO	302	2	206,70	74,56
	NO_2	236	3	6,54	4,34
	O_3	351	2	55,91	16,92
	PM_{10}	311	3	17,04	12,29
	SO_2	311	2	2,76	1,82
Dos	CO	320	2	463,84	123,08
	NO_2	348	2	19,44	7,17
	O_3	349	2	57,39	21,56
	SO_2	344	1	5,79	1,12
Pri	CO	342	2	412,38	156,77
	NO_2	316	2	29,39	12,40
	PM_{10}	355	2	29,82	13,88
	SO_2	351	2	5,36	1,66
Ran	CO	361	3	221,04	141,94
	NO_2	359	2	36,15	13,09
	SO_2	358	3	5,49	1,52
Saj	NO_2	358	2	22,79	7,94
	O_3	362	2	53,57	21,05
Sac	CO	358	2	367,10	86,60
	NO_2	351	2	20,87	11,04
	O_3	351	2	47,94	21,67
	PM_{10}	357	2	24,74	13,19
Sie	NO_2	349	2	3,78	1,99
	O_3	349	2	61,42	17,42
	PM_{10}	338	3	19,75	14,87
	SO_2	347	3	3,37	1,38
Tor	CO	365	2	465,83	181,32
	NO_2	365	1	33,63	15,37
	O_3	363	2	39,25	17,72
	PM_{10}	313	3	29,72	12,40
	SO_2	365	2	3,67	0,92

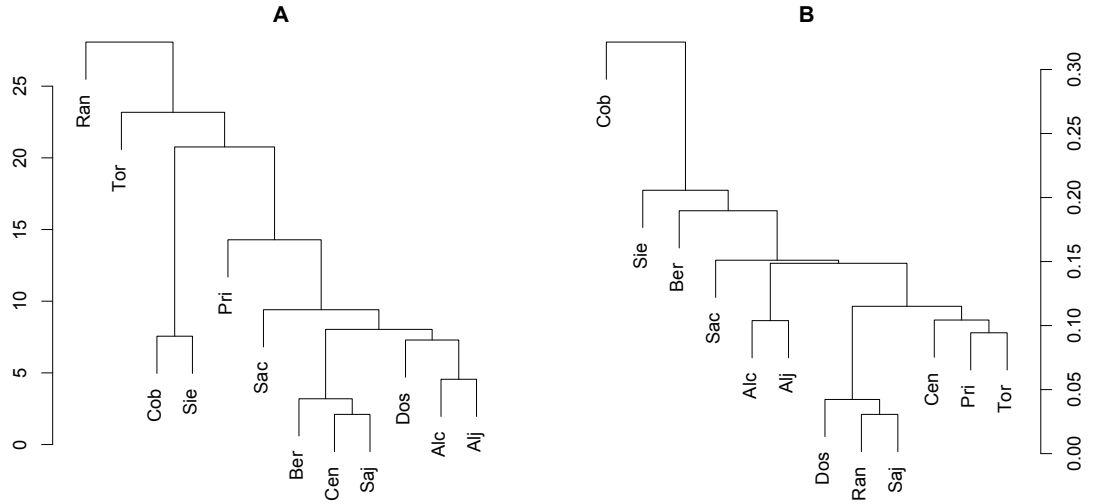
de fondo, como “Dos”, “Alc” y “Alj”, exhiben cercanía en la clasificación, y el resto de estaciones se sitúan en el dendrograma de acuerdo con sus características de contaminación. Se detecta una gran similitud entre dos estaciones “U-B” (“Ber” y “Cen”) y una estación “S-I” (“Saj”). Esta última fue clasificada como tal por su cercanía a una pequeña industria de coches, localizada en el interior de la ciudad, donde no se realizan labores de fabricación propiamente dichas, sino el montaje de piezas inertes no manufacturadas *in situ*. Por tanto, en el dendograma, teniendo en cuenta la verdadera naturaleza urbana de “Saj”, y de foma más lógica, esta se sitúa próxima a estaciones clasificadas como “U-F”, por lo que se plantea una posible clasificación erronea de esta estación.

La Figura 4.2B muestra el agrupamiento de las estaciones en función de los valores de cv_m. Inicialmente, “Cob” es situada en una hoja aparte, ya que esta estación se localiza en un área industrial representada por una de las minas de cobre más grandes de Europa. Como en la Figura 4.2A, “Sie” muestra la mayor similitud con “Cob”; y “Alc”, con “Alj” (ver los valores similares de μ_m and σ_m en la Tabla 4.1). El resto de las localizaciones de las estaciones en los dendrogramas no son comentadas en detalle, ya que responden a los valores que presentan para los estadísticos μ_m y cv_m.

4.4. Imputación de los estadísticos μ_m y cv_m y ACP

Como se mencionó anteriormente, la falta de monitorización de contaminantes en las estaciones de inmisión representa una ausencia de información al respecto. Pollice (2009) utilizó la imputación para solventar el problema de falta de datos por un funcionamiento deficiente de sensores de PM_{10}, y Ambarish et al. (2013), para estimar concentraciones de $PM_{2,5}$ en estaciones y días sin datos monitorizados. Para ampliar esta estimación

Figura 4.2: Los dendrogramas muestran la jerarquía de las estaciones en función de sus valores de μ_m (**A**) y cv_m (**B**) de acuerdo con la información de la Tabla 4.1. Los ejes verticales indican la distancia entre las estaciones, representando su similitud (baja distancia) o diferencia.



más allá del material particulado, se procedió a imputar los valores ausentes de μ_m y cv_m de aquellos contaminantes sin datos observacionales, utilizando la información del resto de contaminantes monitorizados que habían sido modelizados con estos estadísticos (Tabla 4.1). El uso de la imputación en este contexto permite optimizar el diseño de las redes de monitorización, ya que este procedimiento es equivalente al establecimiento de sensores virtuales en estaciones que no disponen de unos reales. Las Tablas 4.2 y 4.3 muestran los resultados del primer y segundo proceso de imputación (μ_m y cv_m).

Tabla 4.2: Valores imputados (en $\mu g/Nm^3$) de μ_m de contaminantes no monitorizados en las estaciones (en negrilla). Los valores no imputados coinciden con aquéllos de la Tabla 4.1 y son mostrados de nuevo por propósito comparativo.

Estaciones	Contaminantes				
	CO	NO_2	O_3	PM_{10}	SO_2
Alc	308.57	21.06	61.12	28.84	4.82
Alj	**394.12**	18.44	62.86	30.55	6.78
Ber	480.55	21.64	51.91	33.53	5.02
Cen	677.75	21.46	53.54	**27.93**	2.80
Cob	206.70	6.54	55.91	17.04	2.76
Dos	463.84	19.44	57.39	**27.69**	5.79
Pri	412.38	29.39	**54.68**	29.82	5.36
Ran	221.04	36.15	**54.83**	**28.16**	5.49
Saj	**420.50**	22.79	53.57	**28.85**	**4.87**
Sac	367.10	20.87	47.94	24.74	**4.56**
Sie	**328.00**	3.78	61.42	19.75	3.37
Tor	465.83	33.63	39.25	29.72	3.67

Al analizar los valores de la Tabla 4.2, puede detectarse alguna duplicidad en la información. Por ejemplo, "Ber" y "Cen" (estaciones U-B) monitorizan ambas NO_2, obteniendo valores casi iguales de este contaminante. Sin embargo, PM_{10} no es medido en Cen. Como una propuesta de mejora, "Cen" podría monitorizar PM_{10}, en

Tabla 4.3: Valores calculados e imputados (en negrilla) de cv_m de los contaminantes en las diferentes estaciones.

Estaciones	Contaminantes				
	CO	NO_2	O_3	PM_{10}	SO_2
Alc	0.17	0.49	0.34	0.48	0.34
Alj	**0.33**	0.51	0.34	0.50	0.43
Ber	0.31	0.66	0.38	0.48	0.38
Cen	0.41	0.42	0.40	**0.52**	0.39
Cob	0.36	0.66	0.30	0.72	0.66
Dos	0.27	0.37	0.38	**0.52**	0.19
Pri	0.38	0.42	**0.39**	0.47	0.31
Ran	0.64	0.36	**0.40**	**0.47**	0.28
Saj	**0.37**	0.35	0.39	**0.48**	**0.32**
Sac	0.24	0.53	0.45	0.53	**0.33**
Sie	**0.31**	0.53	0.28	0.75	0.41
Tor	0.39	0.46	0.45	0.42	0.25

lugar de NO_2. Si esta opción no es considerada, monitorizar NO_2 en "Cen"/"Ber" es también posible, ya que una información similar es obtenida de "Ber"/"Cen". "Ran" y "Tor" (estaciones U-T) registran similares niveles de PM_{10} (el valor en "Ran" es imputado). Por tanto, el gestor de red podría intercambiar la monitorización de estos contaminantes entre estaciones si similares niveles de PM_{10} son de nuevo detectados después de un periodo de estudio. Otra duplicidad puede encontrarse en "Dos" y "Pri" (estaciones U-B) con respecto a SO_2. Como se deduce de estos análisis, pueden obtenerse otras posibilidades de configuración de la red de inmisión, además de las reseñadas, basadas en el conocimiento que de esta posea su gestor y de las necesidades de explotación de la misma.

Otra aplicación de la imputación es la de proporcionar una estimación cuando el número mínimo de observaciones requerido durante un año para un contaminante sea inferior al deseado. En el caso de NO_2-Cob, con $n = 236$, el valor modelizado (6.54, tabla 4.1) podría ser eliminado de la matriz de imputación y realizar de nuevo el proceso de imputación para obtener una estimación de su media anual.

El proceso de imputacion explicado en este apartado considera dos ratios de información. El primero es la proporción de valores observados a no observados (49/11), y el segundo, el número de estaciones estudiadas a su clasificación por tipo (12 estaciones/3 - Urbanas, Suburbanas y Rurales). En líneas generales, no se recomienda reducir ninguna de estas proporciones (aspecto tenido en cuenta en la propuesta de monitorización de Cen-PM_{10}) cuando se estudie una reconfiguración en la red de inmisión por parte de la administración ambiental. Los resultados de la Tabla 4.2 se pueden analizar conjuntamente con los de la Tabla 4.3, aunque la información de esta última no debería ser determinante para adoptar ninguna decisión, sino para complementar la información de la Tabla 4.2.

Una vez obtenida la información completa de los niveles de contaminación tras la imputación, el estudio sugiere aplicar un ACP para mejorar la comprensión de la red en cuanto a la agrupación de las estaciones en función de los niveles y variabilidad de los contaminantes.

4.4.1. ACP

Para proporcionar una diferente aproximación del ACJ, se realizaron dos ACP para estudiar el agrupamiento de las estaciones en función de sus valores μ_m y cv_m, basándonos en la información mostrada en las Tablas 4.2 y 4.3. Debido a las características de este estudio, en ambos análisis fue necesario introducir una tercera componente para explicar una cantidad aceptable de varianza, ya que inicialmente solo se obtuvo el 77,0 % y 79,0 % respectivamente. Los valores de μ_m fueron normalizados, dada la diferente magnitud de las concentraciones en los diferentes parámetros estudiados.

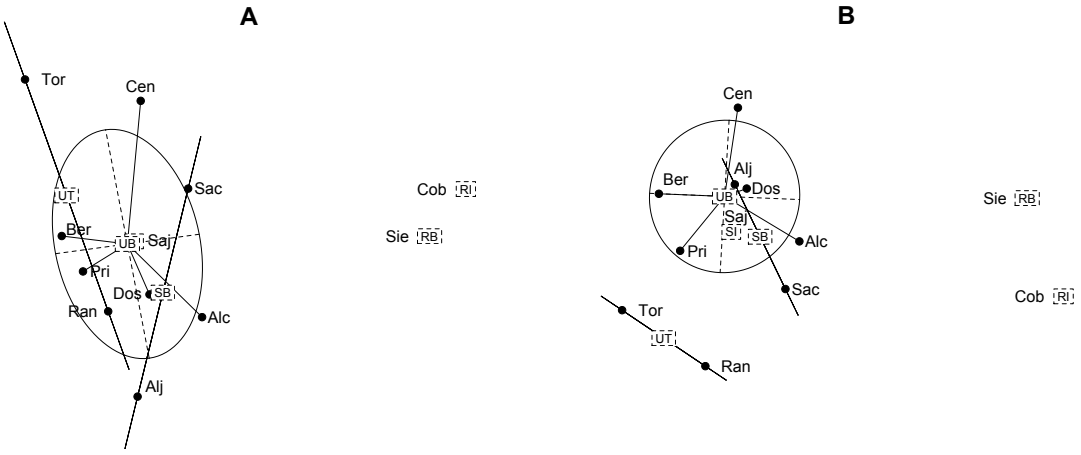
Mediante un análisis con tres componentes, la varianza explicada fue del 94,6 % y 94,3 %, considerándose valores óptimos para el propósito perseguido. Las coordenadas de cada punto (estaciones) en un sistema de tres dimensiones se muestran en el Anexo III (Tablas C.1 y C.2), mientras que en esta sección, el gráfico que

relaciona los componentes principales (PC) PC2 a PC3 se omite por no resultar decisivo para demostrar el agrupamiento de las estaciones.

La Figura 4.3 muestra los resultados del ACP con respecto a μ_m. Como puede verse en la Tabla C.3, las cargas más altas se encuentran en las variables NO_2 y PM_{10} para PC1 (signo negativo), O_3 y SO_2 para PC2 (signo negativo) y CO para PC3 (signo positivo). El ACP sitúa a "Cob" (RI) , "Sie" (RB) y "Tor" (UT) bien separadas del resto, y en menor alcance "Ran"(UB) y "Cen"(UB). El resto de estaciones situadas dentro del elipsoide permanecen más o menos distantes del centro de esta figura dependiendo de sus características. La interpretación que a continuación se incluye precisa que el gestor de la red posea un conocimiento en detalle tanto de la configuración de la red de inmisión como de las características de sus estaciones.

En la Figura 4.3A, los valores más altos de PM_{10} y NO_2 aproximan las estaciones hacia la izquierda del eje PC1, mientras que los valores superiores de SO_2 y O_3 las sitúan en la parte inferior de PC2. Como resultado, las estaciones de tráfico "Tor" y "Ran" caen a la izquierda de PC1 y las estaciones U-B, "Ber" y "Pri", entre ellas. Esta últimas estaciones poseen una gran influencia de tráfico debido a su localización próxima a vías principales y, por tanto, el ACP las sitúa a la izquierda (mayores niveles de PM_{10} y NO_2) de "Ran". Las estaciones rurales ("Sie" y "Cob") se localizan en la parte derecha de PC1, ya que ellas registran los valores más bajos de PM_{10} y NO_2.

Figura 4.3: PCA basado en valores normalizados de μ_m. PC1 se representa en el eje horizontal. Las etiquetas dentro de un recuadro conectan estaciones con la misma clasificación. **A**. Primera y segunda componente (PC2 en el eje vertical). **B**. Primera y tercera componente (PC3 en el eje vertical).

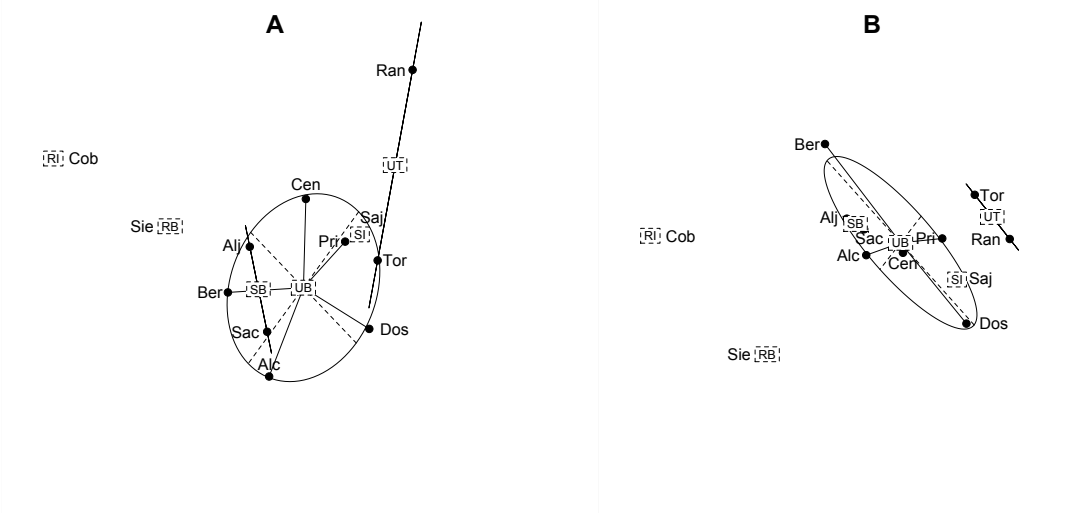


Las estaciones localizadas en el centro urbano de Sevilla ("Cen" y "Sac") registran un valor más bajo de O_3 y se localizan en la parte superior del PC2. A medida que nos acercamos hacia el área suburbana de la ciudad, la concentración de ese contaminante aumenta, lo que se refleja en las estaciones suburbanas ("Alj", "Alc" y "Dos"), colocándolas más abajo en el eje PC2. Teniendo en cuenta que Sevilla no es una ciudad altamente industrializada, la contribución de SO_2 es inferior en comparación con otros parámetros analizados. En la Figura 4.3B, las estaciones se localizan en la parte superior del eje PC2, donde se reflejan las concentraciones superiores de CO, y, por tanto, las estaciones en el centro urbano aparecen en esa posición.

El PCA con respecto a cv_m (Figura 4.4) describe el agrupamiento de las estaciones respecto a la variabilidad de las concentraciones de los contaminantes monitorizados basados en modelos mixtos, aunque la interpretación de estas figuras es menos intuitiva. Las cargas más altas en los PCs destacan en SO_2 para PC1 (signo negativo), CO para PC2 (signo positivo), y NO_2 y PM_{10} para PC3 (signos positivo y negativo, respectivamente). Mientras que, por el contrario, el O_3 no se relaciona fácilmente con ningún PC (Tabla C.4). Como puede percibirse,

las estaciones rurales ("Sie" y "Cob") exhiben una variabilidad en la concentración de contaminantes diferente de las estaciones urbanas, y entre estas, pueden distinguirse claramente las estaciones de fondo (dentro del elipsoide) y de tráfico ("Tor" y "Ran"). En la Figura 4.4A, las estaciones se encuentran posicionadas sobre una fracción más pequeña de variación para el SO_2. Las estaciones localizadas en el centro urbano muestran una pequeña variación de este contaminante, ya que, como se ha mencionado, Sevilla no es una ciudad industrial. La mayor variación de CO coloca a las estaciones en la parte superior del eje PC2. La estación de tráfico "Ran" presenta la mayor variación en sus niveles de contaminantes. Esta interpretación cualitativa, completa aquella cuantitativa basada en el ACP considerando μ_m.

Figura 4.4: ACP basado en valores no normalizados de cv_m con los ejes y las distribución de las PCs como en la figura 3.



No obstante, los resultados del ACP deben interpretarse analizando conjuntamente los resultados del ACJ realizado previamente a la imputación de los estadísticos para confirmar los agrupamientos de las estaciones. Como era de esperar, en la Figura 4.2A y 4.3, "Ran", "Tor", "Cob" y "Sie" son establecidos aparte antes y después de la imputación, y también "Saj" permanece entre estaciones del tipo U-B y S-B, confirmando los resultados del ACJ. Las Figuras 4.2B y 4.4 también mantienen estos emplazamientos. La información añadida después de la imputación puede recolocar estaciones en posiciones diferentes cuando se utiliza cualquier técnica de agrupamiento, y esto puede variar los resultados del ACJ inicial. El conocimiento de la red y la experiencia del gestor son cruciales para dar sentido a estas recolocaciones. Los resultados de los ACJ y ACP basados en μ_m y cv_m, como se esperaba, revelan que, a pesar de las condiciones cambiantes locales atmosféricas y ambientales en las estaciones durante el periodo de estudio, pueden asumirse unos niveles de contaminación y variabilidad en concordancia con sus emplazamientos y predominantes fuentes de emisión.

5

Resumen y conclusiones

Se han empleado modelos de mixturas finitas para ajustar distribuciones de datos observacionales de 5 contaminantes clave monitorizados, según disponibilidad, en 12 estaciones de inmisión durante 2012 en Sevilla. Los 49 modelos resultantes caracterizaron de forma precisa los contaminantes monitorizados y constituyen una herramienta útil para el gestor, por su capacidad para detectar fuentes de contaminación y comparar las tendencias de estas a lo largo del tiempo y el espacio. Pero los modelos de mixturas no deberían ser utilizados únicamente con un propósito descriptivo. La media y desviación típica de cada modelo (μ_m y σ_m) se pueden obtener fácilmente para su caracterización, y derivado de ellos, el coeficiente de variación, representando una huella dactilar de los mismos. Su aplicación en este trabajo supone una aproximación que abre nuevas vías para la caracterización de los modelos mixtos en la disciplina de la contaminación atmosférica. Basado en ellos, el ACJ detectó una clara separación entre las estaciones de acuerdo con su nivel de contaminación (U-T) o emplazamiento (estaciones rurales) del resto.

Cuando el ratio de información perdida es apropiado, la imputación de μ_m y cv_m representa una estrategia valiosa para estimar el nivel y variabilidad de los contaminantes no monitorizados en las estaciones, o cuando la ausencia de datos entre los monitorizados es destacable. Esta imputación permite optimizar el diseño de la red en estudio. Una vez que se obtienen los valores imputados, pueden emplearse análisis estadísticos posteriores para completar la información de la red obtenida hasta el momento, y también para plantear alternativas de configuración en la red, como fue el caso de "Cen" y "Ber" con respecto a la monitorización de NO_2 y PM_{10}. Atendiendo a todo lo anterior, es esencial que el gestor disponga de un conocimiento exhaustivo de la red para poder interpretar correctamente los resultados obtenidos mediante estos análisis estadísticos, con el fin de poder adoptar medidas de mejora si fuera necesario. Este podría ser el caso de la posible clasificación incorrecta de "Saj".

Aunque una de las principales ventajas de la aplicación secuencial de estos procedimientos estadísticos es la de conocer en profundidad la red de monitorización en operación, su aplicación a lo largo de varios años puede resultar incluso más valiosa. Esta orientación puede ayudar a entender cómo la contaminación ha evolucionado durante un periodo de tiempo concreto y cómo la red se ha comprtado en consonancia.

En este trabajo, aunque la implementación computacional del algoritmo EM, el método bootstrap y el SEM fueron desarrollados específicamente para la ocasión en R, existen libros en ese entorno, como mclust, que pueden ser utilizadas para obtener los mismos resultados que los aquí mostrados sin conocimientos destacables de estadística o programación. El resto de análisis empleados también están disponibles bajo ese entorno. Más aún, conocidos programas estadísticos comerciales incorporan la mayoría de las técnicas aquí empleadas, lo que facilita la incorporación de estas técnicas a la explotación rutinaria de las redes de monitoriación.

6

Bibliografía

Aitkin, M. y Aitkin, I. 1996. A hybrid EM/Gauss-Newton algorithm for maximum likelihood in mixture distributions. *Statistics and Computing*, 6: 127-30.

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

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions On Automatic Control*, 19: 716-23.

Akaike, H. 1978. On the likelihood of a time series model. *The Statistician*, 27: 217-35.

Baker, S.G. 1992. A simple method for computing the observed information matrix when using the EM algorithm. *Journal of Computational and Graphical Statistics* 1: 63-76.

Baum, L.E., Petrie, T., Soules, G. y Weiss, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1): 164-71.

Belin, T.R., y Rubin, D.B. 1995. A method for calibrating false-match rates in record linkage. *Journal of the American Statistical Association*, 90: 694-707.

Biernacki, C., Celeux, G. y Govaert, G. 2000. Assessing a Mixture Model for Clustering With the Integrated Completed Likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22: 719-25.

Blischke, W. R. 1962. Moment estimators for the parameters of a mixture of two binomial distributions. *The Annals of Mathematical Statistics*, 33(2): 444-54.

Bozdogan, H. 1993. Choosing the number of component clusters in the mixture-model using a new informational complexity criterion of the inverse-Fisher information matrix. In O. Optiz, B. Lausen y R. Klar, editores, Information and Classification, páginas 40-54. Springer-Verlag.

Chang, N.B. y Tseng, C.C. 1999. Optimal design of multi-pollutant air quality monitoring network in a metropolitan region using Kaohsiung, Taiwan as an example. *Environmental Monitoring and Assessment*, 57(2):121-48.

Charlier, C. 1906. Researches into the theory of probability. Hakon Ohlsson: Lund.

Charlier, C. y Wicksell, S. 1924. On the dissection of frequency functions. *Arkiv för matematik*, astronomi och fysik.

Cohen, A. 1967. Estimation in mixtures of two normal distributions. *Technometrics*, 9(1), pp. 15-28.

Day, N. 1969. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3): 463-74.

Delmar, P., Robin, S., Tronik-Le, R. y Daudin, J.J. 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.

Dempster, A., Laird, N. y Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B (Methodological), 39(1): 1-38.

Du, J. 2002. Combined algorithms for fitting finite mixture distributions. Master's thesis, Mc-Master University Hamilton, Ontario.

España. Real Decreto 812/2007, de 22 de junio, sobre evaluación y gestión de la calidad del aire ambiente en relación con el arsénico, el cadmio, el mercurio, el níquel y los hidrocarburos aromáticos policíclicos. Boletín Oficial del Estado, 23 de junio de 2007, núm. 150, pp. 27171-27177.

España. Ley 34/2007, de 15 de noviembre, de calidad del aire y protección de la atmósfera. Boletín Oficial del Estado, 16 de noviembre de 2007, núm. 275, pp. 46962-46987.

España. Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire. Boletín Oficial del Estado, 29 de enero de 2011, núm. 25, pp. 9574-9626.

Everitt, B. 1996. An introduction to finite mixture distributions. *Statistical Methods in Medical Research*, 5(2): 107-27.

Everitt, B. y Hand, D. 1981. Finite Mixture Distributions. Chapman and Hall, London.

Falls, L. 1970. Estimation of parameters in compound Weibull distributions. *Technometrics*, 12(2): 399-407.

Finch, S.J., Mendel, N.R. y Thode, H.C.Jr. 1989. Probabilistic measures of adequacy of a numerical search for a global maximum. *Journal of the American Statistical Association*, 84(408): 1020-3.

Frühwirth-Schnatter, S. 2010. Finite Mixture and Markov Switching Models. Springer, New York.

Hasselblad, V. 1966. Estimation of parameters for a mixture of normal distributions. *Technometrics*, 8(3): 431-44.

Hasselblad, V. 1969. Estimation of finite mixtures of distributions from the exponential family. *Journal of the American Statistical Association*, 64(328): 1459-71.

Healy, M. y Westmacott, M. 1956. Missing values in experiments analysed on automatic computers. *Journal of the Royal Statistical Society*, Series C (Applied Statistics), 5(3): 203-206.

Holgersson, M. y Jorner, U. 1978. Decomposition of a mixture into normal components: A review. *International Journal of Bio-Medical Computing*, 9(5): 367-92.

Hurvich, C.F. y Tasi, C.L. 1989. Regression and time series model selection in small samples. *Biometrika*, 76(2): 297-307.

Jamshidian, M. y Jennrich, R.I. 2000. Standard errors for EM estimation. *Journal of the Royan Statistical*

Society, series B. 62(2): 257-70.

Karlis, D. y Xekalaki, E. 2003. Choosing initial values for the EM algorithm for finite mixtures. *Computational Statistics & Data Analysis*, 41(3-4): 577-90.

Li, S., Batterman, S., Su, F. y Mukherjee, B. 2013. Addressing extrema and censoring in pollutant and exposure data using mixture of normal distributions. *Atmospheric Environment*, 77: 464-73.

Little, R.J.A. y Rubin, D.B. 2002. Statistical Analysis with Missing Data. Wiley Series in Probability and Mathematical Statistics, Hoboken.

McCulloch, C.E. (1998). Review of "EM Algorithm and Extensions". Journal of the American Statistical Association 93:403-404.

McKendrick, A. 1926. Applications of mathematics to medical problems. *Proceedings of the Edinburgh Mathematical Society*, 44: 98-130.

McLachlan, G. y Basford, K. 1988. Mixture models: inference and applications to clustering. Dekker, New York.

McLachlan, G. J. y Jones, P. N. 1988. Fitting mixture models to grouped and truncated data via the EM algorithm. *Biometrics*, 44(2): 571-78.

McLachlan, G. y Krishnan, T. 1997. The EM Algorithm and Extensions. Wiley Series in Probability and Statistics, New York.

McLachlan, G. y Peel, D. 2000. Finite Mixture Models. Wiley Series in Probability and Statistics, New York.

Medgyessy, P. 1961. Decomposition of superpositions of distribution functions. Publishing House of the Hungarian Academy of Sciences, Budapest.

Meng, X.L. y Pedlow, S. 1992. EM: A bibliographic review with missing articles. Statistical Computing Section, Prococeedings of the American Statistical Association. Alexandria, VA: 24-27.

Meng, X.L. y Rubin, D.B. 1991. Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm. Journal of the American Statistical Association, 86(416): 899-909.

Mengerse, K., Robert, C. y Titterington, D. 2011. Mixtures: Estimation and Applications. Wiley Series in Probability and Mathematical Statistics.

Moharir, P. 1992. Estimation of the compounding distribution in the compound Poisson process model for earthquakes. *Journal of Earth System Science*, 101(4): 347-59.

Murray, G.D. 1977. Contribution to discussion of paper by A.P. Dempster, N.M. Laird and D.B. Rubin. *Journal of the Royal Statistical Society*, Series B, 39: 23-24.

Newcomb, S. 1886. A generalized theory of the combination of observations so as to obtain the best result. American Journal of Mathematics, 8(4): 343-66.

Orchard, T. y Woodbury, M. 1972. A missing information principle: theory and applications. Proceedings of the Sixth Berkeley Symposium on Mathematical Statistic and Probability, 1: 697-715.

Pires, J.C.M., Sousa, S.I.V., Pereira, M.C., Alvim-Ferraz, M.C.M. y Martins, F.G. 2008. Management of air quality monitoring using principal component and cluster analysis-Part I: SO2 and PM10. *Atmospheric Environment*, 42: 1249-60.

R Core Team. 2013. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org/`.

Rao, C. 1948. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society*, Series B (Methodological), 10(2): 159-203.

Redner, R. y Homer, W. F. 1984. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2): 195-239.

Schlattmann, P. 2009. Medical Applications of Finite Mixture Models. Springer, Berlin Heidelberg.

Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461-4.

Seidel, W., Mosler, K. y Alker, M. 2000. A cautionary note on likelihood ratio tests in mixture models. *Annals of the Institute of Statistical Mathematics*, 52(3): 481-7.

Segal, M.R., Bacchetti, P. y Jewell, N.P. 1994. Variances for maximum penalized likelihood estimates obtained via the EM algorithm. *Journal of the Royal Statistical Society*, Series B: 56: 345-52.

Simar, L. 1976. Maximum likelihood estimation of a compound Poisson process. *The Annals of Statistics*, 4(6): 1200-9.

Tan, W. Y. y Chang, W. C. 1972. Some comparisons of the method of moments and the method of maximum likelihood in estimating parameters of a mixture of two normal densities. *Journal of the American Statistical Association*: 67(339): 702-8.

Tanner, M.A. 1996. Tools for statistical inference. Methods for the exploration of posterior distributions and likelihood functions, 3a edición. Springer Serires in Statistics, New York.

Teicher, H. 1961. Identifiability of mixtures. *The Annals of Mathematical Statistics*, 32(1): 244-8.

Teicher, H. 1963. Identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 34(4): 1265-9.

Titterington, D., Smith, A. y Makov, U. 1985. Statistical Analysis of finite mixture distributions. John Wiley & Sons, Chichester.

Unión Europea. Directiva 2004/107/CE del Parlamento Europeo y del Consejo de 15 de diciembre de 2004 relativa al arsénico, el cadmio, el mercurio, el níquel y los hidrocarburos aromáticos policíclicos en el aire ambiente. Diario Oficial de la Unión Europea L 23/3, 26 de enero de 2005, pp. 3-16.

Unión Europea. Directiva 2008/50/CE del Parlamento Europeo y del Consejo de 21 de mayo de 2008 relativa a la calidad del aire ambiente y a una atmósfera más limpia en Europa. Diario Oficial de la Unión Europea L 152/1, 11 de junio de 2008, pp. 1-44.

Wilks DS. 2006. Statistical methods in the atmospheric sciences. 2a ed. Academic Press.

Wu, C. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1), pp. 95-103.

Yakowitz, S.J. y Spragins, J.D. 1968. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39(1): 209-14.

A

Anexo I

A.1. Parametrizaciones obtenidas de las mixturas

En este apartado se recogen los valores de $\hat{\Psi}$ de los 49 modelos resultantes tras aplicar el algoritmo EM a los 49 conjuntos de datos según los parámetros analizados en las estaciones de monitorización de Sevilla (Tabla 1.1).

Cada tabla recoge el número de componentes resultantes según el criterio de información elegido (K_{BIC}, K_{AIC}, K_{AIC_c} y K_{ICL}), el número de observaciones en cada componente (n_k) según K_{BIC}, el parámetro en cuestión (θ_k) y los errores de estos utilizando el método SEM (e.e. SEM) o bootstrap ($\widehat{se_B}$).

Cuando una de las componentes ha presentado una representatividad cercana al cero ($\pi_i \simeq 0$, $i = 1, 2, 3$), el valor de π se ha incluido en la tabla correspondiente y resaltado en negrita. Este ha sido el motivo, en la mayoría de los casos, por el que no se han podido obtener los errores SEM en la parametrización de algún modelo por interrupción del algoritmo.

Tabla A.1: Alcalá de Guadaíra.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Alc	CO	3	83 67 211	π_1	0.23	0.026	0.024	6	6	2
				π_2	0.15	0.026	0.098			
				μ_1	230.78	1.36	1.68			
				μ_2	289.13	1.70	4.38			
				μ_3	337.47	3.28	9.71			
				σ_1	10.40	1.068	1.27			
				σ_2	5.84	1.61	4.81			
				σ_3	38.41	2.38	6.66			
	NO_2	2	176 184	π_1	0.43	0.040	0.049	3	3	1
				μ_1	12.84	0.32	0.66			
				μ_2	26.80	0.85	0.94			
				σ_1	3.60	0.24	0.43			
				σ_2	9.39	0.56	0.50			
	O_3	2	169 187	π_1	0.50	0.042	0.11	3	3	1
				μ_1	44.44	2.37	5.30			
				μ_2	75.37	0.94	2.77			
				σ_1	15.79	1.56	2.33			
				σ_2	10.83	0.66	1.69			
	PM_{10}	3	145 193 20	π_1	0.35	0.051	0.049	3	3	2
				π_2	0.56	0.069	0.057			
				μ_1	18.16	0.88	0.86			
				μ_2	32.28	1.25	1.61			
				μ_3	58.59	5.34	7.68			
				σ_1	4.61	0.60	0.41			
				σ_2	8.62	0.65	0.90			
				σ_3	21.95	3.60	3.33			
	SO_2	2	128 232	π_1	0.34	0.032	0.027	5	5	2
				μ_1	2.73	0.050	0.050			
				μ_2	5.80	0.079	0.073			
				σ_1	0.49	0.036	0.027			
				σ_2	1.07	0.061	0.067			

Tabla A.2: Aljarafe.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Alj	NO_2	3	91 171 102	π_1	0.22	0.033	0.050	3	3	1
				π_2	0.45	0.042	0.063			
				μ_1	7.46	0.30	0.51			
				μ_2	14.99	0.67	0.90			
				μ_3	27.39	1.18	1.58			
				σ_1	1.60	0.22	0.34			
				σ_2	4.14	0.41	0.39			
				σ_3	7.05	0.63	0.76			
	O_3	2	167 199	π_1	0.49	0.037	0.090	3	3	1
				μ_1	46.67	1.21	4.49			
				μ_2	81.09	0.98	2.93			
				σ_1	15.12	1.18	2.26			
				σ_2	11.76	0.74	1.40			
	PM_{10}	3	126 216 19	π_1	0.30	–	0.067	3	3	2
				π_2	0.62	–	0.068			
				$\boldsymbol{\pi_3}$	0.076	0.020	0.037			
				μ_1	18.06	–	1.25			
				μ_2	32.18	–	1.40			
				μ_3	63.14	6.49	7.50			
				σ_1	4.34	0.15	0.71			
				σ_2	8.75	0.47	0.59			
				σ_3	22.09	4.18	3.62			
	SO_2	3	69 158 133	π_1	0.18	0.023	0.022	3	3	1
				π_2	0.35	0.057	0.084			
				μ_1	1.38	0.097	0.10			
				μ_2	7.80	0.12	0.16			
				μ_3	8.00	0.22	0.32			
				σ_1	0.75	0.072	0.062			
				σ_2	0.83	0.11	0.22			
				σ_3	2.42	0.19	0.24			

Tabla A.3: Bermejales.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Ber	CO	2	305 60	π_1	0.72	0.073	0.12	2	2	1
				μ_1	410.82	9.73	16.76			
				μ_2	577.80	32.17	44.10			
				σ_1	101.78	6.12	8.26			
				σ_2	167.90	18.55	17.35			
	NO_2	3	46 183 113	$\boldsymbol{\pi_1}$	0.11	0.020	0.080	3	3	1
				π_2	0.51	–	0.088			
				π_3	0.37	0.026	0.069			
				μ_1	2.97	0.11	2.13			
				μ_2	15.59	0.90	2.12			
				μ_3	35.13	1.00	2.95			
				σ_1	1.84	0.24	1.28			
				σ_2	6.64	–	0.75			
				σ_3	11.57	0.88	1.31			
	O_3	2	73 292	π_1	0.19	0.031	0.087	2	2	1
				μ_1	23.57	1.92	4.15			
				μ_2	58.04	1.14	2.60			
				σ_1	6.08	0.96	2.55			
				σ_2	15.33	0.80	1.21			
	PM_{10}	2	315 26	π_1	0.87	0.062	0.10	3	3	2
				μ_1	29.14	0.73	1.80			
				μ_2	56.67	5.71	8.51			
				σ_1	10.83	0.54	0.95			
				σ_2	20.14	3.44	3.13			
	SO_2	3	116 129 81	π_1	0.31	0.070	0.052	5	5	2
				π_2	0.39	0.14	0.069			
				π_3	0.30	0.056	0.076			
				μ_1	3.45	0.052	0.080			
				μ_2	4.74	–	0.26			
				μ_3	7.01	0.31	0.54			
				σ_1	0.29	0.036	0.047			
				σ_2	0.76	0.20	0.15			
				σ_3	1.93	0.16	0.25			

Tabla A.4: Centro.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Cen	CO	2	266 74	π_1	0.79	0.051	0.16	8	8	2
				μ_1	572.25	9.95	35.58			
				μ_2	1188.43	33.52	154.41			
				σ_1	141.34	7.81	35.37			
				σ_2	158.95	23.24	74.02			
	NO_2	2	129 216	π_1	0.35	0.045	0.072	2	2	1
				μ_1	12.17	0.62	1.063			
				μ_2	26.77	0.75	1.14			
				σ_1	4.17	0.42	0.57			
				σ_2	7.24	0.50	0.59			
	O_3	2	92 256	π_1	0.25	0.036	0.12	3	3	1
				μ_1	24.41	1.77	5.88			
				μ_2	62.26	1.35	4.038			
				σ_1	8.09	1.09	3.36			
				σ_2	15.78	0.96	1.93			
	SO_2	2	253 95	π_1	0.64	0.066	0.064	2	2	1
				μ_1	2.23	0.059	0.074			
				μ_2	3.66	0.17	0.17			
				σ_1	0.58	0.041	0.043			
				σ_2	1.099	0.087	0.072			

Tabla A.5: Cobre Las Cruces.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Cob	CO	2	152 150	π_1	0.50	0.038	0.029	2	2	2
				μ_1	140.06	2.53	3.03			
				μ_2	282.64	2.26	2.64			
				σ_1	33.19	1.85	2.050			
				σ_2	29.39	1.67	2.26			
	NO_2	3	23 126 87	$\boldsymbol{\pi_1}$	0.084	0.018	0.084	5	5	1
				π_2	0.48	–	0.16			
				π_3	0.44	–	0.18			
				μ_1	0.33	0.13	0.84			
				μ_2	5.37	0.77	1.34			
				μ_3	9.94	0.66	2.42			
				σ_1	0.21	0.042	0.56			
				σ_2	2.62	0.32	0.58			
				σ_3	4.00	0.67	0.86			
	O_3	2	126 225	π_1	0.38	0.046	0.092	3	3	1
				μ_1	40.37	1.67	3.67			
				μ_2	67.83	1.057	2.19			
				σ_1	10.34	1.062	2.00			
				σ_2	10.70	0.69	0.97			
	PM_{10}	3	143 161 7	π_1	0.39	–	–	4	4	2
				π_2	0.56	–	–			
				$\boldsymbol{\pi_3}$	0.045	0.015	–			
				μ_1	8.41	0.52	–			
				μ_2	20.40	–	–			
				μ_3	50.073	9.26	–			
				σ_1	3.61	0.19	–			
				σ_2	8.51	–	–			
				σ_3	21.52	5.22	–			
	SO_2	2	30 281	$\boldsymbol{\pi_1}$	0.083	0.018	0.13	5	5	2
				π_2	0.92	0.055	0.13			
				μ_1	0.083	–	0.55			
				μ_2	3.13	0.10	0.34			
				σ_1	0.071	–	0.35			
				σ_2	1.67	0.072	0.13			

Tabla A.6: Dos Hermanas.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Dos	CO	2	157 163	π_1	0.57	0.059	0.069	6	6	1
				μ_1	430.70	15.74	22.86			
				μ_2	519.01	9.033	11.44			
				σ_1	139.17	11.53	13.69			
				σ_2	65.83	8.50	11.57			
	NO_2	2	203 145	π_1	0.51	0.037	0.081	4	4	1
				μ_1	15.11	0.57	0.77			
				μ_2	25.13	0.51	1.22			
				σ_1	3.73	0.35	0.43			
				σ_2	7.31	0.51	0.42			
	O_3	2	161 188	π_1	0.46	0.041	0.12	3	3	1
				μ_1	40.72	1.52	5.98			
				μ_2	74.41	1.11	3.65			
				σ_1	14.67	1.082	3.01			
				σ_2	12.13	0.81	1.81			
	SO_2	1	344	μ	5.79	–	0.058	5	5	1
				σ	1.12	–	0.039			

Tabla A.7: Príncipes.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Pri	CO	2	339 3	π_1	0.99	0.054	–	2	2	2
				$\boldsymbol{\pi_2}$	0.0088	0.0051	–			
				μ_1	404.42	7.19	–			
				μ_2	1311.85	–	–			
				σ_1	132.34	5.083	–			
				σ_2	80.94	33.05	–			
	NO_2	2	56 260	π_1	0.15	0.0083	0.14	2	2	1
				μ_1	12.78	2.01	3.86			
				μ_2	33.09	0.50	2.64			
				σ_1	3.61	0.88	2.22			
				σ_2	10.95	0.29	1.03			
	PM_{10}	2	338 17	π_1	0.91	0.061	0.17	3	3	2
				$\boldsymbol{\pi_2}$	0.091	0.030	0.17			
				μ_1	26.59	0.72	2.38			
				μ_2	52.79	6.29	14.17			
				σ_1	10.53	0.54	1.45			
				σ_2	21.14	3.66	4.75			
	SO_2	2	108 243	π_1	0.27	0.040	0.043	6	6	1
				μ_1	3.44	0.095	0.14			
				μ_2	5.83	0.13	0.11			
				σ_1	0.56	0.063	0.083			
				σ_2	1.42	0.082	0.076			

Tabla A.8: Ranilla.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Ran	CO	3	165 130 66	π_1	0.40	0.043	0.050	6	6	3
				π_2	0.39	0.046	0.043			
				μ_1	126.81	1.61	2.75			
				μ_2	188.45	5.41	9.077			
				μ_3	424.78	24.54	26.10			
				σ_1	14.05	1.19	1.81			
				σ_2	41.51	3.93	4.46			
				σ_3	176.028	14.016	19.37			
	NO_2	2	115 244	π_1	0.29	0.047	0.12	3	3	2
				μ_1	24.080	1.47	2.97			
				μ_2	42.53	1.16	2.15			
				σ_1	6.27	0.86	1.77			
				σ_2	12.17	0.80	0.73			
	SO_2	3	166 135 57	π_1	0.42	0.057	0.095	3	3	1
				π_2	0.37	0.052	0.12			
				μ_1	4.33	0.057	0.11			
				μ_2	5.71	0.13	0.23			
				μ_3	7.78	0.45	0.47			
				σ_1	0.47	0.040	0.088			
				σ_2	0.70	0.089	0.23			
				σ_3	1.76	0.22	0.21			

Tabla A.9: San Jerónimo.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Saj	NO_2	2	225 133	π_1	0.52	0.075	0.076	3	3	1
				μ_1	18.41	0.64	1.00			
				μ_2	27.55	1.61	1.15			
				σ_1	5.54	0.40	0.43			
				σ_2	8.42	0.79	0.68			
	O_3	2	131 231	π_1	0.36	0.043	0.094	3	3	1
				μ_1	28.83	1.75	4.42			
				μ_2	64.84	1.32	3.05			
				σ_1	10.91	1.13	2.44			
				σ_2	13.01	0.88	1.62			

Tabla A.10: Santa Clara.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Sac	CO	2	160 198	π_1	0.39	0.039	0.039	3	3	2
				μ_1	297.38	2.16	2.77			
				μ_2	406.64	6.45	6.33			
				σ_1	19.95	1.83	2.55			
				σ_2	76.034	3.99	3.93			
	NO_2	2	145 206	π_1	0.38	0.050	0.12	3	3	1
				μ_1	11.47	0.92	2.38			
				μ_2	26.066	1.015	1.77			
				σ_1	5.82	0.59	1.40			
				σ_2	9.41	0.69	0.59			
	O_3	2	78 273	π_1	0.20	0.025	0.10	3	3	1
				μ_1	19.55	1.22	4.68			
				μ_2	56.70	1.13	3.57			
				σ_1	5.14	0.68	3.045			
				σ_2	17.40	0.76	1.56			
	PM_{10}	2	318 39	π_1	0.82	0.070	0.084	3	3	1
				μ_1	21.92	0.74	1.14			
				μ_2	41.50	4.052	5.49			
				σ_1	9.26	0.54	0.64			
				σ_2	16.87	2.26	1.78			

Tabla A.11: Sierra Norte.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Sie	NO_2	2	267 82	π_1	0.73	0.063	0.065	5	5	2
				μ_1	2.88	0.085	0.14			
				μ_2	6.59	0.43	0.65			
				σ_1	0.97	0.062	0.089			
				σ_2	1.85	0.23	0.32			
	O_3	2	160 189	π_1	0.49	0.038	0.090	4	4	1
				μ_1	48.74	1.00	4.19			
				μ_2	75.69	0.84	2.24			
				σ_1	13.40	1.045	1.91			
				σ_2	9.72	0.65	1.10			
	PM_{10}	3	156 173 9	π_1	0.39	0.083	–	8	8	2
				π_2	0.56	0.089	–			
				$\boldsymbol{\pi_3}$	0.049	0.018	–			
				μ_1	10.54	0.55	–			
				μ_2	22.81	1.39	–			
				μ_3	58.32	10.73	–			
				σ_1	3.64	0.40	–			
				σ_2	8.95	0.86	–			
				σ_3	35.08	7.035	–			
	SO_2	3	121 73 153	π_1	0.36	0.033	0.032	3	3	2
				π_2	0.20	0.029	0.073			
				μ_1	1.88	0.061	0.060			
				μ_2	3.42	0.12	0.17			
				μ_3	4.67	0.064	0.11			
				σ_1	0.54	0.049	0.064			
				σ_2	0.37	0.058	0.15			
				σ_3	0.48	0.042	0.056			

Tabla A.12: Torneo.

Sitio	Contaminante	K_{BIC}	n_k	θ_k	valor $\hat{\theta}_k$	e.e. SEM	$\widehat{se_B}$	K_{AIC}	K_{AIC_c}	K_{ICL}
Tor	CO	2	218	π_1	0.53	0.040	0.054	2	2	1
				μ_1	339.62	7.80	10.037			
			147	μ_2	615.76	14.43	31.96			
				σ_1	87.37	5.72	7.041			
				σ_2	172.71	13.78	17.29			
	NO_2	1	365	μ	33.63	–	0.78	3	3	1
				σ	15.37	–	0.51			
	O_3	2	101	π_1	0.27	0.034	0.065	3	3	2
				μ_1	16.83	0.99	2.16			
			262	μ_2	47.24	1.065	2.034			
				σ_1	5.41	0.66	1.39			
				σ_2	13.078	0.77	1.096			
	PM_{10}	3	81	π_1	0.22	0.0099	0.064	3	3	1
				π_2	0.63	0.041	0.14			
				π_3	0.15	0.045	0.12			
			205	μ_1	17.16	0.78	1.24			
				μ_2	30.35	0.38	1.71			
			27	μ_3	45.20	3.75	9.55			
				σ_1	3.71	0.52	0.76			
				σ_2	7.93	0.84	1.53			
				σ_3	15.21	2.60	3.25			
	SO_2	2	319	π_1	0.77	0.090	0.15	4	4	1
				μ_1	3.40	0.059	0.088			
			46	μ_2	4.56	0.26	0.47			
				σ_1	0.66	0.047	0.12			
				σ_2	1.20	0.13	0.17			

B

Anexo II

B.1. Implementación computacional

A lo largo de esta sección, se explican las funciones diseñadas en R con las que se ha implementado la modelización mediante mixturas finitas utilizando el algoritmo EM. Se acompaña un ejemplo de uso en cada una de ellas.

B.1.1. Obtención de los valores iniciales para el algoritmo EM

Dado un conjunto de datos inicial (`muestra`), la función obtiene la media y desviación típica del número de particiones (`nc`) equidistantes realizado en el conjunto de datos. Su resultado, los valores iniciales del algoritmo EM (`vi.Q`), son almacenados como una variable global, de tal forma que quedan disponibles en el entorno para ser utilizados por una siguiente función.

```
v.i<-function(muestra,nc) {

     partes<-seq(0,1,1/nc)
     intervalos<-quantile(muestra, prob=partes)
     elem<-findInterval(muestra, intervalos, all.inside=TRUE)
     medias<-desviaciones<-numeric(nc)

     for (i in 1:nc) {
     cluster<-muestra[elem==i]
     medias[i]<-mean(cluster)
     desviaciones[i]<-sd(cluster)
}

     pi<-rep(1/nc,nc)
     vi.Q<<-c(pi, medias,desviaciones)
     vi.Q
}
```

Uso:

```
set.seed(123)
cluster.1<-rnorm(100,10,3) # mu1:  10.271218, sd1: 2.738448
cluster.2<-rnorm(100,30,4) # mu2:  29.569813, sd2: 3.867946
cluster.3<-rnorm(100,50,5) # mu3:  50.602326, sd3: 4.749395

datos<-c(cluster.1, cluster.2, cluster.3)

> v.i(datos,3)
[1]  0.3333333  0.3333333  0.3333333 10.2712177 29.5523448 50.6197935  2.7384476  3.8103611  4.7176350
```

La secuencia de valores obtenidos se corresponden, para la mixtura de 3 componentes creada, con los de $\hat{\Psi}^{(0)} = \{\pi_1^{(0)}, \pi_2^{(0)}, \pi_3^{(0)}, \mu_1^{(0)}, \mu_2^{(0)}, \mu_3^{(0)}, \sigma_1^{(0)}, \sigma_2^{(0)}, \sigma_3^{(0)}\}$.

B.1.2. Función de log-verosimilitud

`logLik` calcula el valor de la función de log-verosimilitud, dados un conjunto de datos (`muestra`) y los valores de los estimadores de los parámetros de la mixtura (`r.EM`). Los objetos `ind.1` y `ind.2` representan las posiciones en $\hat{\Psi}^{(t)}$ de las medias (`ind.1`) y las desviaciones típicas (`ind.2`). Reproduce el cálculo de la expresión (2.10).

```
logLik<-function(muestra, r.EM) {

        nc<-length(r.EM)/3
        ind.1<-seq(nc+1,3*nc)
        ind.2<-ind.1+nc
        f<-length(muestra)
        pesos<-r.EM[1:nc]
        m<-matrix(NA,f,nc)

        for (i in 1:nc){
        m[,i]<-dnorm(muestra,r.EM[ind.1[i]],r.EM[ind.2[i]])
        }

        resultado<-sum(apply(m%*%pesos,2,log))
        resultado
}
```

Uso:

```
> logLik(datos, vi.Q)
[1] -1139.322
```

B.1.3. Criterios de información

- Bayesiano (BIC)

  Calcula el valor del estadístico BIC a partir de los argumentos explicados en las funciones `v.i` y `logLik`. Utiliza el número de parámetros independientes de la mixtura (`n.par.indep`).

  ```
  BIC<-function(muestra, r.EM){

       n.par.indep<-length(r.EM)-1
       2*logLik(muestra, r.EM) - (n.par.indep*log(length(muestra)))
  }
  ```

  Uso:

  ```
  > BIC(datos, vi.Q)
  [1] -2324.273
  ```

- Akaike (AIC)

  Calcula el valor del estadístico AIC.

  ```
  AIC<-function(muestra, r.EM){

      n.par.indep<-length(r.EM)-1
      -2*logLik(muestra, r.EM) + (2*n.par.indep)
  }
  ```

  Uso:

  ```
  > AIC(datos, vi.Q)
  [1] 2294.643
  ```

- Akaike corregido (AIC_c)

  Calcula el valor del estadístico AICc.

```
AIC.c<-function(muestra, r.EM){

       n.par.indep<-length(r.EM)-1
       n<-length(muestra)
       AIC<-(-2*logLik(muestra, r.EM) + (2*n.par.indep))
       num_penalizacion<-2*n.par.indep+1*(n.par.indep+1)
       dem_penalizacion<-(n-n.par.indep-1)
       penalizacion<-num_penalizacion/dem_penalizacion
       AIC+penalizacion
}
```

Uso:

```
> AIC.c(datos, vi.Q)
[1] 2294.729
```

- Integrated classification likelihood (ICL)

  Calcula el valor del estadístico ICL, utilizando un objeto array (`m`) en donde se almacena toda la información necesaria. En la primera capa, las densidades (`m[,i,1]`); en la segunda, las *responsabilites* (`m[,,2]`), y en la tercera, el clúster al que pertenece cada observación (`m[i,1,3]`) y sus probabilidades de asignación (`m[i,2,3]`). El objeto `penalty` almacena la penalización del criterio BIC.

```
ICL<-function(data, param) {

     nc<-length(param)/3
     ind.1<-seq(nc+1, 2*nc)
     ind.2<-ind.1+nc
     pesos<-param[-c(ind.1,ind.2)]
     f<-length(data)
     m<-array(NA,dim=c(f,nc,3))

     for (i in 1:nc){
     m[,i,1]<-pesos[i]*dnorm(data,param[ind.1[i]],param[ind.2[i]])
     }

     m[,,2]<-m[,,1]/apply(m[,,1],1,sum)

     for (i in 1:f){
     m[i,1,3]<-which.max(m[i,,2])
     m[i,2,3]<-m[i,,2][which.max(m[i,,2])]
     }

     # m<<-m

     penalty<-0

     for (i in 1:nc){
     termino<-subset(m[,,3], m[,1,3]==i, select=2)
     termino<-as.vector(termino)
     penalty<-penalty+sum(log(termino))
     }

     penalty<<-penalty
     resultado<-BIC(data,param)+2*penalty
     return(resultado)
}
```

Uso:

```
> ICL(datos,vi.Q)
[1] -2327.473
```

B.1.4. Desarrollo de una iteración del algoritmo EM

El propósito de la siguiente función es su anidamiento con la función principal del algoritmo EM (siguiente función `EM`). En el array `m` se almacenan todos los cálculos intermedios. `iter` calcula una iteración del algoritmo, dados los datos de entrada (`data`) y $\hat{\Psi}^{(t)}$ (`param`). En este caso, el ejemplo que se muestra corresponde a la obtención de $\hat{\Psi}^{(1)}$, dado que `param`= $\hat{\Psi}^{(0)}$. En la línea de código 12, se calcula la expresión (2.17); en la 13, la (2.25); (2.28) en la 14, y (2.29) en la 18. Esta función está diseñada para efectuar el cálculo de la primera iteración sobre una mixtura de cualquier g.

```
iter<-function(data, param) {                                                    1

      nc<-length(param)/3                                                        2
      ind.1<-seq(nc+1, 2*nc)                                                     3
      ind.2<-ind.1+nc                                                            4
      pesos<-param[-c(ind.1,ind.2)]                                              5
      f<-length(data)                                                            6
      m<-array(NA,dim=c(f,nc,3))                                                 7

      for (i in 1:nc){                                                           8
      m[,i,1]<-pesos[i]*dnorm(data,param[ind.1[i]],param[ind.2[i]])              9
      }                                                                          10

     den<-apply(m[,,1],1,sum)                                                    11

     m[,,2]<-m[,,1]/den # Bayes                                                  12

     param[-c(ind.1,ind.2)]<-apply(m[,,2],2,mean)                                13
     param[ind.1]<-apply(m[,,2]*data,2,sum)/apply(m[,,2],2,sum)                  14

     m[,,3]<-outer(data,param[ind.1],"-")                                        15
     m[,,3]<-m[,,3]^2                                                            16

     param[ind.2]<-apply(m[,,2]*m[,,3],2,sum)/apply(m[,,2],2,sum)                17
     param[ind.2]<-sqrt(param[ind.2]) # sd                                       18

     param                                                                       19
}                                                                                20
```

Uso:

```
> iter(datos,vi.Q)
[1]  0.3330608  0.3270753  0.3398639 10.2664917 29.3481776 50.4006326  2.7208532  3.5833521  4.9106148
```

La secuencia de valores obtenidos se corresponden, para la mixtura de 3 componentes creada y en esta primera iteración, con los de $\hat{\Psi}^{(1)} = \{\pi_1^{(1)}, \pi_2^{(1)}, \pi_3^{(1)}, \mu_1^{(1)}, \mu_2^{(1)}, \mu_3^{(1)}, \sigma_1^{(1)}, \sigma_2^{(1)}, \sigma_3^{(1)}\}$.

B.1.5. Algoritmo EM

`EM` efectúa el desarrollo iterativo propiamente dicho del algoritmo a partir de un conjunto de datos (`datos`), $\Psi^{(t)}$ (`vi`), y un valor de ϵ prestablecido (`eps`). En su implementación incorpora a las funciones `logLik` y `iter` anteriores. `err` almacena el cálculo de la expresión (2.30), por lo que se le asigna inicialmente el valor 1.

Los objetos `n.param` se corresponden con el número de parámetros de la mixtura; `c`, con el número de componentes, e `iteraciones` almacena todas las iteraciones del algoritmo hasta su convergencia final, necesarias, por una parte, para el cálculo posterior de los errores SEM, y por otra, para evaluar la convergencia del algoritmo si así se requiriese. El resultado final de las estimaciones de los parámetros de la mixtura y el número de iteraciones + 1 ocurridas se devuelve en un formato de lista.

```
EM<-function(datos,vi, eps=1/1000000){                                           1
```

```
    n.param<-length(vi)                                                                                    2
    iteraciones<<-matrix(rep(NA,n.param), nrow=1)                                                          3
    c<-n.param/3                                                                                           4

    nuevos.p<-vi                                                                                           5
    err<-1                                                                                                 6
    iter<-1                                                                                                7

    while(err>eps) {                                                                                       8

    vi<-iter(data=datos,param=vi)                                                                          9
    antig.p<-nuevos.p                                                                                      10
    nuevos.p<-vi                                                                                           11

    iteraciones<<-rbind(iteraciones, nuevos.p)                                                             12

    err<-abs((logLik(datos, antig.p)-logLik(datos, nuevos.p))/logLik(datos, nuevos.p))                     13

    iter<-iter+1                                                                                           14

    }                                                                                                      15

    r.EM<<-vi                                                                                              16

    return (list(componentes=c,p.EM=c(vi[1:c]),                                                            17
            mu.EM=c(vi[(c+1):(2*c)]),                                                                      18
            sd.EM=c(vi[(2*c+1):(3*c)]), iter=iter))                                                        19
}                                                                                                          20
```

El blucle while se interrumpe (líneas 8 a 15) cuando, a lo largo de las iteraciones, la diferencia absoluta relativa (`err`) entre la función de log-verosimilitud parametrizada según `antig.p` y `nuevos.p` es mayor que la tolerancia asumida (ϵ), devolviendo en ese caso el resultado de $\hat{\Psi}^{(6)}$, indicando mediante "†" en el siguiente ejemplo.

Uso:

```
> EM(datos,vi.Q)
$componentes
[1] 3

$p.EM
[1] 0.3332530 0.3196722 0.3470748

$mu.EM
[1] 10.26989 29.16179 50.14222

$sd.EM
[1] 2.723816 3.354494 5.179383

$iter
[1] 7
```

```
> iteraciones
              [,1]      [,2]      [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]
                NA        NA        NA       NA       NA       NA       NA       NA       NA
nuevos.p 0.3330608 0.3270753 0.3398639 10.26649 29.34818 50.40063 2.720853 3.583352 4.910615
nuevos.p 0.3331725 0.3237826 0.3430449 10.26844 29.26456 50.28866 2.722516 3.475382 5.026074
nuevos.p 0.3332177 0.3218843 0.3448981 10.26925 29.21704 50.22175 2.723239 3.418273 5.095954
nuevos.p 0.3332373 0.3207569 0.3460058 10.26960 29.18882 50.18140 2.723558 3.385458 5.138234
nuevos.p 0.3332474 0.3200800 0.3466726 10.26978 29.17193 50.15699 2.723724 3.366066 5.163858
nuevos.p 0.3332530 0.3196722 0.3470748 10.26989 29.16179 50.14222 2.723816 3.354494 5.179383  †
```

Para la validación de resultados obtenidos de esta implementación, se efectuó su comparación con los de la función Mclust, sin obtener diferencias significativas al respecto:

Comparación de los resultados de la implementación con Mclust, con g especificado (`G=3`):

```
require(mclust)
modelo<-Mclust(datos, modelNames="V", G=3)
```

```
summary(modelo, parameters=TRUE)

----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust V (univariate, unequal variance) model with 3 components:

 log.likelihood   n df       BIC       ICL
      -1137.624 300  8 -2320.878 -2326.321

Clustering table:
  1   2   3
100  96 104

Mixing probabilities:
        1         2         3
0.3332530 0.3196725 0.3470745

Means:
       1        2        3
10.26989 29.16180 50.14223

Variances:
        1         2         3
 7.419175 11.252671 26.825923
```

B.1.6. Obtención de los errores de $\hat{\Psi}$ mediante *bootstrap*

Con esta función se obtienen los valores $\hat{\Psi}$ a través del procedimiento descrito en la sección 2.5.2., con $B = 1000$, así como sus errores. Utiliza la función de R, sample, para la generación de las muestras bootstrap, y las funciones `v.i` y `EM` descritas con anterioridad. Todos los resultados de estimación de $\hat{\Psi}$ sobre cada muestra bootstrap se almacenan en la matriz `m`.

```
errores.boots<-function(sample, nc, rep){

               num.param<-3*nc
               m<-matrix(NA,nrow=rep, ncol=num.param, byrow=TRUE)

               for (i in 1:rep){
               muestra.boot<-sample(sample,replace=TRUE)
               valores.ini<-v.i(muestra.boot,nc)
               EM(muestra.boot, valores.ini)
               m[i,]<-r.EM
               }

               param.medios<-apply(m,2,mean)
               errores.medios<-apply(m,2,sd)

               return (list(componentes=nc,pis.boot=c(param.medios[1:nc]),
                       mus.boot=c(param.medios[(nc+1):(2*nc)]),
                       sigmas.boot=c(param.medios[(2*nc+1):(3*nc)]),
                       errores.pis.boot=c(errores.medios[1:nc]),
                       errores.mus.boot=c(errores.medios[(nc+1):(2*nc)]),
                       errores.sigmas.boot=c(errores.medios[(2*nc+1):(3*nc)])))
}
```

Uso:

```
> errores.boots(sample=datos,nc=3,rep=1000)
$componentes
[1] 3

$pis.boot
[1] 0.3344449 0.3193336 0.3462215

$mus.boot
```

```
[1] 10.26426 29.16339 50.16348

$sigmas.boot
[1] 2.701940 3.336825 5.114673

$errores.pis.boot
[1] 0.02737678 0.02762136 0.02815098

$errores.mus.boot
[1] 0.2717807 0.3765336 0.5633902

$errores.sigmas.boot
[1] 0.1888861 0.2829260 0.4245305
```

Los nombres de los componentes de la lista que muestran los errores bootstrap son `errores.pis.boot`, `errores.mus.boot` y `errores.sigmas.boot`, en referencia a $\hat{\pi}_i$, $\hat{\mu}_i$ y $\hat{\sigma}_i$, con $i = 1, \ldots, 3$. Compárense los resultados de $\hat{\Psi}$ (`pis.boot`, `mus.boot`, `sigmas.boot`) con los obtenidos mediante la implementación del algoritmo EM (función `EM`).

B.1.7. Obtención de los errores de $\hat{\Psi}$ mediante el método SEM

La función final que los calcula, `SEM`, precisa de la utilización de tres funciones previas: `param_ik`, `r` y `stop`. `r` anida a `param_ik` y `stop` a `r`, por lo que el tiempo computacional empleado en la obtención de estos errores es superior a todas las funciones empleadas hasta ahora.

- `param_ik`

  A partir del historial de iteraciones del algoritmo, obtiene la expresion (2.32). Como se aprecia en el ejemplo de uso, únicamente se considera la evolución de uno solo de los estimadores de $\hat{\Psi}$, en este caso $\hat{\mu}_1$, permaneciendo el resto fijos. La línea 4 del código obtiene el valor de $\hat{\Psi}$ a partir del historial de iteraciones `iteraciones`.

```
param_ik<-function(i) {                                                          1

          n.filas<-nrow(iteraciones)                                             2
          n.colum<-ncol(iteraciones)                                             3
          resultados.iteraciones<-iteraciones[n.filas,]                          4

          matriz<-iteraciones                                                    5

          matriz.ficticia<-matrix(0, n.filas, n.colum)                           6

          for(j in 1:n.filas) {                                                  7
          matriz.ficticia[j,i]<-matriz[j,i]                                      8
          matriz.ficticia[j,-i]<-resultados.iteraciones[-i]                      9
          }                                                                      10

          return(matriz.ficticia)                                                11
}                                                                                12
```

  Uso:

```
> param_ik(4)
         [,1]      [,2]      [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]
[1,] 0.333253 0.3196722 0.3470748 10.26649 29.16179 50.14222 2.723816 3.354494 5.179383
[2,] 0.333253 0.3196722 0.3470748 10.26844 29.16179 50.14222 2.723816 3.354494 5.179383
[3,] 0.333253 0.3196722 0.3470748 10.26925 29.16179 50.14222 2.723816 3.354494 5.179383
[4,] 0.333253 0.3196722 0.3470748 10.26960 29.16179 50.14222 2.723816 3.354494 5.179383
[5,] 0.333253 0.3196722 0.3470748 10.26978 29.16179 50.14222 2.723816 3.354494 5.179383
[6,] 0.333253 0.3196722 0.3470748 10.26989 29.16179 50.14222 2.723816 3.354494 5.179383
```

- `r`

  Obtiene el valor de la expresión (2.33). En el ejemplo de uso, se obtiene el valor de $r_{3,2}$.

```
r<-function(i,j, muestra) {

   matriz.ficticia<-param_ik(i)
   n.filas<-nrow(matriz.ficticia)-1
   convergencias<-numeric(n.filas)

   for (k in 1:n.filas){
   num<-iter(data=muestra,matriz.ficticia[k,])[j]-(r.EM)[j]
   den<-matriz.ficticia[k,i]-(r.EM[i])
   convergencias[k]<-num/den
   }

   return(convergencias)
}
```

Uso:

```
> r(3,2, datos)
[1] 0.02005329 0.04702889 0.09900547 0.21606495 0.59737938
```

- `stop`

  Representa una modificación del método SEM, ya que obtiene el valor de r_{ij} en lugar de como se propone en (2.34), como la menor diferencia entre los valores de la secuencia $r_{ij}^{(k)}, r_{ij}^{(k+1)}, r_{ij}^{(k+2)}, \ldots$. Así, se consigue no tener que aplicar diferentes ϵ según los pares de valores i, j de $r_{i,j}$ para obtener la convergencia de $r_{ij}^{(k)}$. En historiales de iteración reducidos del algoritmo EM, como el que se plantea en el ejemplo, con tan solo 6 iteraciones, si el ϵ propuesto es muy laxo, puede no obtenerse ningún $r_{i,j}^{(k)}$ por ausencia de convergencia.

```
stop<-function(i,j,muestra) {

      conv<-r(i,j, muestra)
      elegido<-which(abs(diff(conv))==min(abs(diff(conv))))
      conv[elegido+1]
}
```

Uso:

```
> stop(3,2, datos)
[1] 0.04702889
```

- `SEM`

  Esta función conjuga las anteriormente descritas a través de la función `stop` (línea 30 del código) para la obtención de la matriz DM, implementando el resto de cálculos necesarios de la expresión (2.31). Los elementos de la matriz I_{oc} cuadrada (línea 15) se insertan a través de sus índices mediante el bucle de las líneas 16 a 23.

```
SEM<-function(muestra, param) {                                                    1

     nc<-length(param)/3                                                           2
     nparam<-length(param)                                                         3
     ind.1<-seq(nc+1, 2*nc)                                                        4
     ind.2<-ind.1+nc                                                               5
     pesos<-param[-c(ind.1,ind.2)]                                                 6
     f<-length(muestra)                                                            7
     m<-array(NA,dim=c(f,nc,2))                                                    8

     for (i in 1:nc){                                                              9
     m[,i,1]<-pesos[i]*dnorm(muestra,param[ind.1[i]],param[ind.2[i]])              10
     }                                                                             11

     denominador<-apply(m[,,1],1,sum)                                              12
     m[,,2]<-m[,,1]/denominador                                                    13
     suma.Ez<-apply(m[,,2],2,sum)                                                  14
```

```
        Ioc<-matrix(0,nparam,nparam)                                                                    15

        for (i in 1:nc){                                                                                16
           Ioc[i,i]<-(1/param[i]^2)*suma.Ez[i]                                                          17
           Ioc[i+nc,i+nc]<-(1/param[i+2*nc]^2)*suma.Ez[i]                                               18
           Ioc[i+2*nc,i+2*nc]<-((-1/(param[i+2*nc]^2))*suma.Ez[i]) +                                    19
           ((1/param[i+2*nc]^4)*(sum(m[,i,2]*3*(muestra-param[i+nc])^2)))                               20

           Ioc[i+nc,i+2*nc]<-(2/param[i+2*nc]^3)*(sum(m[,i,2]*(muestra-param[i+nc])))                   21
           Ioc[i+2*nc,i+nc]<-(2/param[i+2*nc]^3)*(sum(m[,i,2]*(muestra-param[i+nc])))                   22
           }                                                                                            23

           v.i(muestra,nc)                                                                              24
           EM(muestra,vi.Q)                                                                             25
           iteraciones<<-iteraciones[-1,]                                                               26

           dm<-matrix(0,nparam,nparam)                                                                  27

           for(i in 1:nparam) {                                                                         28
                   for (j in 1:nparam) {                                                                29
                   dm[i,j]<-stop(i,j, muestra)                                                          30
           }                                                                                            31
           }                                                                                            32

           identidad<-matrix(0,nparam,nparam)                                                           33
           diag(identidad)<-1                                                                           34

           Ioc.inv<-solve(Ioc)                                                                          35
           inv.dm<-solve(identidad-dm)                                                                  36
           AV<-Ioc.inv%*%dm%*%inv.dm                                                                    37
           V.COV<-Ioc.inv+AV                                                                            38
           errores.SEM<-sqrt(diag(V.COV))                                                               39

           return(errores.SEM)                                                                          40
}                                                                                                       41
```

Uso:

```
> SEM(datos, r.EM)
[1] 0.03254683 0.03180168 0.03313496 0.26613849 0.35582356 0.51770878 0.18846662 0.25985883 0.39235493
```

Estos resultados se corresponden con los errores de $\hat{\pi}_i$, $\hat{\mu}_i$ y $\hat{\sigma}_i$, con $i = 1, \ldots, 3$. Compárense con los obtenidos mediante el método bootstrap (función `errores.boot`) y sus resultados `errores.pis.boot`, `errores.mus.boot` y `errores.sigmas.boot`.

B.1.8. Obtención de la incertidumbre de asignación de y_j

La función `cuantiles` determina las incertidumbres de pertenencia de la observación muestral $(y_j, i = 1, \ldots, n)$ a su componente asignada de la mixtura, según las expresiones (2.17) y (2.18). Como resultado, devuelve en un conjunto de g listas (línea 2 del código), el número de observaciones asignadas a cada componente (`$n_clusters`) y los cuantiles de las incertidumbres (`$cuantiles_de_incertidumbres`) en cada una de ellas. Como en otras funciones anteriores, toda la información intermedia de los cálculos se almacena en un objeto array (`m`, línea 8).

```
cuantiles<-function(data, param) {                                                                      1

        probs.en.clusters<-vector("list")                                                               2

        nc<-length(param)/3                                                                             3
        ind.1<-seq(nc+1, 2*nc)                                                                          4
        ind.2<-ind.1+nc                                                                                 5
        pesos<-param[-c(ind.1,ind.2)]                                                                   6
        f<-length(data)                                                                                 7
        m<-array(NA,dim=c(f,nc,3))                                                                      8
```

```
          for (i in 1:nc){
          m[,i,1]<-pesos[i]*dnorm(data,param[ind.1[i]],param[ind.2[i]])
          }

          m[,,2]<-m[,,1]/apply(m[,,1],1,sum)

          for (i in 1:f){
          m[i,1,3]<-which.max(m[i,,2])
          m[i,2,3]<-m[i,,2][which.max(m[i,,2])]
           }

         #m<<-m

         for (i in 1:nc){
         incertidumbres<-1-subset(m[,,3], m[,1,3]==i, select=2)
         incertidumbres<-as.vector(incertidumbres)
         probs.en.clusters[[i]]<-list(n_clusters=length(incertidumbres),
                                cuantiles_de_incertidumbres=quantile(incertidumbres))
         }

         return(probs.en.clusters)
}
```

Uso:

```
> cuantiles(datos,r.EM)
[[1]]
[[1]]$n_clusters
[1] 100

[[1]]$cuantiles_de_incertidumbres
          0%          25%          50%          75%         100%
1.287859e-14 5.167107e-10 1.299451e-08 4.968867e-07 2.697498e-03


[[2]]
[[2]]$n_clusters
[1] 96

[[2]]$cuantiles_de_incertidumbres
          0%          25%          50%          75%         100%
4.392749e-06 1.315026e-05 6.921459e-05 5.909122e-04 4.710164e-01


[[3]]
[[3]]$n_clusters
[1] 104

[[3]]$cuantiles_de_incertidumbres
          0%          25%          50%          75%         100%
0.000000e+00 5.354051e-14 2.349214e-10 7.465486e-08 3.998059e-01
```

B.1.9. Cálculo de los momentos de la mixtura

La función `momentos` calcula los momentos de la mixtura, segun las expresiones (2.7).

```
momentos<-function(parametros){

        nc<-length(parametros)/3
        ind.1<-seq(nc+1,3*nc)
        ind.2<-ind.1+nc
        pesos<-r.EM[1:nc]

        mu<-0
        sigma.0<-0

        for (i in 1:nc){
```

```
        mu<-mu+pesos[i]*r.EM[ind.1[i]]
        sigma.0<- sigma.0 + (pesos[i]*(r.EM[ind.1[i]]^2+r.EM[ind.2[i]]^2))
        }

        sigma<-sigma.0-mu^2

        lista<-list("mu mixtura"=mu, "sigma mixtura"=sqrt(sigma))

        return(lista)
}
```

Uso:

```
> momentos(r.EM)
$‘mu mixtura‘
[1] 29.89482

$‘sigma mixtura‘
[1] 16.81013
```

El argumento `r.EM` representa a $\hat{\Psi}$.

B.1.10. Selección del mejor modelo

Una vez que todas las funciones anteriores han sido descritas, la última etapa del proceso de modelización consiste en seleccionar aquella mixtura cuyo número de componentes mejor describa los datos experimentales en estudio, sobre un número de componentes prefijados ($K = 1, \ldots, 11$). La siguiente función `modelos` calcula el modelo más apropiado para ese conjunto de datos, pero también genera un resumen de dicha modelización resultante. En este resumen, se incluye, entre otros resultados, la comparación de la implementación del algoritmo EM con los resultados de la función Mclust (componente del objeto lista `$resultados.Mclust`), esta vez sin prefijar en esa función el número de componentes de la mixtura (argumento G ausente).

Se sintetiza a continuación el contenido de la función:

- Líneas 2-3: resultados de la función Mclust a efectos de comparación.
- Líneas 4-7: vectores numéricos donde almacenar los resultados de los criterios de información según las $K = 11$ modelos propuestos.
- Líneas 10-16: caso particular de mixtura $K = 1$.
- Líneas 17-24: obtención del valor de los criterios de información para $K = 2, \ldots, 11$ según todas las modelizaciones realizadas, que se almacenan en una lista (`resultados.EM[[i]]`).
- Líneas 26-29: obtención del mejor modelo según el valor de los criterios de información para $K = 2, \ldots, 11$.
- Línea 30: obtención del mejor K según el valor del criterio de información BIC para $K = 2, \ldots, 11$.
- Línea 33: obtención del mejor modelo según K, obtenido mediante el valor del criterio de información BIC.
- Líneas 34-37: operación sobre el mejor modelo utilizando funciones descritas en esta sección.
- Líneas 38-42: devolución de resultados en forma de lista.

```
modelos<-function(muestra){                                                    1

        modelo.mclust<-Mclust(muestra, modelNames="V")                         2
        res.mclust<-summary(modelo.mclust, parameters=TRUE)                    3

        bic<-numeric(11)                                                       4
        icl<-numeric(11)                                                       5
        aic<-numeric(11)                                                       6
        aic.c<-numeric(11)                                                     7
```

```
        resultados.EM<-vector("list")

        # K=1

        media<-mean(muestra) ; desv.tip<-sd(muestra)
        res.1N<-c(1,media,desv.tip) # se especifica peso unico 1

        resultados.EM[[1]]<-res.1N
        bic[1]<-BIC(muestra,res.1N)
        aic[1]<-AIC(muestra,res.1N)
        aic.c[1]<-AIC.c(muestra,res.1N)
        icl[1]<-bic[1]

        for (i in seq(2,11)) {

                        v.i(muestra,i)
                        resultados.EM[[i]]<-EM(muestra,vi.Q)
                        bic[i]<-BIC(muestra,r.EM)
                        aic[i]<-AIC(muestra,r.EM)
                        aic.c[i]<-AIC.c(muestra,r.EM)
                        icl[i]<-ICL(muestra,r.EM)
        }

       # seleccion mediante BIC

       modelo.bic<-which.max(bic)
       modelo.icl<-which.max(icl)
       modelo.aic<-which.min(aic)
       modelo.aic.c<-which.min(aic)

       optimo.bic<-bic[modelo.bic]
       icl.deducido<-icl[modelo.bic]

       eleccion.EM<-resultados.EM[[modelo.bic]]

       recuperamos.EM<-c(eleccion.EM$p.EM,eleccion.EM$mu.EM,eleccion.EM$sd.EM)

       incertidumbres<-cuantiles(muestra,recuperamos.EM)

       errores.sem<-SEM(muestra,recuperamos.EM)

       info.bootstrap<-errores.boots(sample=muestra,nc=modelo.bic,rep=1000)

       param.mixtura<-momentos(recuperamos.EM)

       lista<-list(n=length(muestra),bic=bic,icl=icl, aic=aic, aic.c=aic.c, componentes.bic=modelo.bic,
       componentes.icl=modelo.icl, componentes.aic=modelo.aic, componentes.aic.c=modelo.aic.c,
       optimo.bic=optimo.bic, icl.deducido=icl.deducido,resultados.em=eleccion.EM,
       resultados.Mclust=res.mclust,bootstrap=info.bootstrap,Errores.SEM=errores.sem,
       quantiles_incertidumbres=incertidumbres, parametros.mixtura=param.mixtura)

       return(lista)
}
```

Uso:

```
> modelos(datos)
$n
[1] 300

$bic
 [1] -2559.698 -2523.896 -2320.878 -2337.298 -2355.186 -2364.849 -2381.561 -2396.403 -2412.017 -2425.248 -2439.138

$icl
 [1] -2559.698 -2601.566 -2326.321 -2408.516 -2466.665 -2511.345 -2555.924 -2590.889 -2634.452 -2634.509 -2649.808

$aic
 [1] 2552.291 2505.377 2291.248 2296.557 2303.333 2301.885 2307.485 2311.216 2315.719 2317.839 2320.616

$aic.c
 [1] 2552.314 2505.431 2291.333 2296.675 2303.484 2302.069 2307.704 2311.469 2316.008 2318.165 2320.980

$componentes.bic
```

```
[1] 3

$componentes.icl
[1] 3

$componentes.aic
[1] 3

$componentes.aic.c
[1] 3

$optimo.bic
[1] -2320.878

$icl.deducido
[1] -2326.321

$resultados.em
$resultados.em$componentes
[1] 3

$resultados.em$p.EM
[1] 0.3332530 0.3196722 0.3470748

$resultados.em$mu.EM
[1] 10.26989 29.16179 50.14222

$resultados.em$sd.EM
[1] 2.723816 3.354494 5.179383

$resultados.em$iter
[1] 7


$resultados.Mclust
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust V (univariate, unequal variance) model with 3 components:

 log.likelihood   n df       BIC       ICL
      -1137.624 300  8 -2320.878 -2326.321

Clustering table:
  1   2   3
100  96 104

Mixing probabilities:
        1         2         3
0.3332530 0.3196725 0.3470745

Means:
       1        2        3
10.26989 29.16180 50.14223

Variances:
        1         2         3
 7.419175 11.252671 26.825923

$bootstrap
$bootstrap$componentes
[1] 3

$bootstrap$pis.boot
[1] 0.3331474 0.3197393 0.3471133

$bootstrap$mus.boot
[1] 10.28150 29.16102 50.14907

$bootstrap$sigmas.boot
[1] 2.702722 3.307517 5.146982

$bootstrap$errores.pis.boot
[1] 0.02634813 0.02812192 0.02717928
```

```
$bootstrap$errores.mus.boot
[1] 0.2711535 0.3919006 0.5621406

$bootstrap$errores.sigmas.boot
[1] 0.1873681 0.2792534 0.4352822


$Errores.SEM
[1] 0.03254683 0.03180168 0.03313496 0.26613849 0.35582356 0.51770878 0.18846662 0.25985883 0.39235493

$quantiles_incertidumbres
$quantiles_incertidumbres[[1]]
$quantiles_incertidumbres[[1]]$n_clusters
[1] 100

$quantiles_incertidumbres[[1]]$cuantiles_de_incertidumbres
          0%          25%          50%          75%         100%
1.874167e-12 5.725421e-09 8.768767e-08 2.255958e-06 9.604646e-03


$quantiles_incertidumbres[[2]]
$quantiles_incertidumbres[[2]]$n_clusters
[1] 96

$quantiles_incertidumbres[[2]]$cuantiles_de_incertidumbres
          0%          25%          50%          75%         100%
1.230909e-05 4.246857e-05 1.740833e-04 1.440555e-03 4.810307e-01


$quantiles_incertidumbres[[3]]
$quantiles_incertidumbres[[3]]$n_clusters
[1] 104

$quantiles_incertidumbres[[3]]$cuantiles_de_incertidumbres
          0%          25%          50%          75%         100%
0.000000e+00 5.637463e-12 7.335078e-09 1.000904e-06 4.116807e-01



$parametros.mixtura
$parametros.mixtura$‘mu mixtura‘
[1] 29.89482

$parametros.mixtura$‘sigma mixtura‘
[1] 16.81013
```

C

Anexo III

C.1. Implementaciones mediante libros existentes en R

A lo largo de esta sección, se explican las funciones utilizadas en R de libros existentes en ese entorno, con las que se han llevado a cabo los análisis complementarios a la modelización mediante mixturas. Estas funciones o libros han permitido realizar el ACJ, imputaciones de μ_m y cv_m, y el ACP.

C.1.1. ACJ

- Dendrograma según los valores de μ_m (Figura 4.2A).

```
alc.m<-c(308.57,21.06,61.12,28.84,4.82)
alj.m<-c(NA,18.44,62.86,30.53,6.78)
ber.m<-c(480.55,21.64,51.91,33.55,5.02)
cen.m<-c(677.75,21.46,53.54,NA,2.80)
cob.m<-c(206.7,6.54,55.91,17.04,2.76)
dos.m<-c(463.84,19.44,57.39,NA,5.79)
pri.m<-c(412.38,29.39,NA,29.82,5.36)
ran.m<-c(221.04,36.15,NA,NA,5.49)
saj.m<-c(NA,22.79,53.57,NA,NA)
sac.m<-c(367.1,20.87,47.94,24.74,NA)
sie.m<-c(NA,3.78,61.42,19.75,3.37)
tor.m<-c(465.83,33.63,39.25,29.72,3.67)

m<-matrix(c(alc.m,alj.m,ber.m,cen.m,cob.m,
            dos.m,pri.m,ran.m,saj.m,sac.m,
            sie.m,tor.m),ncol=5, byrow=T)

dimnames(m)<-list(estaciones=c("Alc","Alj","Ber",
             "Cen","Cob","Dos","Pri","Ran","Saj",
             "Sac","Sie","Tor"), par=c())


d<-dist(m,method="euclidean")
figura<-hclust(d,method="single")
par(mar=c(0.15,4,0.85,0.15), cex=1.5)
plot(figura, main="A", ylab="", xlab="", axes=FALSE)
axis(side=2, line=0.8)
```

- Dendrograma según los valores de cv_m (Figura 4.2B).

```
alc.sd<-c(51,10.22,20.92,13.92,1.65)
alj.sd<-c(NA,9.39,21.50,15.22,2.92)
ber.sd<-c(148.03,14.37,19.71,16.01,1.89)
cen.sd<-c(276.24,9.03,21.44,NA,1.08)
cob.sd<-c(74.56,4.34,16.92,12.29,1.82)
dos.sd<-c(123.08,7.17,21.56,NA,1.12)
pri.sd<-c(156.77,12.40,NA,13.88,1.66)
ran.sd<-c(141.94,13.09,NA,NA,1.52)
saj.sd<-c(NA,7.94,21.05,NA,NA)
sac.sd<-c(86.60,11.04,21.67,13.19,NA)
```

```
sie.sd<-c(NA,1.99,17.42,14.87,1.38)
tor.sd<-c(181.32,15.37,17.72,12.40,0.92)

alc.cv<-alc.sd/alc.m
alj.cv<-alj.sd/alj.m
ber.cv<-ber.sd/ber.m
cen.cv<-cen.sd/cen.m
cob.cv<-cob.sd/cob.m
dos.cv<-dos.sd/dos.m
pri.cv<-pri.sd/pri.m
ran.cv<-ran.sd/ran.m
saj.cv<-saj.sd/saj.m
sac.cv<-sac.sd/sac.m
sie.cv<-sie.sd/sie.m
tor.cv<-tor.sd/tor.m

cv<-matrix(c(alc.cv,alj.cv,ber.cv,cen.cv,
             cob.cv,dos.cv,pri.cv,ran.cv,
             saj.cv,sac.cv,sie.cv,tor.cv),
             ncol=5, byrow=T)

dimnames(cv)<-list(estaciones=c("Alc","Alj","Ber",
              "Cen","Cob","Dos","Pri","Ran","Saj","Sac","Sie","Tor"), par=c())


d.cv<-dist(cv,method="euclidean")
figura.cv<-hclust(d.cv,method="single")
par(mar=c(0.15,0.5,0.85,3.15), cex=1.5)
plot(figura.cv, main="B", ylab="", xlab="", axes=FALSE)
axis(side=4, line=0.8)
```

C.1.2. Imputación mediante BA

- Imputación de los valores de μ_m (Tabla 4.2).

```
Alc<-c(308.57,21.06,61.12,28.84,4.82)
Alj<-c(NA,18.44,62.86,30.53,6.78)
Ber<-c(480.55,21.64,51.91,33.55,5.02)
Cen<-c(677.75,21.46,53.54,NA,2.80)
Cob<-c(206.7,6.54,55.91,17.04,2.76)
Dos<-c(463.84,19.44,57.39,NA,5.79)
Pri<-c(412.38,29.39,NA,29.82,5.36)
Ran<-c(221.04,36.15,NA,NA,5.49)
Saj<-c(NA,22.79,53.57,NA,NA)
Sac<-c(367.1,20.87,47.94,24.74,NA)
Sie<-c(NA,3.78,61.42,19.75,3.37)
Tor<-c(465.83,33.63,39.25,29.72,3.67)

m<-matrix(c(Alc,Alj,Ber,Cen,Cob,Dos,Pri,Ran,Saj,Sac,Sie,Tor),ncol=5, byrow=T)

library(randomForest)
set.seed(222)

tipo<-c("U","S","U","U","R","U","U","U","S","S","R","U")
tipo<-factor(tipo, labels=c("R","S","U"))

marco<-data.frame(Alc,Alj,Ber,Cen,Cob,Dos,Pri,Ran,Saj,Sac,Sie,Tor)
marco<-t(marco); marco
marco<-as.data.frame(marco); marco
marco<-data.frame(marco,tipo); marco

colnames(marco)<-c("CO","NO2","O3","PM10","SO2","tipo")

marco.imp <- rfImpute(tipo ~ ., marco, iter=250, ntree=2500)

> marco.imp

    tipo       CO   NO2       O3     PM10      SO2
Alc    U 308.5700 21.06 61.12000 28.84000 4.820000
Alj    S 394.1254 18.44 62.86000 30.53000 6.780000
Ber    U 480.5500 21.64 51.91000 33.55000 5.020000
Cen    U 677.7500 21.46 53.54000 27.93820 2.800000
Cob    R 206.7000  6.54 55.91000 17.04000 2.760000
```

```
Dos    U 463.8400 19.44 57.39000 27.69083 5.790000
Pri    U 412.3800 29.39 54.68649 29.82000 5.360000
Ran    U 221.0400 36.15 54.83417 28.16964 5.490000
Saj    S 420.4997 22.79 53.57000 28.85198 4.876696
Sac    S 367.1000 20.87 47.94000 24.74000 4.563775
Sie    R 328.0027  3.78 61.42000 19.75000 3.370000
Tor    U 465.8300 33.63 39.25000 29.72000 3.670000
```

- Imputación de los valores de cv_m (Tabla 4.3).

```
alc.sd<-c(51,10.22,20.92,13.92,1.65)
alj.sd<-c(NA,9.39,21.50,15.22,2.92)
ber.sd<-c(148.03,14.37,19.71,16.01,1.89)
cen.sd<-c(276.24,9.03,21.44,NA,1.08)
cob.sd<-c(74.56,4.34,16.92,12.29,1.82)
dos.sd<-c(123.08,7.17,21.56,NA,1.12)
pri.sd<-c(156.77,12.40,NA,13.88,1.66)
ran.sd<-c(141.94,13.09,NA,NA,1.52)
saj.sd<-c(NA,7.94,21.05,NA,NA)
sac.sd<-c(86.60,11.04,21.67,13.19,NA)
sie.sd<-c(NA,1.99,17.42,14.87,1.38)
tor.sd<-c(181.32,15.37,17.72,12.40,0.92)

Alc<-alc.sd/Alc
Alj<-alj.sd/Alj
Ber<-ber.sd/Ber
Cen<-cen.sd/Cen
Cob<-cob.sd/Cob
Dos<-dos.sd/Dos
Pri<-pri.sd/Pri
Ran<-ran.sd/Ran
Saj<-saj.sd/Saj
Sac<-sac.sd/Sac
Sie<-sie.sd/Sie
Tor<-tor.sd/Tor

cv<-matrix(c(Alc,Alj,Ber,Cen,Cob,Dos,Pri,Ran,Saj,Sac,Sie,Tor),ncol=5, byrow=T)

marco<-data.frame(Alc,Alj,Ber,Cen,Cob,Dos,Pri,Ran,Saj,Sac,Sie,Tor); marco
marco<-t(marco); marco
marco<-as.data.frame(marco); marco
marco<-data.frame(marco,tipo); marco

colnames(marco)<-c("CO","NO2","O3","PM10","SO2","tipo")

set.seed(222)

tipo<-c("U","S","U","U","R","U","U","U","S","S","R","U")
tipo<-factor(tipo, labels=c("R","S","U"))

marco.imp.cv <- rfImpute(tipo ~ ., cv, iter=250, ntree=2500)

colnames(marco.imp.cv)<-c("CO","NO2","O3","PM10","SO2")
rownames(marco.imp.cv)<-c("Alc","Alj","Ber","Cen","Cob",
                          "Dos","Pri","Ran","Saj","Sac","Sie","Tor")

> marco.imp.cv
           CO       NO2        O3      PM10       SO2
Alc 0.1652785 0.4852802 0.3422775 0.4826630 0.3423237
Alj 0.3331619 0.5092191 0.3420299 0.4985260 0.4306785
Ber 0.3080429 0.6640481 0.3796956 0.4771982 0.3764940
Cen 0.4075839 0.4207829 0.4004483 0.5230816 0.3857143
Cob 0.3607160 0.6636086 0.3026292 0.7212441 0.6594203
Dos 0.2653501 0.3688272 0.3756752 0.5202061 0.1934370
Pri 0.3801591 0.4219122 0.3969689 0.4654594 0.3097015
Ran 0.6421462 0.3621024 0.3993177 0.4765102 0.2768670
Saj 0.3757580 0.3483984 0.3929438 0.4830967 0.3156961
Sac 0.2359030 0.5289890 0.4520234 0.5331447 0.3382175
Sie 0.3138394 0.5264550 0.2836210 0.7529114 0.4094955
Tor 0.3892407 0.4570324 0.4514650 0.4172275 0.2506812
```

C.1.3. ACP

- ACP. Gráficos 4.3A y 4.3B.

```
marco.imp<-marco.imp[,-1]

prcomp(marco.imp, scale=TRUE)

tipo<-c("UB","SB","UB","UB","RI","UB","UB","UT","SI","SB","RB","UT")
tipo<-factor(tipo, labels=c("RB","RI","SB","SI","UB","UT"))

row.names(marco.imp)<-c("Alc","Alj","Ber", "Cen","Cob","Dos","Pri","Ran","Saj","Sac","Sie","Tor")

# gráfico: componente 1 y 2 (eje X CP1, eje Y CP2)
plot(c(-1.5,3.2), c(-2.5,2.5), type="n", axes=FALSE, xlab=NA, ylab=NA, cex.main=2, main="A")
s.class(dfxy=acp$li, fac=tipo, xax=1, yax=2, grid=FALSE, cgrid=FALSE, cpoint=2, addaxes=FALSE,
        origin=c(0,0), add.plot=TRUE)

text(0.4951411,-1.10395170,"Alc", cex=1.3)
text(-0.2675646,-2.1002008902,"Alj", cex=1.3)
text(-1.1648300,0.13564444,"Ber", cex=1.3)
text(-0.4309082,2.13402535,"Cen", cex=1.3)
text(3.0160943,0.69518489,"Cob", cex=1.3)
text(-0.5869739,-0.77521440,"Dos", cex=1.3)
text(-0.91134462,-0.46211689,"Pri", cex=1.3)
text(-1.1153932,-1.02288270,"Ran", cex=1.3)
text(-0.2093059,-0.03934922,"Saj", cex=1.3)
text(0.3702242,0.70132950,"Sac", cex=1.3)
text(2.6062083,0.03300452,"Sie", cex=1.3)
text(-1.462459,2.23441524,"Tor", cex=1.3)

# gráfico: componente 1 y 3 (eje X CP1, eje Y CP3)
plot(c(-1.5,3.2), c(-2.5,2.5), type="n", axes=FALSE, xlab=NA, ylab=NA, cex.main=2, main="B")
s.class(dfxy=acp$li, fac=tipo, xax=1, yax=3, grid=FALSE, cgrid=FALSE, cpoint=2, addaxes=FALSE,
        origin=c(0,0), add.plot=TRUE)

text(0.4951411,-0.04475918,"Alc", cex=1.3)
text(-0.3675646,0.95426525,"Alj", cex=1.3)
text(-1.1648300,0.82312057,"Ber", cex=1.3)
text(-0.4309082,2.03067982,"Cen", cex=1.3)
text(3.0160943,-0.81560159,"Cob", cex=1.3)
text(-0.0969739,0.69652987,"Dos", cex=1.3)
text(-0.91134462,-0.17651229,"Pri", cex=1.3)
text(-0.453932,-1.79500809,"Ran", cex=1.3)
text(-0.45093059,0.28768354,"Saj", cex=1.3)
text(0.3702242,-0.71284352,"Sac", cex=1.3)
text(2.6062083,0.56236212,"Sie", cex=1.3)
text(-1.462459,-1.00991649,"Tor", cex=1.3)
```

- ACP. Gráficos 4.4A y 4.4B.

```
acp.cv<-dudi.pca(df=marco.imp.cv, scannf=F,nf=3, scale=FALSE)
prcomp(marco.imp.cv, scale=FALSE)

plot(c(-0.35,0.32), c(-0.25,0.25), type="n", axes=FALSE, xlab=NA, ylab=NA, cex.main=2, main="A")
s.class(dfxy=acp.cv$li, fac=tipo, xax=1, yax=2, grid=FALSE, cgrid=FALSE, cpoint=2, addaxes=FALSE,
        origin=c(0,0), add.plot=TRUE)

text(-0.02502040,-0.163603600,"Alc", cex=1.3)
text(-0.08760213,0.0005875967,"Alj", cex=1.3)
text(-0.1243724,-0.0634414867,"Ber", cex=1.3)
text(0.03680707,0.0876650573,"Cen", cex=1.3)
text(-0.337088,0.1243761226,"Cob", cex=1.3)
text(0.18614,-0.1145911354,"Dos", cex=1.3)
text(0.075,0.0078347129,"Pri", cex=1.3)
text(0.180,0.2487364153,"Ran", cex=1.3)
text(0.14793443,0.04,"Saj", cex=1.3)
text(-0.065,-0.1185534264,"Sac", cex=1.3)
text(-0.238774,0.0293482729,"Sie", cex=1.3)
text(0.1867166,-0.0185632943,"Tor", cex=1.3)
```

```
# ejes 1 y 3

acp.cv$li

par(cex=1)

plot(c(-0.35,0.32), c(-0.25,0.25), type="n", axes=FALSE, xlab=NA, ylab=NA, cex.main=2, main="B")
s.class(dfxy=acp.cv$li, fac=tipo, xax=1, yax=3, grid=FALSE, cgrid=FALSE, cpoint=2, addaxes=FALSE,
        origin=c(0,0), add.plot=TRUE)


text(-0.055,-0.011283883,"Alc", cex=1.3)
text(-0.08760213,0.039893456,"Alj", cex=1.3)
text(-0.1243724,0.144555121,"Ber", cex=1.3)
text(0.03680707,-0.021980026,"Cen", cex=1.3)
text(-0.337088,0.015915171,"Cob", cex=1.3)
text(0.19,-0.107255301,"Dos", cex=1.3)
text(0.075,0.012104212,"Pri", cex=1.3)
text(0.17652367,0.011361296,"Ran", cex=1.3)
text(0.16793443,-0.045423463,"Saj", cex=1.3)
text(-0.02,0.011999056,"Sac", cex=1.3)
text(-0.238774,-0.150225873,"Sie", cex=1.3)
text(0.1867166,0.073340232,"Tor", cex=1.3)
```

A continuación, se muestran los resultados numéricos del ACP:

Tabla C.1: Coordenadas de los puntos representando las estaciones en la Figura 4.3.

Est.	X	Y	Z
Alc	0.29	-1.10	-0.044
Alj	-0.46	-2.22	0.75
Ber	-1.36	0.03	0.62
Cen	-0.43	1.93	1.83
Cob	3.41	0.69	-0.81
Dos	-0.32	-0.78	0.69
Pri	-1.11	-0.46	-0.17
Ran	-0.81	-1.02	-1.79
Saj	-0.50	-0.039	0.087
Sac	0.13	0.70	-0.71
Sie	2.98	0.033	0.56
Tor	-1.79	2.23	-1.01

Tabla C.2: Coordenadas de los puntos representando las estaciones en la Figura 4.4.

Est.	X	Y	Z
Alc	-0.025	-0.18	-0.011
Alj	-0.057	0.00	0.039
Ber	-0.094	-0.063	0.14
Cen	-0.036	0.067	0.0079
Cob	0.38	0.12	0.016
Dos	0.14	-0.11	-0.10
Pri	0.10	0.0078	0.012
Ran	0.21	0.24	0.011
Saj	0.12	0.017	-0.045
Sac	-0.028	-0.11	0.024
Sie	-0.19	0.029	-0.15
Tor	-0.15	-0.018	0.073

Tabla C.3: Cargas factoriales del ACP correspondiente a la Figura 4.3 basado en valores normalizados de μ_m. Los valores en negrilla se corresponden con las contribuciones más altas a los CPs.

Variable	PC1	PC2	PC3
CO	-0.333	0.356	**0.778**
NO_2	**-0.566**	0.0363	-0.428
O_3	0.302	**-0.621**	0.410
PM_{10}	**-0.605**	-0.172	0.199
SO_2	-0.330	**-0.674**	0.0402

Tabla C.4: Cargas factoriales del ACP correspondiente a la Figura 4.4 basado en valores no normalizados de cv_m. Los valores en negrilla se corresponden con las contribuciones más altas a los CPs.

Variable	PC1	PC2	PC3
CO	-0.252	**0.922**	0.185
NO_2	-0.530	-0.116	**0.604**
O_3	0.213	-0.0571	0.295
PM_{10}	-0.474	0.203	**-0.690**
SO_2	**-0.619**	0.301	0.188