Universitat d'Alacant
Universidad de Alicante

Clustering EBEM. Modelos de Mezclas Gausianas Basados en Maximización de Entropía.

Antonio Peñalver Benavent

Tesis Doctorales

www.eltallerdigital.com

UNIVERSIDAD de ALICANTE

Tesis doctoral

CLUSTERING EBEM. MODELOS DE MEZCLAS GAUSIANAS BASADOS EN MAXIMIZACIÓN DE ENTROPÍA

Presentada por *Antonio Peñalver Benavent*
Dirigida por *Francisco Escolano Ruiz*

Programa *Sistemas Industriales, Computación y Reconocimiento de Formas*
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante

15 de octubre de 2007

A mi padre
a quien las circunstancias adversas de la vida le impidieron lograr éste,
que era su sueño...

Prólogo

La verdad es que se me hace extraño llegar a esta parte, que a pesar de aparecer casi al principio, es el final de un trabajo de mucho, mucho tiempo. Quizá ahora es el momento de echar la vista atrás y recordar que todo empezó con una idea de mi director, Francisco Escolano, quien tras un tiempo revisando trabajos anteriores en los que se empleaban modelos de mezclas gausianas para tareas de clasificación, me planteó la siguiente cuestión: ¿que pasaría si empleáramos la entropía como medida de gausianidad para comprobar la calidad del ajuste del modelo a los datos? Recuerdo que pensé: ¿entropía? ¿qué es eso? llevaba varios meses en el grupo de investigación *Robot Vision Group* de la Universidad de Alicante y ya empezaba a entender, tras asistir a muchas charlas y horas de estudio, los fundamentos de la Teoría Bayesiana, el algoritmo EM y otras técnicas que empleaban los compañeros y Paco me comentaba algo nuevo que no había escuchado jamás. Ante mi cara de asombro, me prestó un libro de Teoría de la Información: *Elements of Information Theory* de Cover y Thomas para que fuera introduciéndome en la materia y tras hojearlo brevemente me di cuenta enseguida que teníamos mucho trabajo por delante.

Sin embargo, a los pocos meses ya teníamos desarrollada una primera versión del algoritmo y un método para estimar entropía a partir de un conjunto de observaciones y lo presentamos al CIARP 2003 [1]. Poco después, conceptos de Teoría de la Información y en especial la entropía, fueron utilizados con éxito por otros miembros del grupo para tareas diferentes, convirtiéndose en algo cotidiano a la hora de afrontar la solución a bastantes problemas en robótica y análisis de imagen.

Después de eso llegaron cambios importantes: de Departamento, Área de

[1] Iberoamerican Congress on Pattern Recognition

Conocimiento, Universidad, ... estuvimos prácticamente 2 años sin avanzar, hasta que en 2005 retomamos el trabajo y conseguimos una nueva forma para estimar la entropía, mejoramos el proceso de incorporación dinámica de componentes al modelo y lo aplicamos satisfactoriamente a la segmentación de imágenes en color, presentando los resultados en el S+SSPR 2006 [2], Iberamia 2006 [3] y ICPR 2006 [4]. Posteriormente probamos diferentes técnicas para la selección del orden del modelo que nos permitían detener el algoritmo cuando el número de componentes era óptimo y decidimos por fin escribir los resultados del trabajo.

Durante todo este tiempo son muchas las personas que de una u otra forma han contribuido a que este trabajo pudiera llevarse a cabo. Quiero agradecer en primer lugar a mi director, Francisco Escolano, por su dedicación y especialmente por su paciencia y comprensión tras el *parón* mencionado anteriormente. Su visión e intuición sobre qué técnicas debíamos probar y cuales no iban a ninguna parte fueron siempre acertadas. A los compañeros del grupo de investigación RVG, especialmente a los más *jóvenes*: Boyan y Pablo, cuyas líneas de investigación estaban más próximas a las mías y mostraron interés en el trabajo desde el principio, no dudando en ningún momento cruzar medio mundo para exponer los resultados cuando yo no pude hacerlo. A mi familia, que supo entender día tras día y noche tras noche que *papá* estuviera físicamente en la habitación de al lado, pero su cabeza estuviera en otro sitio.

He dejado para el final a la persona que sin lugar a dudas ha propiciado con su interés y apoyo que este trabajo saliera adelante. Tras varios años de carrera docente te cruzas todo tipo de personas, con algunas puedes trabajar codo con codo y enseguida te das cuenta que el trabajo cunde el doble, con otras (las menos) puedes hablar de casi cualquier cosa y se convierten en amigos fuera del ámbito de la Universidad, pero cruzarte con alguien que reúna las dos circunstancias anteriores es realmente difícil. A mi me ocurrió con Juanma; gracias amigo, por estar siempre ahí, en los momentos buenos y en los malos, que también los hubo. Este trabajo es tan tuyo como mío.

En cuanto al lector, sólo deseo que le guste el trabajo y las ideas que en él

[2] International Workshop on Statistical Pattern Recognition
[3] Ibero-American Artificial Intelligence Conference
[4] International Conference on Pattern Recognition

se proponen y que sepa disculpar los pequeños errores, que como experto en la materia podrá encontrar.

Antonio Peñalver
15 de octubre de 2007

Resumen

En este trabajo presentamos una nueva aproximación al problema de la estimación de los parámetros de un modelo de mezcla gausiana. Aunque el algoritmo *Expectation-Maximization* (EM) proporciona una solución iterativa de máxima verosimilitud, es conocida su sensibilidad a la elección de los valores iniciales del modelo, pudiendo converger a un máximo local de la función verosimilitud. Generalmente, algunas técnicas como k-means suelen emplearse para establecer los valores iniciales del modelo, sin embargo, puesto que se trata igualmente de algoritmos locales, sólo se incrementa la velocidad de convergencia del algoritmo hacia algún máximo local, pero no queda en ningún caso asegurada la consecución del máximo global. Por otra parte, el resultado obtenido es igualmente dependiente del número de componentes de la mezcla, que en la mayoría de las situaciones es desconocido a priori.

Para solventar los inconvenientes descritos anteriormente, introducimos un criterio basado en la estimación de la entropía de la densidad de probabilidad asociada a cada componente, que permite medir la calidad del ajuste de un modelo de mezcla con un determinado número de componentes. Proponemos dos métodos para estimar la entropía asociada a cada núcleo y una modificación del algoritmo EM clásico para encontrar el número óptimo de componentes de la mezcla. Además, empleamos dos criterios de parada para seleccionar el orden del modelo, uno basado en la entropía global de la mezcla y otro basado en el principio de *Longitud de Descripción Mínima* (MDL). El algoritmo comienza con un sólo núcleo y va añadiendo dinámicamente nuevos núcleos en las zonas del espacio de observaciones en que el ajuste es menos fino. De este modo, se elimina el problema de la inicialización del modelo y se obtiene el orden del mismo (número óptimo de núcleos) que mejor

se ajusta al conjunto de observaciones dadas.

El algoritmo ha sido probado con éxito en estimación de densidad de probabilidad asociada a los datos, reconocimiento de patrones y segmentación de imágenes en color. Además comparamos los resultados de la técnica con los obtenidos con EM clásico y otras que también ajustan dinámicamente el modelo y que han sido propuestas con anterioridad. Aunque el problema ha sido tratado por numerosos investigadores, la mejor forma de resolver la cuestión en la práctica es todavía un problema abierto.

Abstract

In this work, we address the problem of estimating the parameters of a Gaussian mixture model. Although the standard *Expectation-Maximization* (EM) algorithm yields the maximum-likelihood solution, it is well-known that it is prone to the selection of the starting parameters of the model and it may converge to the boundary of the parameter space. Usually, some approaches like k-means are used to set the starting values of the model; however, only the convergence speed of the algorithm to a local maxima is increased because these approaches are local too, and a global maximum is not ensured in any case. Furthermore, the resulting mixture depends on the number of selected components, but the optimal number of kernels in the mixture may be unknown beforehand.

In order to solve the drawbacks cited above, we introduce a criterion based on the entropy of the probability density function associated to each kernel to measure the quality of a given mixture model with a fixed number of kernels. We propose two methods to approximate the entropy of each kernel and a modification of the classical EM algorithm in order to find the optimum number of the mixture components. Furthermore, we use two stopping criteria to find the order of the model, one of them is based on the global entropy of the mixture and the other one is based on the *Minimum Description Length* (MDL). The algorithm starts with only one component and new kernels are dynamically introduced in the regions of the sample space in which the fitting is coarser. By this way, we avoid the boundary of the parameters space and we obtain the order of the model (optimal number of kernels) which yields the best fitting to the set of given data.

We have successfully tested our algorithm in probability density estimation, pattern recognition and color image segmentation. Furthermore, we

compare our results with those obtained with the classical EM algorithm with a fixed number of kernels, and with other approaches previously proposed that automatically select the number of components too. Although this problem has been pointed out by many researchers, the best way to solve it in practice is still an open question.

Contenido

Capítulo 1

Introducción y objetivos

En este capítulo realizaremos una visión general de la tesis. Comenzaremos presentando los modelos de mezclas finitas y en particular los modelos de mezclas que emplean núcleos gausianos, las diferentes áreas en las que su aplicación es de especial interés y los distintos métodos existentes para ajustar adecuadamente el modelo a la resolución de un problema particular. Finalizaremos el capítulo con una descripción de los objetivos del trabajo así como con un esquema general de funcionamiento de la técnica propuesta.

1.1. Motivación

Los modelos de mezclas finitas, y especialmente los basadas en núcleos gausianos, son una potente herramienta probabilística para modelado de datos en una o más dimensiones. En la actualidad, es ampliamente conocida la utilidad de este tipo de modelos en cualquier área que implique una representación estadística de los datos como reconocimiento de patrones, visión por computador, análisis de señal e imagen o aprendizaje.

En el área de reconocimiento estadístico de patrones los modelos de mezclas permiten llevar a cabo un planteamiento formal, basado en modelos probabilísticos del aprendizaje no supervisado [Jain y Dubes, 1988] [Jain *et al.*, 2000] [McLachlan y Basford, 1988] [McLachlan y Peel, 2000]

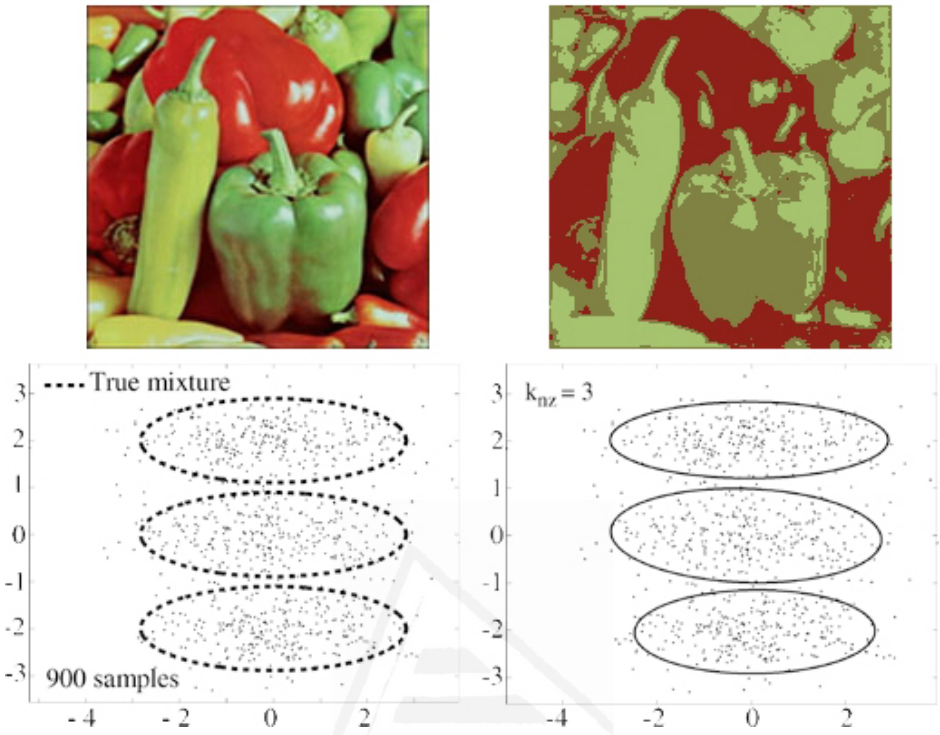

Figura 1.1: Dos ejemplos de utilización de modelos de mezclas: Segmentación de imágenes en color (arriba) y estimación de densidad de probabilidad asociada a un conjunto de datos (abajo).

[Titterington *et al.*, 1985]. En al área de visión por computador pueden ser empleados para segmentación de imágenes en color [Peñalver *et al.*, 2006c], para predicción no lineal de imágenes [Zhang y Ma, 2004], para clasificación no supervisada de imágenes en el contexto de teoría de la información [Goldberger *et al.*, 2006] o para estimar los principales planos 3D del entorno (paredes, suelo y techo) en tareas de navegación de robots [Sáez *et al.*, 2003].

Los modelos de mezclas finitas permiten representar las observaciones de una forma natural, asumiendo que han sido generadas por una fuente perteneciente a un conjunto de posibles fuentes aleatorias, aunque se desconoce inicialmente la fuente que generó cada uno de los datos. Si conseguimos determinar los parámetros que definen a cada una de las fuentes, así como la fuente que generó cada una de las observaciones, podremos realizar un clustering del conjunto inicial de observaciones. Este enfoque basado en

modelos, permite afrontar la selección del número óptimo de elementos o la evaluación de la validez de los modelos obtenidos de una manera formal, a diferencia de otros métodos heurísticos como *k*-means [Lloyd, 1982] o los métodos acumulativos jerárquicos [Jain y Dubes, 1988]. En la imagen de la figura 1.1 mostramos dos de las aplicaciones que se llevarán a cabo en el presente trabajo: segmentación de imágenes en color y estimación de densidad de probabilidad.

La utilidad de los modelos de mezclas no está limitada únicamente al aprendizaje no supervisado. Los modelos de mezclas permiten representar funciones de densidad de probabilidad (PDF's) más o menos complejas. Por tanto, son una buena opción para la representación de funciones de densidad de probabilidad condicionadas, como por ejemplo funciones de verosimilitud, en escenarios de aprendizaje bayesiano supervisado [Hastie y Tibshirani, 1996] [Hinton *et al.*, 1997] [Streit y Luginbuhl, 1994] o para inicializar probabilidades a priori en estimación bayesiana de parámetros [Dalal y Hall, 1983]. Los modelos de mezclas también han sido empleados con éxito para realizar una selección adecuada de características en tareas de reconocimiento de patrones [Pudil *et al.*, 1995] [Law *et al.*, 2004].

1.2. Técnicas para el ajuste de mezclas gausianas

A lo largo de los años, diferentes autores han propuesto distintas técnicas para ajustar adecuadamente los parámetros de un modelo de mezclas a un conjunto de datos observados. Uno de los métodos más empleados para ello es el algoritmo *Expectation-Maximization* (EM) [Dempster *et al.*, 1977] [McLachlan y Krishnan, 1997] [McLachlan y Peel, 2000], que converge a una estimación de *máxima verosimilitud* (ML) del conjunto de parámetros de la mezcla. La técnica propuesta en el presente trabajo se basa en una modificación de dicho algoritmo, no obstante, existen otros métodos que vamos a revisar brevemente, como los métodos *Reversible Jump Markov Chain Montecarlo* o aquellos basados en combinaciones del Algoritmo EM y algoritmos genéticos. En todos los casos, el número de componentes de la mezcla es desconocido a priori y debe ser estimado igualmente. En el capítulo 2 realizaremos una revisión detallada de los modelos de mezclas gausianas y en

especial, de otras técnicas anteriores basadas también en el algoritmo EM, que permiten determinar el número óptimo de componentes de la mezcla.

1.2.1. Métodos de Monte Carlo

Una de las aplicaciones más extendidas del algoritmo *Reversible Jump Markov Chain Monte Carlo* (RJMCMC) [Green, 1995] es la de ajustar modelos de mezclas gausianas con un número de núcleos desconocido a priori. Este planteamiento es empleado en [Richardson y Green, 1997] [Nobile y Green, 2000] [Robert *et al.*, 2000] [Fernandez y Green, 2002] [Green y Richardson, 2001] [Bottolo *et al.*, 2003] para el caso unidimensional o las recientes extensiones al caso multi-dimensional de [Zhang *et al.*, 2004] [Dellaportas y Papageorgiou, 2006]. El algoritmo estima simultáneamente tanto los parámetros de cada uno de los núcleos como el número óptimo de ellos, lo que se conoce habitualmente como *orden del modelo*.

La idea común de este conjunto de técnicas es la de ajustar los parámetros de la mezcla tras un proceso iterativo que combina movimientos de unión y división de los núcleos existentes. La elección de uno de los dos movimientos se realiza de forma aleatoria. El movimiento de fusión selecciona dos núcleos aleatoriamente del conjunto de núcleos disponibles en un paso de ejecución del algoritmo, mientras que el movimiento de división selecciona igualmente un núcleo al azar y lo descompone en otros dos. La única restricción tras la ejecución de cualquiera de los dos movimientos es que se conserven los dos primeros momentos estadísticos antes y después del proceso.

Aunque estos métodos pueden, en principio, encontrar una solución óptima al problema del ajuste del modelo, tienen un coste computacional muy elevado. Por ello son menos eficientes que el algoritmo EM en aplicaciones de visión por computador o reconocimiento de patrones, en las que se requiere el tratamiento de un número elevado de datos.

1.2.2. Algoritmos genéticos

Otra técnica para el ajuste de los parámetros de la mezcla combina el algoritmo EM y algoritmos genéticos (GA) para realizar una estimación óptima de los mismos, como en [Martinez y Vitria, 2000]. Este planteamiento ha sido aplicado incluso a tareas de navegación en robots [Martinez y Vitria, 2001].

En ambos casos, el número de componentes de la mezcla debe ser fijado de antemano, por lo que la técnica no permite realizar una estimación óptima del orden del modelo al mismo tiempo que se obtienen los parámetros de la muestra.

Más recientemente, en [Pernkopf y Bouchaffra, 2006] se propone una generalización del algoritmo anterior que en combinación con el *Principio de Longitud de Descripción Mínima* (MDL)[1], que será revisado con detalle en el capítulo 2, reduce la sensibilidad del algoritmo a la inicialización y permite averiguar el número de componentes de la mezcla al mismo tiempo que se ajustan los parámetros de la misma.

El principal inconveniente del algoritmo, comparado con el EM clásico, es que requiere la estimación de una serie de parámetros adicionales a los de la propia mezcla. Además muestra una dependencia importante ante la presencia de falsos positivos en el conjunto de observaciones a modelar.

1.2.3. Algoritmo EM

Por los inconvenientes descritos con anterioridad, en el presente trabajo nos decantamos por la utilización del algoritmo EM. Este algoritmo, cuyas iniciales provienen de la expresión *Expectation-Maximization* (EM) [Dempster *et al.*, 1977] [McLachlan y Krishnan, 1997] [McLachlan y Peel, 2000] es uno de los métodos más empleados para la estimación de los parámetros de un modelo de mezclas finitas. Se trata de un procedimiento iterativo que converge a una estimación de *máxima verosimilitud* de los parámetros del modelo. Puesto que el presente trabajo se basa en una modificación de este algoritmo, en el capítulo 2 llevaremos a cabo una revisión de las diferentes técnicas de ajuste propuestas con anterioridad y basadas en el mismo algoritmo.

Este algoritmo posee algunos inconvenientes que es necesario solventar para poder ser aplicado eficientemente a la resolución de problemas en los que es adecuado: puesto que se trata de un método local, es sensible a la inicialización y por tanto, podría converger hacia un máximo local de la función de verosimilitud. Otro aspecto importante a tener en cuenta para ajustar el modelo es el número de componentes de la mezcla, pues un número ele-

[1] MDL proviene del término en inglés *Minimum Description Length*

vado de ellos podría provocar una partición excesiva del espacio de datos (*over-fitting*), mientras que un número excesivamente reducido no sería lo suficientemente flexible para aproximarse al modelo real que siguen los datos.

1.3. Objetivos

En este trabajo proponemos una modificación del algoritmo EM clásico que permita dar una solución a los problemas anteriores. Los objetivos que se plantean son los siguientes:

- Determinar automáticamente el número óptimo de componentes de la mezcla. A este subproblema se le denomina en la literatura *Selección del Orden del Modelo*. En el capítulo 2 se hará una revisión detallada de las diferentes técnicas propuestas con anterioridad para determinar dicho número. Posteriormente, en el capítulo 3 propondremos dos criterios de parada adecuados para diferentes tipos de problema: uno basado en la entropía de Shannon y otro basado en los principios de *Longitud Mínima*, tanto de descripción (MDL), como de mensaje (MML).

- Modificar el funcionamiento básico del algoritmo EM para que no sea sensible a la inicialización. Para evitar tener que ajustar los parámetros iniciales de los núcleos comenzaremos con un sólo núcleo cuyos parámetros iniciales serán obtenidos a partir del conjunto de observaciones. Posteriormente se irán añadiendo dinámicamente nuevos núcleos a la mezcla. La descripción detallada del algoritmo propuesto se realiza en el capítulo 3.

- Para comprobar la calidad del ajuste del modelo a los datos dado un número determinado de núcleos y determinar qué zona del espacio de datos es la peor ajustada, planteamos la utilización de la entropía de Shannon. Es conocido por la *Teoría de la Información* que a igualdad de matriz de covarianza, la distribución de mayor entropía es la distribución normal y su cálculo puede realizarse de forma cerrada. Cuando los datos estén correctamente ajustados: la entropía real y la teórica estarán próximas. El principal problema asociado a este planteamiento es el cálculo de la entropía real asociada a los datos. Proponemos el empleo de

dos técnicas, una que requiere previamente el calculo de la densidad de probabilidad asociada a los datos y otra que realiza una estimación directa de la entropía sin estimar previamente la función densidad de probabilidad. Cada técnica posee ventajas e inconvenientes que serán revisados con detalle en el capítulo 4.

- Evitar que el algoritmo converja hacia un máximo local del espacio de parámetros. Para ello se propone insertar de forma adecuada cada uno de los nuevos núcleos, ajustando convenientemente el conjunto inicial de parámetros de cada uno de ellos. Proponemos dos técnicas, una heurística y otra basada en descomposición matricial con un fundamento teórico más elaborado. En el capítulo 2 se realiza una descripción detallada de otras técnicas propuestas con anterioridad para evitar dicho problema.

- Para verificar el funcionamiento del algoritmo, se han realizado múltiples experimentos de estimación de densidad de probabilidad, clasificación de patrones y segmentación de imágenes en color empleando las dos medidas de estimación de entropía y diferentes criterios de parada que serán detallados en el capítulo 4. Los resultados se han comparado con los obtenidos con el algoritmo EM clásico con un número fijo de núcleos de partida y con otras de las técnicas revisadas en el capítulo 2. Los resultados mejoran claramente los obtenidos con EM clásico y eliminan los problemas de convergencia hacia máximos locales de la función de verosimilitud relativos a la inicialización asociados a otras técnicas. En el capítulo 5 detallaremos las conclusiones obtenidas en los diferentes experimentos llevados a cabo con la aplicación de la técnica y los desarrollos futuros.

En la figura 1.2 mostramos un esquema resumen del algoritmo propuesto en un ejemplo de estimación de densidad de probabilidad en dos dimensiones: Para evitar problemas derivados de la inicialización, partimos de un sólo núcleo cuyo conjunto de parámetros es directamente obtenido del conjunto de observaciones. Se realizan diferentes iteraciones del algoritmo EM hasta obtener la solución de máxima verosimilitud para ese número de núcleos. A continuación se comprueba si el orden del modelo es correcto. Para ello

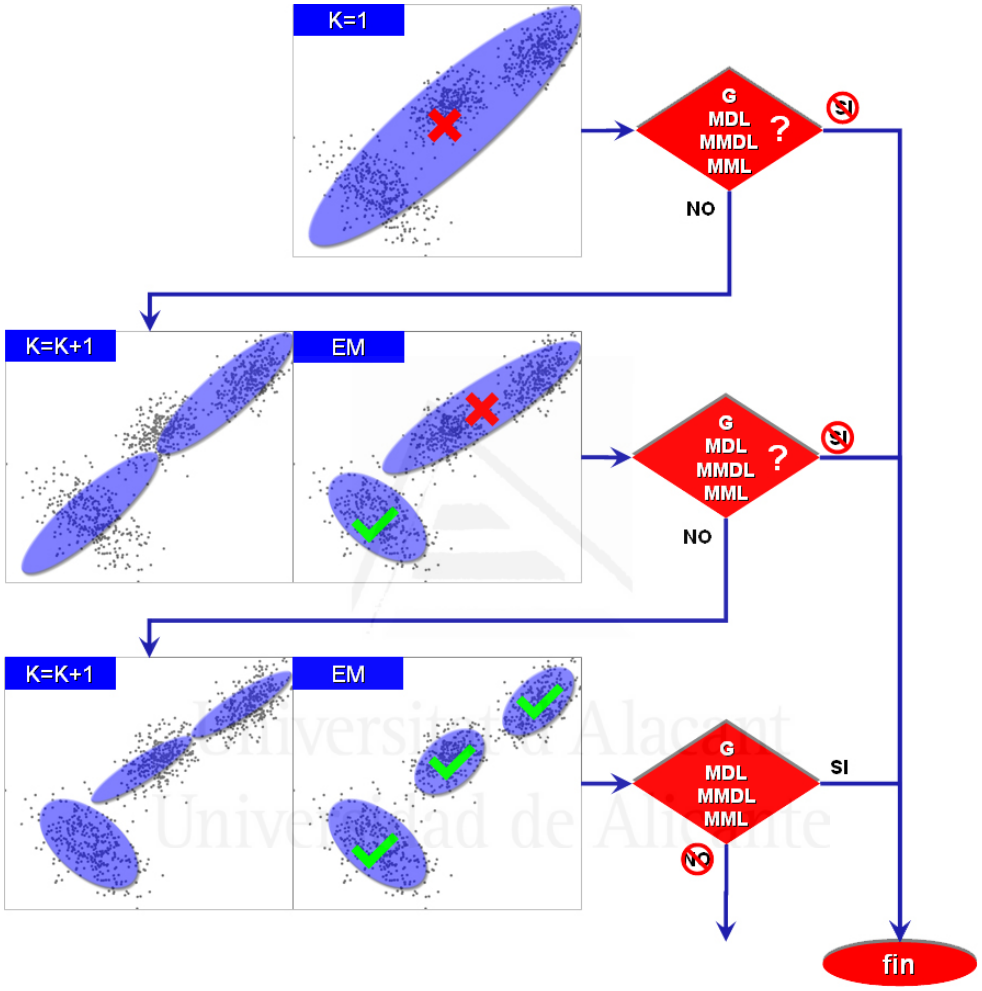

Figura 1.2: Resumen del algoritmo propuesto para el ajuste del modelo de mezcla gausiana y selección del número óptimo de núcleos. La imagen muestra un ejemplo de estimación de densidad de probabilidad en un espacio de 2 dimensiones, aunque como se verá a lo largo del trabajo, la técnica permite su aplicación en espacios de dimensionalidad mayor y problemas diferentes.

emplearemos una medida basada en entropía y otros criterios basados en los principios de descripción mínima, que serán detallados en capítulos posteriores. Si el número de núcleos es correcto, el algoritmo finaliza y el conjunto óptimo de parámetros es el obtenido en la última iteración. Si no, se selecciona el núcleo peor ajustado. El proceso consiste en la comparación entre su entropía real obtenida a partir de los datos, con la máxima teórica en el caso de que los datos fueran verdaderamente gausianos. El núcleo que presenta mayor diferencia se descompone en otros dos que son correctamente inicializados. El proceso de selección del peor núcleo y la correcta inicialización de sus parámetros serán explicados con detalle a lo largo del trabajo. A continuación se realizan nuevas iteraciones EM del algoritmo para $K + 1$ núcleos hasta alcanzar convergencia y se repite nuevamente el proceso.

Capítulo 2

Estado del arte

En este capítulo realizamos una introducción a las diferentes técnicas propuestas en la literatura para estimar la densidad de probabilidad asociada a un conjunto de observaciones o datos y en particular los modelos de mezclas gausianas. Para ello se lleva a cabo una revisión exhaustiva de las diferentes técnicas propuestas con anterioridad para ajustar el modelo a los datos y los principales inconvenientes asociados a cada una de ellas.

2.1. Modelos de mezclas gausianas

Tradicionalmente, las técnicas para la estimación de la densidad de probabilidad asociada a un conjunto de observaciones se han clasificado en *técnicas paramétricas* y *no paramétricas*. Cada una de ellas tiene sus propias ventajas e inconvenientes que describimos a continuación:

- Las técnicas **paramétricas** asumen que la densidad de probabilidad asociada a los datos sigue una forma específica, que puede diferir bastante de la densidad real. Sin embargo, este planteamiento permite que la función densidad sea evaluada rápidamente cada vez que surge una nueva observación.

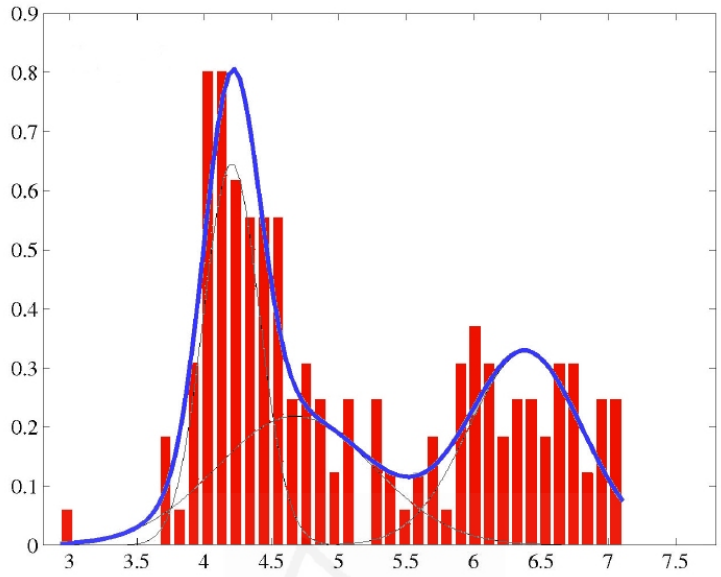

Figura 2.1: Ajuste de un conjunto de datos de una sola dimensión con una mezcla de tres distribuciones gausianas.

- Los métodos **no-paramétricos**, por el contrario, permiten que la forma de la función densidad sea mucho más general, pero sufre el inconveniente de que el número de parámetros del modelo crece proporcionalmente al número de observaciones.

Para combinar las ventajas de ambos métodos, es necesario encontrar técnicas que no estén restringidas a una forma de función en particular, pero en las que el tamaño del modelo crezca en proporción a la complejidad del problema a resolver y no al tamaño del conjunto de observaciones disponibles. A las técnicas que cumplen lo anterior se las denomina *semi-paramétricas*. El principal inconveniente es que el ajuste de los parámetros del modelo a un conjunto de observaciones en particular es computacionalmente más costoso, comparado con los procedimientos más simples requeridos por las técnicas paramétricas y no-paramétricas. Estas necesitan evaluar unas pocas expresiones para ajustar el valor de los parámetros o un simple almacenamiento del conjunto de observaciones.

En este trabajo centraremos nuestra atención en los modelos de mezclas, pertenecientes a las técnicas semi-paramétricas definidas en el apartado ante-

rior y que modelan la densidad de probabilidad asociada al conjunto de datos como una superposición lineal de funciones denominadas núcleos. A los modelos de mezclas que emplean como funciones núcleo funciones normales o gausianas, se les conoce habitualmente como *Modelos de Mezclas Gausianas*.

En la figura 2.1 mostramos un conjunto de observaciones en un espacio de una sola dimensión ajustadas por un modelo de mezcla gausiana de tres componentes. Las barras verticales representan la distribución del conjunto de datos. Las líneas de trazo fino muestran cada una de las distribuciones gausianas del modelo, mientras que la línea azul de trazo grueso representa la mezcla resultante de todas ellas a partir de los parámetros del modelo.

2.2. Definición

Una variable aleatoria d-dimensional $\mathbf{y}$ sigue una distribución de mezcla finita cuando su *función densidad de probabilidad* (PDF) $p(y|\Theta)$ puede ser expresada como una suma ponderada de funciones de densidad de probabilidad denominadas núcleos. Cuando todas esas funciones núcleo son gausianas, la mezcla es denominada de la misma forma: *Mezcla Gausiana*:

$$p(\mathbf{y}|\Theta) = \sum_{i=1}^{K} \pi_i p(\mathbf{y}|\Theta_i) \tag{2.1}$$
$$0 \leq \pi_i \leq 1, \qquad i = 1, ..., K, \text{ y } \sum_{i=1}^{K} \pi_i = 1,$$

donde K representa el número de núcleos del modelo, $\pi_1, ..., \pi_k$ son las probabilidades a priori de cada uno de los núcleos y Θ_i es el conjunto de parámetros que describe a cada núcleo. En el caso de las mezclas gausianas $\Theta_i = \{\mu_i, \Sigma_i\}$, esto es, la media y la matriz de covarianza que caracterizan a la distribución normal.

El conjunto completo de parámetros a estimar en un modelo de mezcla dado es: $\Theta \equiv \{\Theta_1, ..., \Theta_k, \pi_1, ..., \pi_k\}$. La obtención del conjunto óptimo de parámetros Θ^*, es decir, los valores que mejor se ajustan a un conjunto de observaciones, suele plantearse como aquellos que maximizan la función de verosimilitud de la función densidad de probabilidad a ser estimada. Una revisión

de las técnicas en máxima verosimilitud en este contexto puede ser consultada en [Redner y Walker, 1984]. Para reducir la complejidad de las operaciones matemáticas del modelo, se suele emplear el logaritmo de la verosimilitud, que para un conjunto de observaciones dado Y, se define como:

$$\ell(Y|\Theta) = \log p(Y|\Theta) = \log \prod_{n=1}^{N} p(y_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(y_k|\Theta_k). \tag{2.2}$$

donde $Y = \{y_1, ...y_N\}$ es un conjunto de N observaciones de la variable Y obtenidas de forma independiente e igualmente distribuidas. Es bien sabido que la estimación de *máxima verosimilitud* (ML):

$$\Theta^*_{ML} = \arg \max_{\Theta} \{\ell(\Theta)\}. \tag{2.3}$$

no puede ser determinada analíticamente. Es importante destacar que la maximización de la función anterior no es una tarea trivial por varias razones:

- Existen valores del conjunto de parámetros para los que la verosimilitud es infinita. Esto ocurre cuando uno de los núcleos gausianos se colapsa en alguna de las observaciones, es decir, la media del núcleo coincide con el dato y la matriz de covarianza es nula.
- La presencia de grupos de observaciones muy próximas unas de otras puede provocar un máximo local de la función. Como consecuencia, se obtendría una pobre representación de la densidad de probabilidad real.

Para evitar las situaciones descritas anteriormente, han sido propuestas diferentes técnicas. Ver [Day, 1969] para una descripción detallada.

El mismo problema ocurre con el criterio bayesiano *máximo a posteriori* (MAP), definido como:

$$\Theta^*_{MAP} = \arg \max_{\Theta} \{(\ell(\Theta) + \log p(\Theta))\}. \tag{2.4}$$

Por tanto, es necesaria la utilización de algún método iterativo, como el algoritmo EM, para la determinación del conjunto óptimo de parámetros. Tanto si se opta por el criterio ML, como si se emplea MAP, las estimaciones deben

cumplir las restricciones definidas en 2.1. Para una descripción más detallada de los modelos de mezclas recomendamos [McLachlan y Basford, 1988] [Bishop, 1994] [McLachlan y Peel, 2000] [Titterington *et al.*, 1985]. Aquí únicamente revisamos las ideas fundamentales y definimos la notación que se empleará en el resto del trabajo.

2.3. El algoritmo EM (Expectation-Maximization)

Una de las elecciones más usuales para obtener estimaciones de máxima verosimilitud o máximo a posteriori para los parámetros de la mezcla es el algoritmo EM [Dempster *et al.*, 1977] [McLachlan y Basford, 1988] [McLachlan y Krishnan, 1997] [McLachlan y Peel, 2000]. EM es un procedimiento iterativo que permite encontrar soluciones de máxima verosimilitud a problemas en los que existen *variables ocultas*. En el caso de las mezclas gausianas [Redner y Walker, 1984], dichas variables son un conjunto de N etiquetas $Z = \{z^1, ..., z^N\}$ asociadas a cada una de las observaciones. Cada etiqueta es un vector binario $z^i = [z_1^{(i)}, ..., z_k^{(i)}]$, con $z_m^{(i)} = 1$ y $z_p^{(i)} = 0$ si $p \neq m$, indicando que $y^{(i)}$ ha sido generada por el núcleo m. De este modo, si denominamos $X = \{Y, Z\}$ al conjunto completo de observaciones, podríamos expresar el logaritmo de la verosimilitud como:

$$\log p(Y, Z|\Theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_k^n \log[\pi_k p(y_n|\Theta_k)]. \tag{2.5}$$

El algoritmo EM genera una secuencia de estimaciones del conjunto de parámetros $\{\Theta^*(t), t = 1, 2, ...\}$ alternando los pasos E *(Expectation)* y M *(Maximization)* hasta lograr la convergencia. Las cuestiones relativas a la convergencia del método han sido ampliamente estudiadas en [McLachlan y Peel, 2000] [Xu y Jordan, 1996]. A continuación detallamos las acciones realizadas en cada uno de los pasos del algoritmo.

Paso E.

En este paso del algoritmo se realiza una estimación del valor esperado de las variables ocultas del problema a partir de los datos observados Y y la

estimación actual de los parámetros del modelo $\Theta^*(t)$. Dicho valor esperado puede ser expresado de la siguiente forma:

$$E[z_k^{(n)}|y,\Theta^*(t)] = P[z_k^{(n)} = 1|y,\Theta^*(t)]) = \frac{\pi_k^*(t)p(y^{(n)}|\Theta_k^*(t))}{\Sigma_{j=1}^K \pi_j^*(t)p(y^{(n)}|\Theta_k^*(t))}, \tag{2.6}$$

por tanto, la probabilidad de que la observación $\mathbf{y}_n$ haya sido generada por el núcleo k puede calcularse como:

$$p(k|\mathbf{y}_n) = \frac{\pi_k p(\mathbf{y}^{(n)}|k)}{\Sigma_{j=1}^K \pi_j p(\mathbf{y}^{(n)}|k)} \tag{2.7}$$

Paso M.

A partir del valor esperado de Z, el nuevo conjunto de parámetros $\Theta^*(t+1)$ puede ser expresado mediante:

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N} p(k|\mathbf{y}_n), \tag{2.8}$$

$$\mu_k = \frac{\sum_{n=1}^N p(k|\mathbf{y}_n)\mathbf{y}_n}{\sum_{n=1}^N p(k|\mathbf{y}_n)}, \tag{2.9}$$

$$\Sigma_k = \frac{\sum_{n=1}^N p(k|\mathbf{y}_n)(\mathbf{y}_n - \mu_k)(\mathbf{y}_n - \mu_k)^T}{\sum_{n=1}^N p(k|\mathbf{y}_n)}, \tag{2.10}$$

En [Redner y Walker, 1984], puede ser consultada una descripción más detallada del algoritmo. En este trabajo vamos a centrarnos en el hecho de que si el número de núcleos K no es conocido de antemano, no puede ser obtenido a partir de la maximización del logaritmo de la verosimilitud como el resto de parámetros del modelo, ya que $\ell(\Theta)$ crece con K.

Expresado formalmente, si denominamos M_k a la clase de todos los posibles modelos de mezclas de k componentes construidas a partir de algún tipo de función de densidad de probabilidad (p.e. todas las gausianas d-dimensionales con matriz de covarianza no restringida), entonces $M_k \subseteq M_{k+1}$, es decir, las clases están anidadas. Como ejemplo, si $\Theta = \{\Theta_1, \Theta_2, ..., \Theta_k, \pi_1, \pi_2, ..., \pi_{k-1}, \pi_k\}$, define una mezcla en M_k y $\Theta' = \{\Theta_1, \Theta_2, ..., \Theta_k, \Theta_{k+1}\pi_1, \pi_2, ..., \pi'_{k-1}, \pi'_k, \pi'_{k+1}\}$, define una mezcla en M_{k+1}. Si

$\Theta_{k+1} = \Theta_k$ y $\pi_k = \pi_k^{'} + \pi_{k+1}^{'}$, entonces Θ y $\Theta^{'}$ representan la misma función de densidad de probabilidad. Por lo tanto, el valor de los parámetros que maximizan la verosimilitud $p(\mathbf{y}|\Theta_{ML}^{*})$ es una función no decreciente en k lo que implica que no puede ser empleado para la estimación del número de componentes de la mezcla.

Además, si se escoge un número equivocado de núcleos K, se puede obtener una estimación errónea. Para ilustrar este hecho, la figura 2.2 muestra el efecto de emplear un sólo núcleo para describir los datos, cuando realmente las observaciones pertenecen a dos núcleos gausianos claramente diferenciados.

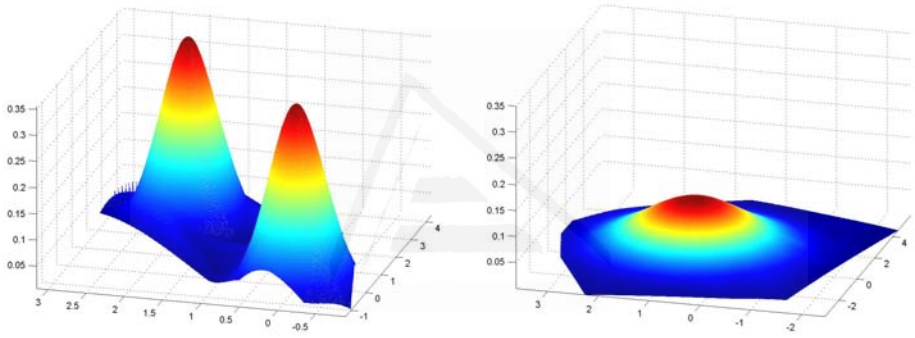

Figura 2.2: Si el número de núcleos del modelo no es establecido correctamente, los datos pueden describirse erróneamente. En el ejemplo, dos distribuciones gausianas con medias $\mu_1 = [0, 0]$ y $\mu_2 = [3, 2]$ (izquierda) son modeladas con un único núcleo con media $\mu = [1.5, 1]$ (derecha).

2.4. Inconvenientes de los esquemas basados en EM

La mayor parte de los algoritmos existentes para ajustar los parámetros de una mezcla con un número desconocido de núcleos a priori utilizan el algoritmo EM. Aunque el funcionamiento del algoritmo es satisfactorio, presenta algunos inconvenientes que se detallan a continuación:

- **Inicialización**: El éxito del algoritmo EM depende en gran medida de los valores iniciales del conjunto de parámetros. La mayoría de las soluciones propuestas incluyen una o varias de las

siguientes estrategias: (i) emplear varias inicializaciones aleatorias y seleccionar la que concluye con un mayor valor de verosimilitud [Hastie y Tibshirani, 1996] [McLachlan y Krishnan, 1997] [McLachlan y Peel, 2000] [Roberts *et al.*, 1998] o bien (ii) realizar un clustering previo con alguno de los algoritmos existentes para ello [Hastie y Tibshirani, 1996] [McLachlan y Krishnan, 1997] [McLachlan y Peel, 2000]. Cualquiera de las soluciones anteriores requiere un tiempo de cómputo adicional al del propio algoritmo EM.

- **Convergencia hacia un máximo local**: Cuando se ajustan los parámetros de un modelo de mezcla gausiana sin ningún tipo de restricción sobre las matrices de covarianza de los núcleos, alguno de los π_i podría aproximarse a cero y por consiguiente, el núcleo podría estar arbitrariamente próximo a la singularidad. Cuando el número de núcleos es superior al óptimo esto puede ocurrir con relativa frecuencia, convirtiéndose por tanto en un serio problema para métodos que requieren estimaciones de los parámetros de la muestra para varios valores de K. Este problema puede ser resuelto empleando algún tipo de restricción para las matrices de covarianza, como se sugiere en [Kloppenburg y Travan, 1997].
- **No se estima el orden del modelo**: Como se ha descrito anteriormente, el algoritmo por si mismo no permite la estimación del número de componentes del modelo.

2.5. Determinación del orden del modelo

El problema de la estimación del número óptimo de núcleos en la muestra es comúnmente conocido como selección del orden del modelo. Desde un punto de vista computacional, la mayoría de esos métodos pueden clasificarse como *estocásticos* y *deterministas*. Los primeros están basados en los métodos Markov Chain Monte Carlo (MCMC) y ya han sido brevemente revisados en el capítulo 1 y, en cualquier caso, se trata de métodos computacionalmente muy costosos para tareas de reconocimiento de patrones. En este apartado realizaremos una revisión de las técnicas deterministas que emplean el algoritmo EM para ajustar el modelo de mezcla a los datos.

Los métodos deterministas obtienen una serie de modelos posibles, normalmente empleando EM para un número de núcleos k entre k_{min} y k_{max} entre los que se supone que se encontrará el valor óptimo. A continuación, el número de componentes es seleccionado de la siguiente forma:

$$k^* = \arg\min_k \{\varsigma(\Theta^*(k), k), k = k_{min}, ..., k_{max}\}, \tag{2.11}$$

donde $\varsigma(\Theta^*(k), k)$ es algún criterio de selección y $\Theta^*(k)$ es una estimación de los parámetros de la mezcla para el caso de disponer de k núcleos. Por regla general, el criterio tiene la forma:

$$\varsigma(\Theta^*(k), k) = -\log(p(\mathbf{y}|\Theta^*(k))) + \wp(k), \tag{2.12}$$

con $\wp(k)$ una función creciente cuyo objetivo es penalizar altos valores de k que generen una mezcla con un excesivo número de núcleos. Algunos ejemplos de de trabajos que incluyen criterios del tipo anterior serían los siguientes:

- Criterios de aproximación bayesiana como el *Laplace-empirical criterion* (LEC) en [Roberts *et al.*, 1998] o el criterio *Schwarz's Bayesian inference criterion* (BIC) en [Schwarz, 1978] [Campbell *et al.*, 1997] [Dasgupta y Raftery, 1998] [Fraley y Raftery, 1997].

- Criterios basados en conceptos procedentes de la Teoría de la Información, como el *Principio de Longitud de Descripción Mínima* (MDL) [Rissanen, 1983], cuya expresión formal coincide con BIC, el criterio de *mínima longitud de mensaje* (MML) [Wallace y Freeman, 1987] [Oliver *et al.*, 1996] [Wallace y Dowe, 1999], el *criterio de información de Akaike* (AIC) [Whindham y Cutler, 1992] o el *criterio de complejidad de la información* (ICOMP) [Bozdogan, 1993].

- Criterios basados en el cálculo de la verosimilitud de los datos completos, expresada en la ecuación 2.5, también conocido como verosimilitud de clasificación. Entre estos se encuentran el método de *approximate weight of evidence* (AWE) [Banfield y Raftery, 1993], *classification likelihood criterion* (CLC) [Biernacki y Govaert, 1997], *normalized entropy criterion* (NEC) [Biernacki *et al.*, 1999] [Celeux y Soromenho, 1996] o el criterio *integrated classification likelihood* (ICL) [Biernacki *et al.*, 2000].

En [McLachlan y Peel, 2000] se puede consultar una descripción más detallada de todas estas técnicas, que incluye un estudio comparativo entre cada una de ellas. En dicho estudio, los métodos ICL y LEC presentan resultados que mejoran los del resto.

Los métodos deterministas descritos anteriormente, plantean criterios de selección de la clase de modelo M_k que posee el mejor conjunto de parámetros $\Theta^*(k)$. Sin embargo, en un modelo de mezclas, la distinción entre la selección de la clase de modelo (es decir, el número de núcleos) y la estimación del modelo (el conjunto de parámetros que describen el modelo), no está clara. Por ejemplo, una mezcla de tres componentes en la que la probabilidad a priori de uno de los núcleos es cero, no puede distinguirse de otra mezcla con dos componentes.

Otras técnicas más recientes, introducen modificaciones en el algoritmo EM clásico para fusionar y/o añadir núcleos dinámicamente siguiendo algún criterio. Puesto que cada autor emplea una nomenclatura diferente para especificar las ecuaciones del modelo, se ha preferido seguir en cada caso la nomenclatura empleada originalmente por sus respectivos autores. A continuación se realiza una revisión detallada de las técnicas, que por su funcionamiento, se asemejan más a la propuesta en el presente trabajo. Dichas técnicas podemos clasificarlas en: (i) técnicas que parten de un número inicial de núcleos elevado y realizan fusiones de los mismos hasta lograr el óptimo; (ii) técnicas que partiendo de un número reducido de núcleos (normalmente uno) van añadiendo nuevos a la mezcla; (iii) y técnicas que realizan ambos tipos de operaciones.

2.5.1. Técnicas basadas en la fusión de núcleos

Entre ellas encontramos [Figueiredo *et al.*, 1999] [Figueiredo y Jain, 2000] [Figueiredo y Jain, 2002]. Los autores proponen una modificación del algoritmo EM clásico, que consiste en inicializar con un número de núcleos aleatoriamente dispuestos. El *Principio de Longitud de Descripción Mínima* (MDL) [Rissanen, 1983] es aplicado iterativamente para eliminar alguno de los núcleos, hasta alcanzar el número óptimo. De este modo, no se emplea un criterio de selección del modelo para escoger uno entre un conjunto de modelos candidatos, sino que se integra la estimación de los parámetros y la selección

del orden del modelo en un sólo algoritmo.

La idea es emplear el algoritmo EM (para un número de núcleos k fijo) que permita obtener una secuencia de estimaciones de los parámetros de la muestra $\hat{\Theta}_{(k)}$, con $k = k_{min}, ..., k_{max}$. El valor óptimo de k debe obtenerse como aquel que minimiza una función de coste de la forma:

$$\hat{k} = \arg\min_{k}\{\mathcal{C}(\hat{\Theta}_{(k)}, k), k = k_{min}, ..., k_{max}\} \tag{2.13}$$

donde $\mathcal{C}(\hat{\Theta}_{(k)}, k)$ representa algún criterio de selección del orden del modelo y $\hat{\Theta}_{(k)}$ la estimación actual de los parámetros del modelo suponiendo que éste está compuesto por k núcleos. La función de coste deberá incluir tanto la maximización del logaritmo de la verosimilitud como un término adicional, cuya función es la de penalizar valores elevados de k. Una de las funciones que cumplen el criterio anterior es la que se basa en el *Principio de Longitud de Descripción Mínima* o criterio MDL, cuya función de coste es:

$$\mathcal{C}_{MDL}(\hat{\Theta}_{(k)}, k) = -L(\hat{\Theta}_{(k)}, \mathrm{y}_{obs}) + \frac{N(k)}{2}\log n, \tag{2.14}$$

donde $N(k)$ representa el número de parámetros necesarios para especificar una mezcla de k núcleos. Si denominamos d a la dimensión del problema, entonces para el caso general tenemos: $N(k) = (k-1)+k(d+d(d+1)/2)$. Donde $L(.)$ representa el logaritmo de la verosimilitud de los datos observados y con el conjunto actual de parámetros $\hat{\Theta}_{(k)}$.

En algunos casos, la utilización de este criterio genera estimaciones del número de parámetros del modelo por debajo del número óptimo [Kontkanen *et al.*, 1996] [Smyth, 1996]. Para evitar este problema se propone una modificación de la expresión anterior que penaliza en menor medida que MDL un aumento en el número de núcleos de la mezcla inicial. La función de coste de la ecuación 2.14, puede ser descompuesta como la suma de la longitud de código de los datos observados y_{obs} y la estimación actual de los parámetros de la mezcla $\hat{\Theta}_{(k)}$. Formalmente:

$$\mathcal{C}_{MDL}(\mathrm{y}_{obs}, \hat{\Theta}_{(k)}) = \mathcal{L}(\mathrm{y}_{obs}, \hat{\Theta}_{(k)}) = \mathcal{L}(\mathrm{y}_{obs}|\hat{\Theta}_{(k)}) + \mathcal{L}(\hat{\Theta}_{(k)}), \tag{2.15}$$

donde $\mathcal{L}(\mathrm{y}_{obs}|\hat{\Theta}_{(k)}) = -L(\hat{\Theta}_{(k)}, \mathrm{y}_{obs})$ es la longitud de código óptima de Shannon y $\mathcal{L}(\hat{\Theta}_{(k)})$ representa la longitud de código de precisión finita de los

parámetros $\hat{\Theta}_{(k)}$de la mezcla (de precisión real), por lo que se requiere que los valores sean truncados. Si la precisión no es suficiente, $\mathcal{L}(\hat{\Theta}_{(k)})$ tendrá un valor bajo, pero los parámetros codificados de esta forma diferirán en mucho de los óptimos y la primera parte de la expresión será mayor. Con una mayor resolución, los parámetros codificados se aproximarán a los óptimos, pero la segunda parte de la expresión será más elevada. En [Rissanen, 1983] se propone tomar como longitud óptima de código para los parámetros, en el caso de valores elevados de n, $(1/2)\log(n)$, resultando la expresión 2.14.

En la mayoría de problemas en los que MDL o BIC son empleados como criterio, todos los datos tienen la misma importancia para la estimación del conjunto de parámetros de la muestra. Sin embargo, éste no es el caso de los modelos de mezclas, en los que cada dato tiene asociado un peso α_m dependiente del núcleo m que lo haya generado. Teniendo en cuenta este hecho, el tamaño del conjunto de muestras asociado al núcleo θ_m sería $n\alpha_n$ en lugar de n, por lo que la expresión la ecuación 2.14 quedaría:

$$\begin{aligned} \mathcal{C}_{MDL}(\hat{\Theta}_{(k)}, k) &= -L(\hat{\Theta}_{(k)}, \mathrm{y}_{obs}) + \frac{k-1}{2}\log n + \frac{N(1)}{2}\sum_{m=1}^{k}\log(n\alpha_m) \\ &= -L(\hat{\Theta}_{(k)}, \mathrm{y}_{obs}) + \frac{N(k)}{2}\log n + \frac{N(1)}{2}\sum_{m=1}^{k}\log\alpha_m \end{aligned} \quad (2.16)$$

El tercer sumando de la ecuación anterior es negativo, por tanto, el criterio expresado de esta forma introduce una menor penalización ante un número superior de núcleos que el criterio original. $N(1)$ representa al número de parámetros de un núcleo.

La técnica puede ser aplicada tanto a modelos de mezclas gausianas como los que emplean otro tipo de núcleos, aunque toda la experimentación se realiza con mezclas gausianas. El algoritmo presenta buenos resultados en general, pero es sensible a la presencia de falsos positivos, es decir observaciones que no serían correctamente modeladas por ninguno de los componentes de la mezcla. Como posible solución al problema se plantea incluir un componente extra con alta varianza que describa la totalidad del conjunto de observaciones anómalas.

Como estrategia de inicialización en dimensiones bajas $(d = 1, 2)$ se emplea una mezcla inicial compuesta por k_{max} núcleos, uniformemente distri-

buidos sobre la región ocupada por las observaciones (definida por los valores mínimo y máximo en cada dimensión). Para dimensiones superiores, se crean igualmente k_{max} clusters iniciales, pero empleando k-means como técnica de inicialización. Si k_{max} es suficientemente grande, la técnica propuesta muestra escasa sensibilidad a los valores iniciales del modelo, siempre que se parta de un número inicial de núcleos suficientemente grande. La figura 2.3 muestra la evolución del algoritmo para un conjunto de observaciones pertenecientes a 3 núcleos en un contexto de estimación de densidad de probabilidad asociada a los datos. El algoritmo comienza con 30 núcleos uniformemente distribuidos y finaliza al llegar a un sólo núcleo, obteniendo como valor óptimo 3, que es el estado que genera mejor valor de la función objetivo sobre criterio MDL.

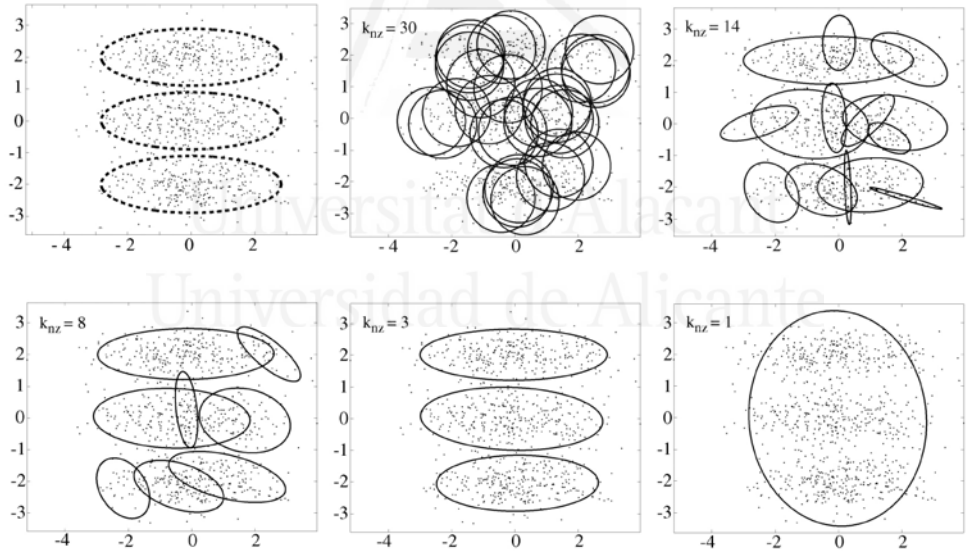

Figura 2.3: Evolución del algoritmo con estrategia de fusión de núcleos para un conjunto de muestras generado artificialmente en dos dimensiones. La primera imagen muestra el conjunto original de datos y el resto los diferentes estados intermedios del algoritmo. (Tomada de [Figueiredo y Jain, 2002])

2.5.2. Técnicas basadas en la introducción de nuevos núcleos

Otras técnicas parten de un número inicial de núcleos reducido para ir añadiendo progresivamente nuevos núcleos a la mezcla hasta alcanzar algún criterio definido previamente. En [Vlassis y Likas, 1999], se define una nueva medida denominada *kurtosis total*, basada en la kurtosis ponderada con las probabilidades a priori de cada uno de los núcleos de la mezcla. Esta medida proporciona información de la calidad del ajuste de la mezcla a los datos en cada paso del algoritmo. La kurtosis es una medida estadística definida para variables aleatorias uni-dimensionales. En el caso de distribuciones gausianas se verifica:

$$\int_{-\infty}^{+\infty} \left(\frac{x - \mu_j}{\sigma_j} \right)^4 p(x|j) dx = 3, \tag{2.17}$$

donde x representa una variable aleatoria uni-dimensional, j representa a un núcleo de la mezcla y μ_j y σ_j representan la media y la varianza del núcleo respectivamente. Aplicando la regla de Bayes y resolviendo la integral anterior por el método de Monte Carlo, para un conjunto de observaciones $x_i, i = 1, 2, ..., n$ de la variable aleatoria x se obtiene la expresión para estimar la kurtosis de un núcleo j de la mezcla:

$$k_j = \frac{\sum_{i=1}^{n} \left(\frac{x_i - \mu_j}{\sigma_j} \right)^4 p(j|x_i)}{\sum_{i=1}^{n} p(j|x_i)} - 3. \tag{2.18}$$

A partir de la expresión anterior, se calcula la *kurtosis total* como:

$$K_T = \sum_{j=1}^{K} \pi_j |k_j|, \tag{2.19}$$

que es la suma ponderada de cada una de las kurtosis de los núcleos de la mezcla con la probabilidad a priori de dicho núcleo. Esta medida se emplea para estimar si el ajuste es correcto. Valores próximos a cero indican que los núcleos individuales ajustan adecuadamente a los puntos de su vecindad, por lo que la mezcla completa constituye una buena aproximación a la densidad de probabilidad desconocida que seguían los datos.

Por el contrario, un valor elevado de la medida anterior indicaría que uno o varios núcleos no ajustan adecuadamente las observaciones próximas. En

ese caso, se selecciona el núcleo j que en mayor medida contribuye a incrementar la kurtosis total, es decir, el que tiene mayor valor para el producto $\pi_j|k_j|$,y se descompone en otros dos que deben ser posteriormente inicializados.

Los dos nuevos núcleos son creados con medias $\mu_j + \sigma_j$ y $\mu_j - \sigma_j$ respectivamente. Como varianza de los dos nuevos núcleos se mantiene la varianza original del núcleo de procedencia σ_j y las nuevas probabilidades a priori se establecen a la mitad de la probabilidad del núcleo original $\pi_j/2$.

Como estrategia de inicialización del algoritmo se comienza con un solo núcleo cuya media se corresponde con la de los datos y varianza establecida a un valor fijo, p.e. $0{,}5$. De este modo, el algoritmo realiza una maximización de la verosimilitud al mismo tiempo que trata de minimizar la kurtosis. El incremento en el número de núcleos de la muestra se detiene cuando la disminución en el valor de la kurtosis total no es superior a un umbral definido previamente.

El principal problema de la técnica es la sensibilidad a la presencia de falsos positivos al utilizar la kurtosis como medida del grado de gausianidad de cada núcleo, así como la limitación de su aplicación a problemas de una sola dimensión.

Posteriormente, en [Vlassis *et al.*, 2000] la misma técnica se extiende al caso multi-dimensional. Para ello se emplea la definición de kurtosis multidimensional [Mardia, 1970]:

$$\beta_j = \int_{-\infty}^{+\infty} \{(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\}^2 f(x;\phi_j) dx, \tag{2.20}$$

donde x es un vector que representa una observación, μ_j y Σ_j representan la media y la matriz de covarianza del núcleo j y ϕ_j representa el conjunto de parámetros asociado al núcleo j. Aplicando igualmente la regla de Bayes y sustituyendo los valores de las probabilidades a priori en cada paso del algoritmo EM, se obtiene la siguiente expresión para la kurtosis multidimensional que puede ser empleada como test de normalidad:

$$\beta_j = \frac{\sum_{i=1}^{n} p(j|x_i)\{(x_i-\mu_j)^T \Sigma_j^{-1} (x_i-\mu_j)\}^2}{\sum_{i=1}^{n} p(j|x_i)}. \tag{2.21}$$

En el caso de que el núcleo sea verdaderamente gausiano, el valor de β_j debería ser aproximadamente $d(d+2)$, donde d es la dimensión del problema.

Como estrategia de inicialización del nuevo núcleo introducido en la mezcla, se sigue el criterio heurístico basado en que, normalmente, la principal fuente de multi-modalidad (en oposición a gausianidad) aparecerá a lo largo de la dirección de máxima varianza del núcleo con peor valor de kurtosis, o lo que es lo mismo, la dirección principal de PCA (Principal Component Analysis). Por tanto, se requiere un análisis de autovalores y autovectores de la matriz de covarianza del núcleo seleccionado.

La introducción de un nuevo núcleo se basa en los resultados de [Lindsay, 1983], según el cual, si a partir de una mezcla con k componentes añadimos un nuevo componente $f_{k+1}(x;\phi^*)$ con probabilidad a priori $a \in (0,1)$ tal que la nueva mezcla pueda expresarse del siguiente modo:

$$p_{k+1}(x) = af_{k+1}(x;\phi^*) + (1-a)p_k(x), \tag{2.22}$$

siempre se obtendrá un incremento de la función logaritmo de la verosimilitud, a menos que ya se haya alcanzado el máximo de dicha función.

De esta forma, las ecuaciones para determinar los parámetros del nuevo núcleo se obtendrían del siguiente modo. La nueva media:

$$m^* = m_c \pm \sqrt{\lambda}(v_c + 0{,}1w), \tag{2.23}$$

donde λ es el mayor autovalor de la matriz de covarianza Σ_c, v_c es el autovalor correspondiente al autovector anterior y w es un vector que introduce una perturbación aleatoria a partir de una distribución gausiana d-dimensional. La nueva matriz de covarianza se define:

$$\Sigma^* = 0{,}25\lambda I_d, \tag{2.24}$$

donde I_d es la matriz identidad de dimensión d. Por último, la probabilidad a priori del nuevo núcleo a se establece a $0{,}5$. Según la ecuación 2.22, tras la introducción del nuevo componente, podemos considerar la muestra como si tuviera únicamente dos componentes, el primer componente sería el que se acaba de añadir, $f_{k+1}(x;\phi^*)$ y el segundo sería la muestra anterior $p_k(x)$. De este modo, se realizan pasos EM parciales para encontrar los parámetros del nuevo núcleo: a y ϕ^* que maximizan la nueva verosimilitud, manteniendo sin cambios los parámetros ajustados con anterioridad.

La nueva propuesta soluciona la limitación de la aplicación a problemas de una sola dimensión, pero todavía adolece del mismo problema que la original, y la consecución de un buen ajuste de la mezcla a los datos requiere que no existan falsos positivos que puedan condicionar el cálculo de la kurtosis multi-dimensional. La figura 2.4 muestra la evolución del algoritmo para un conjunto de datos en dos dimensiones en una mezcla de seis componentes.

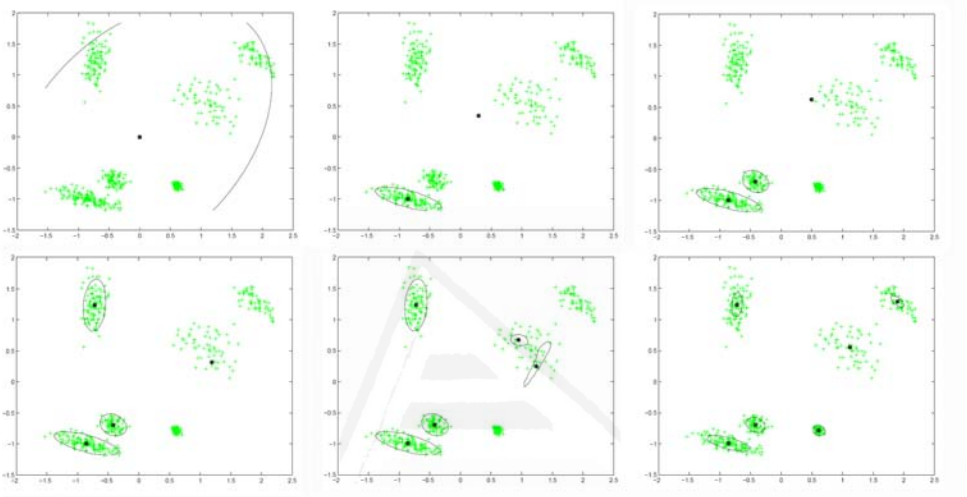

Figura 2.4: Evolución del modelo de mezcla para un conjunto de datos artificiales en 2-D con 6 núcleos. (Tomada de [Vlassis *et al.*, 2000]).

Para solucionar el inconveniente de la sensibilidad de la kurtosis a la presencia de falsos positivos, en [Vlassis y Likas, 2000] se plantea una solución voraz al problema, añadiendo sucesivamente nuevos componentes a la mezcla hasta lograr el número deseado k. Según los resultados teóricos obtenidos en [Lindsay, 1983] y [Li y Barron, 2000], bajo algunas suposiciones el ajuste de una mezcla empleando criterios de máxima verosimilitud puede realizarse de forma voraz. Si la inserción de cada nuevo componente se lleva a cabo de forma óptima, la mezcla obtenida de forma incremental se ajusta a los datos, al menos como la obtenida en 2.1. La consecuencia práctica de este resultado es que la tarea de ajustar una mezcla con k componentes puede ser reemplazada por la tarea más simple de ajustar sucesivamente muestras de dos componentes únicamente.

No obstante, los parámetros asociados a la nueva componente son funda-

mentales para lograr un ajuste correcto. Para la determinación de los mismos se plantea una búsqueda global entre todos los datos, seguida de varios pasos EM parciales hasta lograr la convergencia.

Formalmente, si añadimos una nueva componente $\phi(x;\theta)$ a una mezcla $f_k(x)$, podemos definir la nueva mezcla de la siguiente forma:

$$f_{k+1}(x) = (1-a)f_k(x) + a\phi(x;\theta), \tag{2.25}$$

con $a \in (0,1)$. De este modo, para una mezcla actual $f_k(x)$, el valor de a y el vector de parámetros θ del nuevo núcleo $\phi(x;\theta)$ deben ser seleccionados de modo que el nuevo valor del logaritmo de la verosimilitud:

$$\ell_{k+1} = \sum_{i=1}^{n} \log f_{k+1}(x_i) = \sum_{i=1}^{n} \log[(1-a)f_k(x_i) + a\phi(x_i;\theta)] \tag{2.26}$$

se maximice. Durante el proceso, los parámetros de $f_k(x)$ se mantienen constantes. De este modo, el problema original de ajustar un modelo de mezcla gausiana maximizando el logaritmo de la verosimilitud, ha sido sustituido por el aprendizaje sucesivo de una mezcla de dos componentes $f_{k+1}(x)$,en la que la primera componente es la mezcla anterior $f_k(x)$ y la segunda es el nuevo núcleo $\phi(x;\theta)$, con $\theta = (\mu, \Sigma)$, su media y matriz de covarianza. La búsqueda de los parámetros a, μ y Σ que maximizan la expresión 2.26 se realiza de la siguiente forma:

- **Búsqueda local**: Puesto que la nueva mezcla tiene dos componentes, el algoritmo EM puede ser empleado para buscar el máximo de ℓ_{k+1}, respecto a los parámetros a, m y S. Además, puesto que los parámetros de la mezcla anterior $f_k(x)$ permanecen invariables durante la localización del nuevo componente, los pasos EM pueden ser parciales, ajustando únicamente los parámetros del nuevo núcleo. Aunque se trata de un método simple y rápido, es todavía un método local y por tanto, sensible a a los valores iniciales del conjunto de parámetros a, m y S del nuevo núcleo, por lo que se necesita una estrategia adicional de búsqueda global.

- **Búsqueda global**: Para facilitar la búsqueda global sobre el conjunto de parámetros, se realiza una aproximación de Taylor de segundo grado

de la función logaritmo de la verosimilitud 2.26 en el punto $a = a_0$, con $a_0 = 0{,}5$. Como resultado se obtiene la siguiente expresión:

$$\hat{\ell}_{k+1} = \ell_{k+1}(a_0) - \frac{[\ell'_{k+1}(a_0)]^2}{2\ell''_{k+1}(a_0)}, \tag{2.27}$$

con ℓ'_{k+1} y ℓ''_{k+1} representando la primera y segunda derivadas de ℓ_{k+1} respecto de a. Maximizando la función cuadrática resultante con respecto a a se obtiene el siguiente valor óptimo para el parámetro a:

$$\hat{a} = \frac{1}{2} - \frac{1}{2}\frac{\sum_{i=1}^{n}\delta(x_i,\theta)}{\sum_{i=1}^{n}\delta(x_i,\theta)^2}, \tag{2.28}$$

donde

$$\delta(x_i,\theta) = \frac{f_k(x) - \phi(x;\theta)}{f_k(x) + \phi(x;\theta)}. \tag{2.29}$$

Si el valor obtenido para a cae fuera del intervalo (0,1), entonces se selecciona $\hat{a} = 0{,}5$ si $k = 1$ o $\hat{a} = 2/(k+1)$ si $k \geq 2$, según se especifica en [Li y Barron, 2000]. El procedimiento anteriormente descrito hace que la función verosimilitud definida en 2.26 sea independiente del valor del parámetro a. El siguiente paso es encontrar valores iniciales para μ y Σ. Una búsqueda global sobre el espacio de parámetros de todos los posibles $[\mu, \Sigma]$ no es factible, debido al elevado coste computacional, por lo que dicha búsqueda debe ser restringida.

Para el caso de la media μ, los posibles valores se limitan al conjunto de datos x. Para el caso de la matriz de covarianza Σ, se limita a una matriz diagonal de la forma $\Sigma = \sigma^2 I$, por lo tanto puede tratarse como una función de la distancia Euclídea entre cada punto x_i y la media μ. La distancia euclídea entre cada par de puntos $\|x_i - x_j\|$ es pre-calculada al comienzo del algoritmo.

Por último, como elección para el valor de la desviación típica σ se selecciona un valor dependiente del número de observaciones disponibles n y de la dimensión de las mismas d, según la recomendación propuesta en [Wand, 1994]:

$$\sigma = \beta \left[\frac{4}{(d+2)n} \right]^{\frac{1}{d+4}}, \tag{2.30}$$

con β un valor constante a establecer.

El planteamiento descrito anteriormente posee una complejidad temporal para cada evaluación de $\hat{\ell}_{k+1}$ de $O(n)$, siendo la complejidad total de la búsqueda global $O(n^2)$.

La condición de parada del algoritmo suele ser el alcanzar un número máximo de núcleos permitido k. Si se pretende estimar el número óptimo de componentes de la mezcla, los autores proponen ejecutar el algoritmo para un valor elevado de k y entonces seleccionar el valor óptimo $\hat{k}$ a partir de algún criterio de selección del orden del modelo como el *Principio de Longitud de Descripción Mínima* [Rissanen, 1983] citado con anterioridad.

2.5.3. Técnicas basadas en la fusión e introducción de núcleos

Para finalizar la revisión a otras técnicas previas, presentamos el algoritmo SMEM (*Split and Merge EM*) propuesto por [Ueda *et al.*, 2000], en el que durante el proceso de ajuste del modelo se realizan dos tipos de operaciones: división de un núcleo actual en otros dos y fusión de dos núcleos existentes en uno sólo, a partir de un criterio de selección de núcleos candidatos.

La idea de realizar operaciones de división y fusión de núcleos ya había sido propuesta con anterioridad en el análisis de modelos de mezclas gausianas desde el punto de vista bayesiano [Richardson y Green, 1997], donde se combinaban los movimientos anteriores con el método de *Markov Chain Monte Carlo*. No obstante, como se comentaba con anterioridad 1.2.1, el coste de los métodos de Markov es computacionalmente mucho más costosos que el algoritmo EM.

El funcionamiento del algoritmo es el siguiente: Tras realizar los pasos **E** y **M** y alcanzar la convergencia, se obtiene el nuevo conjunto de parámetros Θ y se seleccionan los núcleos candidatos i, j, k para las operaciones de división y fusión, según un criterio que será descrito posteriormente. A continuación se determinan los valores iniciales de los nuevos núcleos introducidos (ecuaciones 2.32 y 2.33) y se realizan pasos EM parciales hasta lograr la convergencia. Si como resultado se obtiene una mejora de la verosimilitud de la mezcla, se

acepta el cambio y continúa el proceso. Si no, se seleccionan nuevos candidatos según la lista ordenada y se repiten los pasos EM parciales. El algoritmo finaliza cuando no se consigue ninguna mejora en la verosimilitud tras seleccionar todos los posibles candidatos en las operaciones de división y mezcla.

Si definimos el modelo de mezcla de la siguiente forma:

$$p(x;\Theta) = \Sigma_{m=1}^{M} \alpha_m p_m(x;\theta_m), \tag{2.31}$$

donde α_m representa la probabilidad a priori del núcleo m y $\Theta = (\alpha_m, \theta_m), m = 1, ..., M$ es el conjunto de parámetros a estimar, tras una sucesión de pasos E y M, obtenemos el conjunto actual de parámetros Θ^*.

Si denominamos j y k a los dos núcleos candidatos a fusionar, generando el nuevo núcleo i' y denominamos k al núcleo candidato a descomponerse en otros dos j' y k', entonces los valores iniciales de los parámetros del nuevo núcleo i' resultado de la fusión serían una combinación lineal de de los núcleos originales antes de la mezcla:

$$\alpha_{i'} = \alpha_i^* + \alpha_j^*, \text{ y } \theta_{i'} = \frac{\alpha_i^*\theta_i^* + \alpha_j^*\theta_j^*}{\alpha_i^* + \alpha_j^*} \tag{2.32}$$

Por otro lado, los valores iniciales de los nuevos núcleos j' y k' resultado de la división, se obtendrían de la siguiente forma:

$$\alpha_{j'} = \alpha_{k'} = \frac{\alpha_k^*}{2}, \ \theta_{j'} = \theta_k^* + \varepsilon, \text{ y } \theta_{k'} = \theta_k^* + \varepsilon', \tag{2.33}$$

donde ε y ε' es un vector o matriz con una pequeña perturbación aleatoria. En el caso de mezclas gausianas, las matrices de covarianza de los nuevos núcleos $\Sigma_{j'}$ y $\Sigma_{k'}$ deben ser definidas positivas, por lo que en este caso sus valores iniciales podrían ser:

$$\Sigma_{j'} = \Sigma_{k'} = det(\Sigma_k^*)^{1/d} I_d, \tag{2.34}$$

con $det(\Sigma)$ el determinante de la matriz de covarianza y I_d la matriz identidad de dimensiones $d \times d$.

En cada paso del algoritmo se seleccionan los núcleos candidatos a fusionar y dividir, por lo que se necesita un criterio que permita ordenarlos de mayor a menor medida para, tal y como se comentaba en la especificación del algoritmo, seleccionarlos en ese orden y comprobar si mejora el valor del

logaritmo de la verosimilitud. Los criterios para la fusión y división son los siguientes:

- *Criterio para fusión:* De forma general, si existen muchos datos cuya probabilidad a posteriori de haber sido generados por dos núcleos diferentes i y j es muy similar, entonces esos dos núcleos deberían fusionarse en uno solo. Expresado formalmente:

$$J_{merge}(i,j;\Theta^*) = \frac{P_i(\Theta^*)^T P_j(\Theta^*)}{\|P_i(\Theta^*)^T\| \, \|P_j(\Theta^*)\|}, \tag{2.35}$$

  donde $P_i(\Theta^*) = (P(i|x_1;\Theta^*), ..., P(i|x_N;\Theta^*))^T \in \Re^N$ es un vector de dimension N que contiene las probabilidades a posteriori para el núcleo i. T representa la traspuesta del vector y $\|.\|$ representa su norma Euclídea. Cuanto mayor sea el valor de la expresión anterior, mejores candidatos para fusionar serán los núcleos i y j.

- *Criterio para división:* Como criterio para seleccionar núcleos candidatos a ser divididos se emplea la *Divergencia local de Kullback*:

$$J_{split}(k;\Theta^*) = \int f_k(x;\Theta^*) \log \frac{f_k(x;\Theta^*)}{p_k(x;\theta_k^*)} dx, \tag{2.36}$$

  que representa la divergencia entre dos distribuciones de probabilidad: la densidad local de los datos alrededor del núcleo k definida por $f_k(x)$ y la densidad de probabilidad de dicho núcleo especificada por los valores actuales del conjunto de parámetros Θ^*. La densidad local de los datos alrededor del núcleo k se define como:

$$f_k(x;\Theta^*) = \frac{\sum_{n=1}^{N} \delta(x - x_n) P(k|x_n;\Theta^*)}{\sum_{n=1}^{N} P(k|x_n;\Theta^*)}, \tag{2.37}$$

  La expresión anterior representa la distribución empírica asociada a los datos, ponderada con las probabilidades a posteriori de cada núcleo k. El núcleo con mayor valor de $J_{split}(k;\Theta^*)$ posee la peor estimación de densidad de probabilidad a su alrededor, por lo que debería descomponerse en dos.

Una vez definidos los criterios para fusión y división de núcleos, es necesario ordenar los candidatos según el siguiente criterio: primero se establece el orden de los candidatos a fusionar según J_{merge}; después para cada par de candidatos a fusionar $\{i,j\}_c$, se ordenan los candidatos a dividir $\{k\}_c$, excluyendo los núcleos $\{i,j\}_c$. Combinando los resultados y renumerando, se obtienen las tripletas: $\{i,j,k\}_c$, $c = 1, ..., M(M-1)(M-2)/2$, con M el número núcleos de la mezcla.

A modo de resumen, el algoritmo descrito consta de los siguientes pasos:

1. Ejecutar el algoritmo EM clásico para actualizar los valores del conjunto de parámetros Θ hasta lograr la convergencia, obteniendo Θ^* como valor actual de los parámetros y el correspondiente logaritmo de la verosimilitud tras finalizar.

2. Ordenar los candidatos para división y fusión según la estimación actual de parámetros Θ^* y los criterios descritos anteriormente en 2.35 y 2.36.

3. Para cada tripleta de candidatos, realizar el proceso de fusión y división según las ecuaciones 2.32, 2.33 y 2.34. Realizar pasos EM parciales para los nuevos núcleos $\{i,j,k\}_c$ y a continuación ejecutar nuevamente el EM completo hasta la convergencia. Si el valor del logaritmo de la verosimilitud obtenido tras el proceso mejora el anterior, se aceptan los cambios. Si no se selecciona a los siguientes candidatos de la lista y se vuelve al paso 2.

4. Finalizar con Θ^* como conjunto de parámetros final del modelo.

El método propuesto es aplicado en problemas de estimación de densidad de probabilidad, reducción de dimensionalidad, compresión de imágenes y reconocimiento de patrones. No obstante, a pesar de que las operaciones de fusión y división de núcleos reducen el problema de la sensibilidad del algoritmo EM a la inicialización, el número de núcleos de la muestra se mantiene constante durante todo el proceso, quedando el problema de la estimación del número óptimo de núcleos como un trabajo futuro. En la figura 2.5 se muestra la evolución del algoritmo en el contexto de estimación de densidad de probabilidad asociada a un conjunto de datos en dos dimensiones.

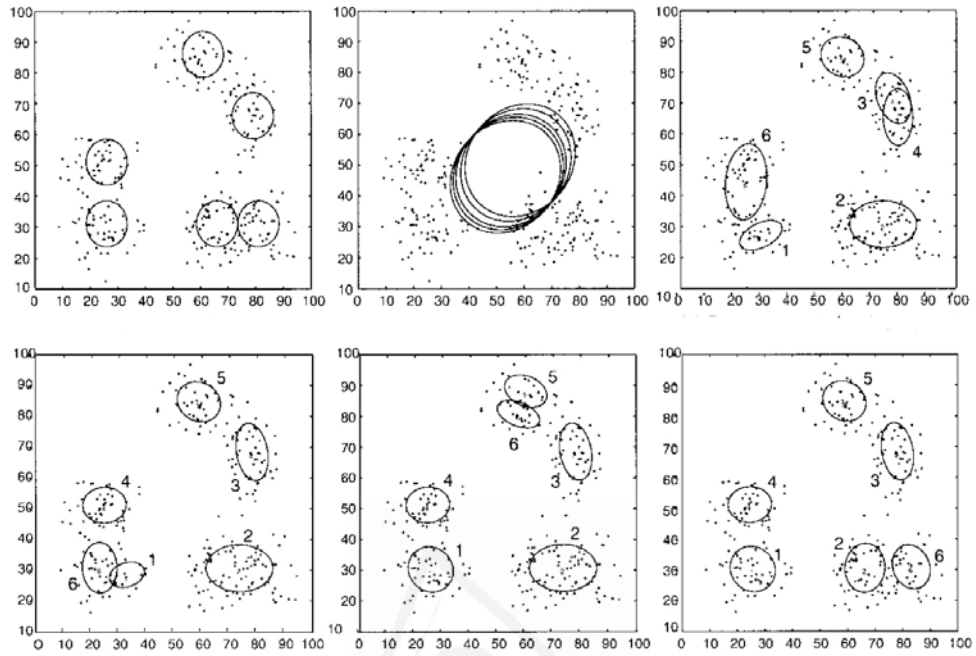

Figura 2.5: Evolución del algoritmo SMEM para un conjunto artificial de datos en dos dimensiones con seis núcleos en un problema de estimación de densidad de probabilidad. El número de componentes de la mezcla permanece constante durante todo el proceso, aunque en sucesivas iteraciones se procede a la fusión y división de los núcleos que cumplen con los criterios descritos anteriormente. La primera imagen muestra la mezcla real. La 2ª muestra la inicialización de los núcleos. La 3ª, 4ª y 5ª la evolución del algoritmo en las iteraciones 141, 186 y 212 respectivamente. La última imagen muestra el resultado del ajuste realizado. (Tomada de [Ueda *et al.*, 2000]).

Posteriormente, en [Zhang *et al.*, 2003] se propone una modificación del proceso de inicialización de los parámetros de los nuevos núcleos descrita en el método anterior y se realizan experimentos de aplicación del modelo a la segmentación de imágenes en color. La imagen de la figura 2.6 muestra los resultados de segmentación obtenidos con la aplicación del método. Las columnas 1 y 3 de cada fila muestran las imágenes originales, mientras que las columnas 2 y 4 muestran los resultados de segmentación obtenidos tras la aplicación del algoritmo. El número de núcleos ha sido fijado a priori para cada uno de los ejemplo y los resultados mostrados se han conseguido tras un post-proceso empleando análisis de componentes conectadas. Además,

las imágenes se han transformado del espacio original RGB a YUV antes de aplicar la técnica. Posteriormente las imágenes resultantes son pasadas nuevamente al espacio RGB antes de ser representadas. En el capítulo 4 llevaremos a cabo nuestros propios experimentos de segmentación directamente sobre el espacio RGB y sin llevar a cabo ningún tipo de post-proceso.

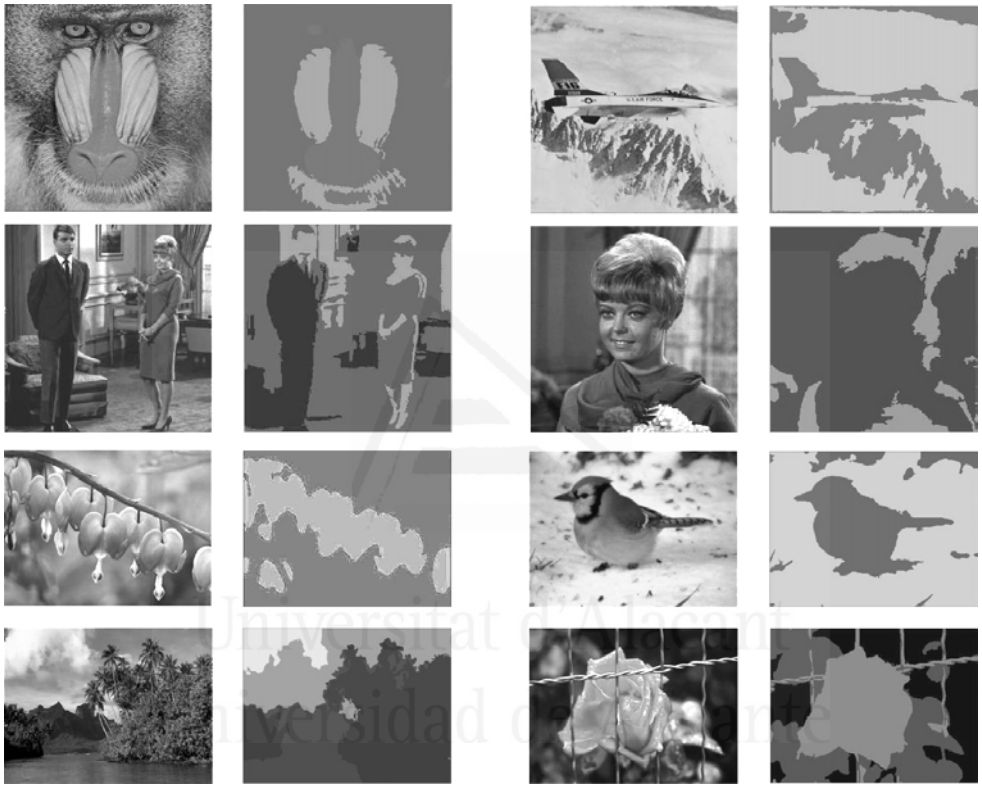

Figura 2.6: Aplicación del algoritmo SMEM al problema de la segmentación de imágenes en color con un nuevo criterio de introducción de núcleos al sistema basado en la descomposición de las matrices de covarianza. El nuevo método mejora los resultados obtenidos por el algoritmo SMEM original, aunque el número de núcleos del modelo no varía durante la ejecución del algoritmo y debe ser igualmente establecido previamente. (Obtenida de [Zhang *et al.*, 2003]).

La propuesta original de [Ueda *et al.*, 2000] para llevar a cabo la separación de los núcleos es un procedimiento heurístico sin soporte teórico, en el que los valores iniciales de las nuevas medias se definen de forma independiente a las matrices de covarianza. Puesto que media y matriz de covarianza

representan el primer y el segundo momento de la distribución, parece lógico tratarlos de forma conjunta. Por ello, los autores proponen una serie de ecuaciones, cuya solución proporciona los parámetros de los nuevos núcleos, preservando los dos primeros momentos de la distribución.

Para el procedimiento de fusión, si asumimos que los núcleos i y j generan el nuevo núcleo i', los parámetros de dichos núcleos deberían estar relacionados según las expresiones:

$$\pi_{i'} = \pi_i + \pi_j \tag{2.38}$$

$$\pi_{i'} p(x|i') = \pi_i p(x|i) + \pi_j p(x|j) \tag{2.39}$$

$$\pi_{i'}(\Sigma_{i'} + \mu_{i'}\mu_{i'}^T) = \pi_i(\Sigma_i + \mu_i\mu_i^T) + \pi_j(\Sigma_j + \mu_j\mu_j^T) \tag{2.40}$$

A las ecuaciones anteriores se las denomina ecuaciones de fusión. El cálculo de los nuevos parámetros es directo, pues la operación de fusión es un *well-posed problem*. Por otro lado, puesto que la división es el proceso contrario a la fusión, el núcleo k debe ser descompuesto en dos nuevos núcleos a los que denominamos j' y k'. De este modo, podemos expresar las ecuaciones de división como:

$$\pi_k = \pi_{j'} + \pi_{k'} \tag{2.41}$$

$$\pi_k \mu_k = \pi_{j'}\mu_{j'} + \pi_{k'}\mu_{k'} \tag{2.42}$$

$$\pi_k(\Sigma_k + \mu_k\mu_k^T) = \pi_{j'}(\Sigma_{j'} + \mu_{j'}\mu_{j'}^T) + \pi_{k'}(\Sigma_{k'} + \mu_{k'}\mu_{k'}^T) \tag{2.43}$$

En este caso, se trata de un *ill-posed problem*, puesto que el número de ecuaciones es menor que el número de incógnitas, difícil de resolver con datos multi-dimensionales para los que el número de parámetros a establecer es elevado, con la exigencia añadida de que las matrices de covarianza deben ser definidas positivas.

Claramente, la forma de calcular los valores iniciales de los parámetros de los nuevos componentes de la mezcla propuesto por [Ueda *et al.*, 2000] no es

solución a las ecuaciones anteriores. Para obtener dicha solución, y aprovechando la circunstancia de que las matrices de covarianza son definidas positivas, se proponen dos métodos basados en la descomposición de las matrices de covarianza: *Descomposición en Valores Singulares* (SVD)[1] y *Descomposición de Cholesky* [Golub y Lan, 1996] que sí son una solución a las ecuaciones anteriores.

En el caso de SVD, las matrices de covarianza Σ_k, $\Sigma_{j'}$ y $\Sigma_{k'}$, pueden ser descompuestas respectivamente en función de los autovalores y autovectores de dichas matrices. El método de descomposición está basado en un teorema [Golub y Lan, 1996] según el cual, para cualquier matriz de covarianza Σ, es posible encontrar otra matriz $A = [a_1, a_2, ..., a_n]$, tal que $\Sigma = AA^T = \Sigma_{j=1}^{n} a_j a_j^T$ y $a_i^T a_j = 0$ si $i \neq j$ y $a_i^T a_j = \lambda_i$ si $i = j$, con $\lambda_j > 0$ los autovalores de la matriz de covarianza y $j = 1, 2, ...n$.

A partir del teorema anterior, las matrices de covarianza pueden ser definidas de la siguiente forma $\Sigma_k = A_k A_k^T$, $\Sigma_{j'} = A_{j'} A_{j'}^T$ y $\Sigma_{k'} = A_{k'} A_{k'}^T$, con $A_k = [a_1^k, a_2^k, ..., a_n^k]$, $A_{j'} = [a_1^{j'}, a_2^{j'}, ..., a_n^{j'}]$ y $A_{k'} = [a_1^{k'}, a_2^{k'}, ..., a_n^{k'}]$. De esta manera, el problema de determinar las nuevas matrices de covarianza $\Sigma_{j'}$ y $\Sigma_{k'}$ dado Σ_k se convierte en determinar $A_{k'}$, $A_{j'}$ dado A_k.

$$\pi_{j'} = \pi_k \alpha, \;\; \pi_{k'} = \pi_k (1 - \alpha) \tag{2.44}$$

$$\mu_{j'} = \mu_k - \sqrt{\frac{\pi_{k'}}{\pi_{j'}}} \upsilon a_l^{(k)}, \;\; \mu_{k'} = \mu_k + \sqrt{\frac{\pi_{j'}}{\pi_{k'}}} \upsilon a_l^{(k)} \tag{2.45}$$

$$a_m^{j'} = \begin{cases} \sqrt{\beta(1 - \upsilon^2)\frac{\pi_k}{\pi_{j'}}} a_m^k & \text{si } m = l \\ \sqrt{\frac{\pi_{k'}}{\pi_{j'}}} a_m^k & \text{si } m \neq l \end{cases} \tag{2.46}$$

$$a_m^{k'} = \begin{cases} \sqrt{(1 - \beta)(1 - \upsilon^2)\frac{\pi_k}{\pi_{k'}}} a_m^k & \text{si } m = l \\ \sqrt{\frac{\pi_{j'}}{\pi_{k'}}} a_m^k & \text{si } m \neq l \end{cases} \tag{2.47}$$

con $l \in \{1, 2, ..., n\}$, y α, υ y $\beta \in (0, 1)$. Aplicando los resultados del teorema anterior sobre las expresiones de las ecuaciones 2.46 y 2.47, se puede obtener la expresión para las nuevas matrices de covarianza $\Sigma_{j'}$ y $\Sigma_{k'}$:

[1] proviene del término inglés *Singular Value Decomposition*

$$\Sigma_{j'} = \frac{\pi_{k'}}{\pi_{j'}}\Sigma_k + (\beta - \beta\upsilon^2 - 1)\frac{\pi_k}{\pi_{j'}} + a_l^k(a_l^k)^T \tag{2.48}$$

$$\Sigma_{k'} = \frac{\pi_{j'}}{\pi_{k'}}\Sigma_k + (\beta\upsilon^2 - \beta - \upsilon^2)\frac{\pi_k}{\pi_{k'}} + a_l^k(a_l^k)^T \tag{2.49}$$

De las expresiones anteriores se deduce que las nuevas matrices de covarianza son definidas positivas. Además los parámetros $\mu_{j'}$, $\mu_{k'}$, $\Sigma_{j'}$ y $\Sigma_{k'}$ determinados por las ecuaciones 2.45, 2.48 y 2.49 son soluciones a las ecuaciones 2.42 y 2.43.

El valor de l debería ser escogido de forma aleatoria entre $\{1, 2, ..., n\}$. Sin embargo, en la práctica se toma $l = 1$, con $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Es decir, en la descomposición de un núcleo k, los nuevos núcleos j' y k' se introducen en la dirección de máxima variabilidad de dicho núcleo (representada por el autovector con mayor autovalor de la matriz de covarianza Σ_k).

La descomposición de Cholesky es el método con menor coste computacional para resolver la inversa de una matriz simétrica y definida positiva:

$$\Sigma_k = L_k L_k^T, \tag{2.50}$$

con L_k una matriz triangular superior con elementos positivos en la diagonal. El método para obtener los parámetros de los nuevos núcleos es similar al anterior, pero en este caso los vectores a_l^k, $a_l^{j'}$ y $a_l^{k'}$ son reemplazados por el l-ésimo vector columna de las matrices L_k, $L_{j'}$ y $L_{k'}$. Según las ecuaciones 2.46 y 2.47, la expresión de las matrices anteriores quedaría:

$$\begin{aligned} L_{j'} &= [a_1^{j'}, a_2^{j'}, .., a_n^{j'}] \\ &= [a_1^k, a_2^k, .., a_n^k] \times diag\left\{\sqrt{\frac{\pi_{k'}}{\pi_{j'}}}, .., \sqrt{\beta(1-\upsilon^2)\frac{\pi_k}{\pi_{j'}}}, .., \sqrt{\frac{\pi_{k'}}{\pi'_j}}\right\} \end{aligned} \tag{2.51}$$

$$\begin{aligned} L_{k'} &= [a_1^{k'}, a_2^{k'}, .., a_n^{k'}] \\ &= [a_1^k, a_2^k, .., a_n^k] \times diag\left\{\sqrt{\frac{\pi_{j'}}{\pi_{k'}}}, .., \sqrt{\beta(1-\upsilon^2)\frac{\pi_k}{\pi_{k'}}}, .., \sqrt{\frac{\pi_{j'}}{\pi'_k}}\right\} \end{aligned} \tag{2.52}$$

Si las matrices de covarianza son diagonales, las dos descomposiciones (SVD y Cholesky) generan los mismos resultados. Los resultados obtenidos

mediante esta técnica pueden ser considerados como una extensión del método propuesto originalmente por [Richardson y Green, 1997] para problemas en espacios multi-dimensionales.

Capítulo 3

Algoritmo EM para mezclas gausianas basado en entropía

En este capítulo presentamos nuestro algoritmo EM para modelos de mezclas gausianas basado en entropía. Comenzaremos con la definición formal de la entropía de una densidad de probabilidad y sus características, principalmente las que la hacen adecuada para medir el grado de gausianidad o normalidad de un conjunto de datos. Posteriormente, realizaremos una descripción detallada de las diferentes técnicas propuestas que permiten su estimación a partir de un conjunto de observaciones o datos. Finalizaremos con la propuesta de un algoritmo EM, que empleando la medida anterior y diferentes criterios de parada, puede ajustar correctamente un modelo de mezclas gausianas partiendo de un sólo núcleo inicial, así como determinar el número óptimo u orden del modelo.

3.1. Introducción

En este trabajo se emplean dos técnicas diferentes para la estimación de la entropía asociada a los distintos núcleos de la muestra, cada una de ellas adecuada a la resolución de un tipo de problema en particular: Método basado

en Ventanas de Parzen [Parzen, 1962] y Método basado en *Entropic Spanning Graphs*. El primer método requiere estimar previamente la densidad de probabilidad asociada a los datos, mientras que el segundo realiza una estimación directa de la entropía sin necesidad de estimar previamente la densidad de probabilidad.

A continuación presentamos la técnica que permite ajustar los valores iniciales de los parámetros de los nuevos núcleos introducidos.

Por último, se detallará el algoritmo propuesto y dos criterios de parada para la determinación del número óptimo de núcleos del modelo, uno basado también en la entropía promedio de los datos y otro basado en el *Principio de Mínima Descripción*. Dado que se trata de un algoritmo EM para modelos de mezclas gausianas que permite introducir dinámicamente nuevos núcleos aplicando un criterio basado en entropía, lo hemos denominado EBEM: *Entropy-based EM*.

3.2. Medidas de gausianidad

En la literatura pueden encontrarse diferentes medidas del grado de normalidad o gausianidad asociado a un conjunto de datos. Además de la entropía, que es la medida empleada en el presente trabajo, diversas disciplinas han empleado medidas estadísticas como la *kurtosis* y otras derivadas de la entropía como la *entropía negativa o negentropy*, ampliamente utilizada en el contexto del Análisis de Componentes Independientes (ICA)[1] [Hyvarinen, 1998] [Hyvarinen y Oja, 2000] [Miller y Fisher, 2003]. Seguidamente realizamos un resumen de las más importantes:

3.2.1. Kurtosis

La kurtosis de una distribución de probabilidad es una medida estadística conocida también como momento de orden cuatro. Dada una variable aleatoria x, de la que se dispone de un conjunto de N observaciones x_i con media μ y varianza σ, la kurtosis k se define como:

[1] abreviatura de las siglas inglesas Independent Component Analysis

$$k = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu}{\sigma}\right)^4 - 3. \tag{3.1}$$

Para el caso de distribuciones gausianas el valor de la kurtosis es cero, por lo que dicha medida puede ser empleada como medida del grado de normalidad de una distribución de probabilidad. El principal inconveniente del empleo de esta medida es la gran sensibilidad que muestra a la presencia de falsos positivos en el conjunto de observaciones de la variable, así como el hecho de que se define inicialmente sólo para variables aleatorias de una sola dimensión. Esta medida es empleada en los trabajos iniciales de [Vlassis y Likas, 1999] y [Vlassis *et al.*, 2000].

3.2.2. Entropía negativa

La entropía negativa [Comon, 1994] ha sido empleada como medida del grado de gausianidad asociado a un conjunto de observaciones en el contexto de *Análisis de Componentes Independientes* (ICA) [Girolami y Fyfe, 1997]. La medida se propone como una alternativa a las medidas estadísticas basadas en la estimación de los momentos de orden tres y cuatro para distribuciones aproximadamente simétricas y mesocúrticas y se plantea en problemas pertenecientes a espacios de una sola dimensión:

$$J(p_u) = H(p_g) - H(p_u), \tag{3.2}$$

donde $H(p_u)$ es la entropía asociada a un conjunto de datos u y $H(p_g)$ representa la entropía equivalente de una distribución gausiana con igual media y covarianza que p_u. Puesto que, como se verá más adelante, el segundo teorema de Gibbs [Jones y Sibson, 1987] demuestra que una distribución gausiana maximiza la entropía sobre todas las distribuciones no gausianas de igual varianza, la entropía negativa así definida es siempre positiva para distribuciones no normales.

No obstante, es necesario realizar una estimación de $H(p_u)$ por alguno de los métodos que serán expuestos a continuación.

3.3. Estimación de la entropía

El concepto de entropía [Shannon, 1948] [Cover y Thomas, 1991] fue desarrollado originalmente por físicos en el contexto de equilibrio en Termodinámica y más tarde extendido a mecánica estadística. Por último, Shannon incluyó el concepto en la *Teoría de la Información* como uno de los principios fundamentales. En este trabajo estamos interesados en esta última definición y a la entropía definida de esta forma se la conoce como *Entropía de Shannon*.

La idea del concepto básico de entropía en Teoría de la Información está relacionada con la incertidumbre asociada a cualquier experimento o señal aleatoria. Si consideremos como señal una cadena de caracteres extraída de un texto escrito en español, éste se habrá codificado a partir de un conjunto de letras, espacios y signos de puntuación. Puesto que estadísticamente la frecuencia de aparición de algunos caracteres (por ejemplo la letra **w**) es muy baja, mientras otros son más comunes (como la letra **a**), la cadena de caracteres es menos aleatoria de lo que en un principio se podría pensar. Obviamente, no es posible predecir con exactitud cuál será el siguiente símbolo en la cadena, dado su carácter aleatorio, pero si es posible definir una medida para cuantificar esa aleatoriedad: la entropía.

Shannon, en su trabajo original, establece algunas restricciones que debe cumplir la medida anterior:

- La medida de información debe ser proporcional (continua). Es decir, una modificación en la probabilidad de aparición de uno de los elementos de la señal no debe variar en exceso el valor de la entropía.
- Si todos los elementos de la señal tienen la misma probabilidad de aparición, entonces la incertidumbre es máxima y por tanto la entropía será también máxima.

Según esto, para el caso uni-dimensional, una distribución de probabilidad *picuda* posee una entropía muy baja, pues la incertidumbre asociada a la misma también es muy baja: existen unos pocos valores con una probabilidad de aparición muy alta y otros con probabilidad de aparición casi nula. Por el contrario, una distribución de probabilidad *homogénea* posee una entropía muy alta, pues casi todos los valores tienen la misma probabilidad asociada y

por tanto la incertidumbre es muy elevada. En la figura 3.1 se pueden observar dos ejemplos de distribuciones discretas uni-dimensionales con diferente valor de varianza σ^2. La distribución de la izquierda posee mayor valor de entropía que la de la derecha.

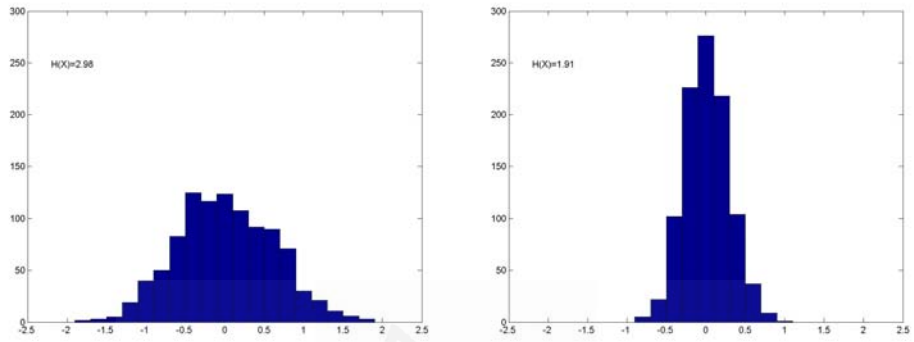

Figura 3.1: Ejemplo de dos histogramas junto con sus valores de entropía obtenidos mediante la generación de 1000 observaciones de dos distribuciones gausianas de varianzas $\sigma^2 = 0{,}4$ (izquierda) y $\sigma^2 = 0{,}08$ (derecha). Cuanto más compacta es la distribución, menor es la entropía.

Formalmente, para una variable aleatoria discreta Y con $y_1, ..., y_N$ el conjunto de posibles valores que puede tomar, se define la entropía de Shannon como:

$$H(Y) = -E_y[\log(P(Y))] = -\sum_{i=1}^{N} P(Y = y_i) \log P(Y = y_i). \tag{3.3}$$

Para el caso de distribuciones continuas, se denomina *Entropía Diferencial* y se define como:

$$H(y) = -\int p(y) \log p(y), \tag{3.4}$$

con $p(y)$ la función densidad de probabilidad asociada a y. Un resultado fundamental de la *Teoría de la Información*, conocido como segundo teorema de Gibbs [Jones y Sibson, 1987], es que las distribuciones gausianas poseen el máximo valor de entropía de entre todas las variables de igual varianza. Para demostrar la afirmación anterior, debemos maximizar la función de entropía

de la expresión 3.4 entre $(-\infty, +\infty)$ con las siguientes restricciones para el caso uni-dimensional:

$$\int_{-\infty}^{+\infty} p(y)dy = 1 \tag{3.5}$$

$$\int_{-\infty}^{+\infty} yp(y)dy = \mu \tag{3.6}$$

$$\int_{-\infty}^{+\infty} (y-\mu)^2 p(y)dy = \sigma^2 \tag{3.7}$$

Introduciendo multiplicadores de Lagrange λ_1, λ_2 y λ_3 a cada una de las restricciones anteriores y aplicando cálculo de variaciones para maximizar la función:

$$\int_{-\infty}^{+\infty} p(y)\{\ln p(y) + \lambda_1 + \lambda_2 y + \lambda_3 (y-\mu)^2\}dy - \lambda_1 - \lambda_2\mu - \lambda_3\sigma^2 \tag{3.8}$$

obtenemos:

$$p(y) = \exp\{-1 - \lambda_1 - \lambda_2 y - \lambda_3 (y-\mu)^2\} \tag{3.9}$$

Sustituyendo hacia atrás la expresión anterior en la definición de las restricciones se obtiene la distribución que maximiza la entropía con la forma:

$$p(y) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}, \tag{3.10}$$

que coincide con la distribución gausiana para una dimensión con parámetros de media y varianza μ y σ^2 respectivamente. Según esto, la entropía asociada a la distribución de los datos representados por un núcleo cualquiera de la mezcla, debería alcanzar un valor máximo cuando dicha distribución sea verdaderamente gausiana. Este valor máximo de entropía para una distribución gausiana uni-dimensional se obtendría a partir de las ecuaciones 3.3 y 3.10:

$$\begin{aligned} H_{max}(y) &= -E_y[\log(P(y))] \\ &= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}E\left[\frac{(x-\mu)^2}{\sigma^2}\right] \end{aligned}$$

$$= \frac{1}{2}\left[\log(2\pi\sigma^2) + 1\right] = \frac{1}{2}\log(2\pi\sigma^2 e) \tag{3.11}$$

siendo *log* el logaritmo neperiano. Del mismo modo se obtendría la expresión para el caso d-dimensional:

$$H_{max}(y) = \frac{1}{2}\log[(2\pi e)^d|\Sigma|]. \tag{3.12}$$

De las ecuaciones anteriores se desprende que la entropía de una distribución gausiana depende únicamente de la varianza σ^2 para el caso unidimensional, o de la matriz de covarianza Σ para el caso d-dimensional.

De este modo, si se dispusiera de una técnica para estimar la entropía de la distribución de probabilidad asociada a los datos representados por un núcleo de la mezcla, se podría comparar con el máximo teórico obtenido con las expresiones 3.10 o 3.12. Cuanto más próximos estén ambos valores, mayor grado de gausianidad tendrá el núcleo objeto de estudio y por tanto más preciso será el ajuste de las observaciones próximas a éste. Por el contrario, si ambos valores difieren la distribución asociada a los datos no será gausiana, si no que presentará rasgos de multi-modalidad en general y por tanto no será suficiente con un sólo núcleo para modelar correctamente los datos. En este caso, será necesario introducir un nuevo núcleo a la mezcla que permita realizar un ajuste más preciso de las observaciones.

La estimación de la entropía de Shannon de una densidad de probabilidad a partir de un conjunto de observaciones de la misma ha sido ampliamente estudiada en el pasado [Beirlant *et al.*, 1996] [Paninski, 2003] [Viola, 1995] [Viola *et al.*, 1996] [Hyvarinen y Oja, 2000] [Wolpert y Wolf, 1995] [Donoho, 1993] [Hall y Morton, 1993] [Joe, 1989]. La mayoría de las técnicas de estimación no paramétricas están basadas en la estimación previa de la función densidad de probabilidad asociada a los datos, seguida de una sustitución de dicha estimación en la expresión de la entropía. Este método ha sido ampliamente utilizado y se le conoce como *plug-in*.

Otros métodos de estimación, menos empleados, son *Sample Spacing Estimators*, restringidos únicamente a problemas de una sola dimensión y estimaciones basadas Vecinos más Cercanos. Ver [Beirlant *et al.*, 1996] para una revisión detallada de estos métodos.

En [Hero y Michel, 2002] se propone un método alternativo para la estimación de entropía y divergencia basado en la utilización de *Entropic Spanning Graphs*. A este método se le conoce como *non plug-in*, puesto que la entropía es directamente estimada a partir del conjunto de observaciones sin realizar una estimación de la densidad de probabilidad asociada a las mismas. En las siguientes secciones presentamos dos aproximaciones diferentes para la estimación de la entropía que se emplearán posteriormente en el algoritmo: una técnica *plug-in* basada en *Ventanas de Parzen* [Parzen, 1962] y otra *non plug-in* basada en *Entropic Spanning Graphs*.

Cada método tiene sus propias ventajas e inconvenientes: El método *plug-in* basado en las ventanas de Parzen tiene como principal problema la dimensión infinita de los espacios asociados a densidades de probabilidad no restringida. Más concretamente, la calidad de la estimación es pobre sin la restricción de suavidad de la función a estimar. Además, generalmente no existen estimadores de la densidad de probabilidad no sesgados o bien presentan una gran varianza y por tanto una gran sensibilidad a la presencia de falsos positivos. Por último, en el caso de problemas de dimensión elevada, la resolución de la integral requerida para evaluar la entropía podría ser tremendamente compleja y el número de observaciones necesarias para realizar una estimación precisa muy elevado, debido a la maldición de la dimensionalidad. Por el contrario, las técnicas basadas en grafos presentan una convergencia asintótica más rápida, especialmente para densidades abruptas y para espacios de dimensión alta [Hero *et al.*,]. Además, se elimina la necesidad de seleccionar y ajustar parámetros como el tamaño de celda del histograma o la anchura de núcleo para la estimación de densidad.

El principal inconveniente de de los métodos basados en *Entropic Spanning Graphs* es que no realizan una estimación directa de la entropía de Shannon, por lo que es necesario desarrollar una nueva técnica que permita aproximar esta última a partir de la estimación obtenida.

3.3.1. Método de las ventanas de Parzen

El método de estimación de densidades de probabilidad de las Ventanas de Parzen [Parzen, 1962] es una técnica no paramétrica, puesto que no asume ninguna forma a priori de la distribución, empleando directamente los

datos disponibles para formular el modelo. Para una clase cualquiera k, los estimadores de Parzen responden a la cuestión de qué información acerca de la densidad de probabilidad $P(Y|k)$ proporciona cada observación individualmente. Si denominamos y_i a la i-ésima observación de una clase k, cuya densidad de probabilidad asociada queremos estimar, podemos afirmar lo siguiente:

1. $P(y_i|k) > 0$.
2. Si suponemos que $P(Y|k)$ es continua, ésta tomará valores positivos y distintos de cero en una inmediata vecindad de y_i.
3. Cuanto más nos alejemos de y_i menos puede afirmarse sobre $P(Y|k)$ basándonos únicamente en y_i

Con estas afirmaciones, fácilmente asumibles, podemos concluir que la información acerca de $P(Y|k)$ obtenida a partir de y_i puede representarse mediante una función denominada núcleo. Esta función se expresa como $K(y, y_i)$, está centrada en el punto y_i y alcanza un máximo en él, decreciendo monótonamente a medida que se incrementa la distancia entre y y y_i. Más concretamente, las características deseables de la función $K(.)$ serían las siguientes:

1. $K(Y, Z)$ debería alcanzar el máximo para $Y = Z$.
2. $K(Y, Z)$ debería ser aproximadamente cero para valores de Y distantes de Z.
3. $K(Y, Z)$ debería ser una función suave y continua y decrecer monótonamente conforme aumenta la distancia entre Y y Z.
4. Si $K(Y_1, Z) = K(Y_2, Z)$, entonces Y_1 y Y_2 deberían tener el mismo grado de similitud con Z.

Una vez establecida la aportación de cada observación de forma individual, debemos establecer la información que proporcionan, en conjunto, la totalidad de las observaciones de una clase. La forma general de la expresión de una densidad de probabilidad a partir de un conjunto de funciones núcleo es:

$$P^*(Y,a) \equiv \frac{1}{N_a} \sum_{y_a \in a} K(y - y_a), \tag{3.13}$$

donde a es una muestra de la variable Y, N_a es el tamaño de la muestra y K es una función núcleo, centrada en y_a y que tal y como especificábamos anteriormente, alcanza un máximo en él, decreciendo monótonamente conforme se incrementa la distancia.

Antes de describir en detalle las funciones núcleo usadas habitualmente, introducimos la forma general de dichas funciones [Devijver y Kittler, 1982]:

$$K(y, y_i) = \frac{1}{\rho^d} h \left[\frac{\delta(y, y_i)}{\rho} \right], \tag{3.14}$$

donde: ρ es un parámetro del estimador, estrictamente positivo, que satisface:

$$\lim_{N_i \to \infty} \rho^d(N_i) = 0. \tag{3.15}$$

Esta condición sugiere que el ancho del núcleo depende, en última instancia, del número de muestras disponibles. Cuanto mayor sea el número de muestras de la densidad de probabilidad a estimar, menor será el ancho del núcleo.

$\delta(y, y_i)$ es una métrica definida sobre P, determinada por el tipo de núcleo que se vaya a emplear. $h[.]$ es una función que alcanza un máximo cuando $\delta(y, y_i) = 0$ y es monótona decreciente conforme $\delta(y, y_i)$ aumenta.

Si se exige que $h[.]$ sea no negativa, la única condición impuesta sobre ella es:

$$\int K(y, y_i) dy = 1 \tag{3.16}$$

En [Devijver y Kittler, 1982] se demuestra que las condiciones impuestas por las ecuaciones 3.15 y 3.16 garantizan que la expresión de la ecuación 3.13 es una función de densidad de probabilidad y proporciona una estimación consistente y no sesgada de $P(Y|a)$.

Varios tipos de funciones poseen las características anteriores: núcleo *gausiano, hiperesférico e hipercúbico*.

En nuestro planteamiento, se ha optado por el primer tipo, pues el cálculo de su derivada es sencillo, situando una gausiana en cada elemento de la muestra:

$$K(y, y_a) = \frac{1}{(2\pi)^{d/2}|\Psi|^{1/2}} \exp\left\{-\frac{1}{2}(y - y_a)^T \Psi^{-1}(y - y_a)\right\}, \tag{3.17}$$

donde el único dato a estimar es el ancho del núcleo ψ, caracterizado en este caso, por la matriz de covarianza. La calidad de la estimación depende precisamente de este valor. Como se verá en el apartado siguiente, valores muy pequeños realizarán una estimación de varianza elevada, mientras que valores muy grandes generarán una estimación demasiado sesgada, perdiendo detalles de la densidad.

Cuando las muestras están muy dispersas, el ancho del núcleo debería ser relativamente grande. Por el contrario, si están agrupadas el ancho del núcleo debería ser menor, para considerar tan solo la inmediata vecindad de las observaciones.

En su forma más general es posible tener en cada dimensión un ancho diferente y considerar núcleos en los que se contemple la correlación entre las variables. Por esta razón se toma la distancia de Mahalanobis entre la observación actual y el prototipo considerado.

Otra función que cumple con lo especificado en [Devijver y Kittler, 1982] es la función núcleo *hiperesférico*, definido de la siguiente forma:

$$K\ (Y, y_i)\ =\ \begin{cases} {v_\rho}^{-1} & \text{si } \{Y \mid \delta_E(Y, y_i)\ \leq\ \rho\} \\ 0 & \text{si } \{Y \mid \delta_E(Y, y_i)\ >\ \rho\} \end{cases} \tag{3.18}$$

donde $\delta_E(Y, y_i)$ es la distancia Euclídea entre Y y el prototipo y_i y v es el volumen de una hiperesfera d-dimensional de radio ρ que se obtiene a partir de la siguiente expresión:

$$v_\rho\ =\ \frac{\pi^{\frac{d}{2}}\,\rho^d}{\Gamma\left(\frac{d}{2}+1\right)}, \tag{3.19}$$

donde Γ es una función que se comporta de manera diferente dependiendo de si d es par o impar ya que su argumento sólo puede ser un entero. Es decir, si d es par, $\Gamma(n+1) = n!$ y si d es impar, entonces $\Gamma(n+1) = n\Gamma(n)$, con

d	$\frac{d}{2}+1$	$\Gamma(\frac{d}{2}+1)$	v_ρ
1	$\frac{1}{2}+1$	$\frac{1}{2}\Gamma(\frac{1}{2}) = \frac{1}{2}\sqrt{\pi}$	$\pi^{\frac{1}{2}}\rho/\frac{1}{2}\sqrt{\pi}$
2	$1+1$	$1! = 1$	$\pi\rho^2$
3	$\frac{3}{2}+1$	$\frac{3}{2}\Gamma(\frac{3}{2}) = \frac{3}{2}\Gamma(\frac{1}{2}+1) = \frac{3}{2}\frac{1}{2}\sqrt{\pi}$	$\pi^{\frac{3}{2}}\rho^3/\frac{3}{2}\frac{1}{2}\sqrt{\pi}$
4	$2+1$	$2! = 2$	$\pi^2\rho^4/2$

Tabla 3.1: Valores de los volúmenes de una hiperesfera para espacios de dimensiones comprendidas entre uno y cuatro.

$\Gamma(0{,}5) = \sqrt{\pi}$. En la tabla 3.1 mostramos las expresiones de los volúmenes de una hiperesfera de radio ρ para varias dimensiones.

Entre las características más relevantes de este núcleo, cabe destacar que proporciona un estimador constante por tramos de la función de densidad y resulta muy atractivo computacionalmente ya que los cálculos son relativamente sencillos y no demasiado costosos.

El último núcleo empleado para la estimación de densidad de probabilidad con ventanas de Parzen es el núcleo *Hipercúbico*. La forma funcional de una función núcleo hipercúbico es la siguiente:

$$K\left(Y, y_i\right) = \begin{cases} (2\rho)^{-d} & \text{si } \{Y \mid \delta_T(Y, y_i) \leq \rho\} \\ 0 & \text{si } \{Y \mid \delta_T(Y, y_i) > \rho\} \end{cases} \tag{3.20}$$

donde $\delta_T(Y, y_i)$ es la distancia de Chebyshev que se calcula como el valor absoluto de la máxima diferencia entre sus coordenadas individuales:

$$\delta_T(Y, y_i) = \max_{j=1..d} \left\{ |Y^j - y_i^j| \right\}. \tag{3.21}$$

La distancia de Chebyshev es una métrica muy atractiva computacionalmente frente a la distancia euclídea. Esta distancia recibe, en ocasiones, el nombre de distancia de Manhattan. En cualquier caso, podría emplearse la distancia euclídea si el problema es de baja dimensionalidad.

Al igual que el núcleo hiperesférico, proporciona un estimador constante por tramos de la función de densidad.

En la figura 3.2 mostramos la forma de los núcleos estudiados para espacios unidimensionales (A, B y C) y bidimensionales (D, E y F). En todos los casos, el núcleo está centrado en una observación y_i.

En el caso unidimensional, los núcleos hipercúbico (B) e hiperesférico (C) son muy similares, ya que el ancho del núcleo se reduce a un segmento de anchura, que es una función de ρ. La diferencia está en la aportación individual de cada observación. Con este tipo de núcleos, la estimación es constante en todo el ancho del núcleo (no hay un decrecimiento suave conforme nos alejamos de la observación sobre la que está centrado). La consecuencia es que la función de densidad estimada tiene forma escalonada, de ahí la afirmación de que era constante por tramos.

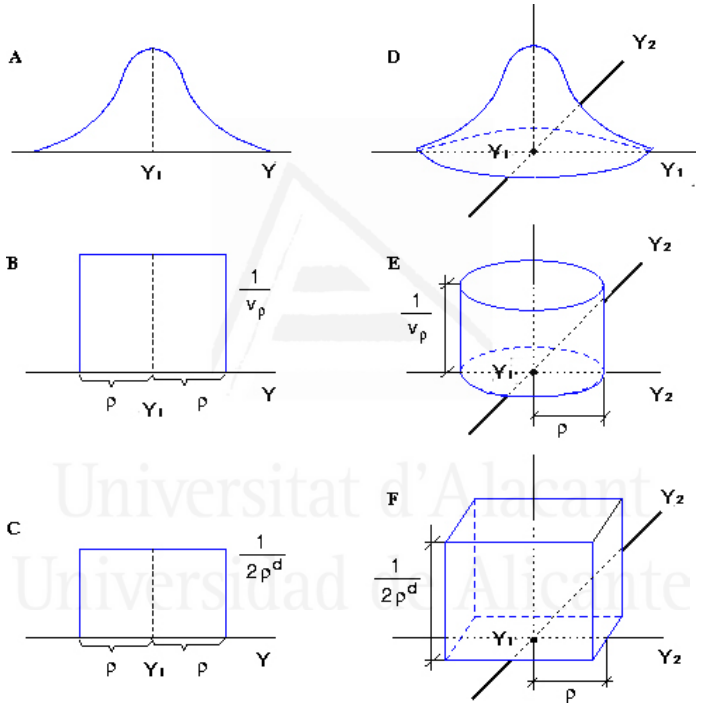

Figura 3.2: Forma de los núcleos para: (A, B y C) $d = 1$, (D, E y F) $d = 2$.

Para los núcleos bidimensionales el procedimiento de estimación es similar: el núcleo gaussiano (D) produce una estimación suave mientras que los núcleos hiperesférico (E) e hipercúbico (F) producen estimadores escalonados, solo que ahora la forma de los escalones es diferente. El núcleo hipercúbico da lugar a escalones rectangulares (resultantes de intersección de paralelepípedos) mientras que el núcleo hiperesférico da lugar a escalones de formas menos regulares (resultantes de la intersección de cilindros). En la

figura 3.3 mostramos la forma de la densidad de probabilidad estimada con núcleo Hipercúbico y Gausiano.

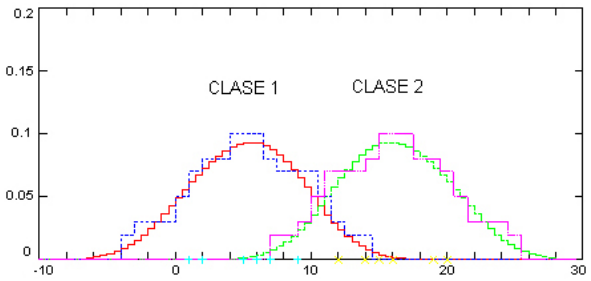

Figura 3.3: Estimación de las densidades de probabilidad en el rango $[-10, 30]$ con un núcleo hipercúbico de ancho 5 y con un núcleo gaussiano de ancho 3. El núcleo gausiano genera una estimación más fina de la densidad de probabilidad de las dos clases.

Puesto que el núcleo gausiano permite realizar un ajuste más preciso de la densidad de probabilidad asociada a los datos, se ha optado por este tipo de núcleo para la estimación de la entropía de la distribución.

Puesto que la integral de la ecuación 3.4 es difícil de calcular, podemos realizar una estimación de la entropía a partir de la media de los datos disponibles, según la siguiente expresión:

$$H^*(Y) = E_b[\log(P^*(Y, a))] = \frac{1}{N_b} \sum_{y_b \in b} \log(P^*(y_b, a)), \tag{3.22}$$

donde b es una segunda muestra de tamaño N_b de la misma variable. La media de los datos, así obtenida, converge hacia la media real de la distribución en un ratio de $1/\sqrt{N_b}$. Combinando las expresiones 3.13 y 3.22 obtenemos:

$$H^*(Y) = \frac{1}{N_b} \sum_{y_b \in b} \log(\frac{1}{N_a} \sum_{y_a \in a} K_\psi(y_b - y_a))), \tag{3.23}$$

por lo que podemos calcular la entropía de una distribución a partir de dos conjuntos de datos a y b. El único parámetro del modelo es el ancho de los núcleos empleados para el cálculo de las ventanas de Parzen. Esta definición de entropía ha sido empleada con anterioridad en [Viola, 1995] en el contexto de estimación de Información Mutua en imágenes y en

[Erdogmus *et al.*, 2004] para la estimación de la entropía de Rényi de orden α en el contexto de *Blind deconvolution of linear channels*. Este tipo de entropía será ampliamente estudiada en el apartado siguiente.

Por simplicidad, suponemos que los núcleos poseen una matriz de covarianza diagonal, siendo $\psi = \text{Diag}(\sigma_1^2, ...\sigma_{N_a}^2)$, con N_a el número de elementos de la muestra a. Esto implica la suposición de correlación nula y por lo tanto, una simplificación importante en el cálculo de la distancia, que se transforma en una distancia euclídea, mucho más simple computacionalmente:

$$K(y, y_a) = \frac{1}{\prod_{i=1}^{d} \sigma_i (2\pi)^{d/2}} \prod_{j=1}^{d} \exp\left\{ -\frac{1}{2} \left(\frac{y^j - y_a^j}{\sigma_j} \right)^2 \right\}, \tag{3.24}$$

donde y^j representa la j-ésima componente del dato y y y_a^j representa la j-ésima componente del núcleo y_a. En [Viola *et al.*, 1996] se propone un método de ajuste del ancho óptimo basado en máxima verosimilitud. A partir de la definición de entropía de la ecuación 3.3, obtenemos:

$$H_b(Y) \equiv -E_b[\log(P(Y))] = -\frac{1}{N_b} \sum_{y_b \in b} \log(P(y_b)) = -\frac{1}{N_b} \log(\ell(b)), \tag{3.25}$$

donde $\ell(b)$ es la verosimilitud de los datos. Por tanto, maximizar la verosimilitud es equivalente a minimizar la entropía obtenida a partir de los datos. La técnica consiste en calcular la derivada de la entropía respecto del ancho de los núcleos y realizar un descenso por gradiente que permita obtener el ancho óptimo:

$$\frac{\partial}{\partial \sigma_d} H^*(Y) = \frac{1}{N_b} \sum_{y_b \in b} \sum_{y_a \in a} \frac{K_\psi(y_b - y_a)}{\sum_{y_a \in a} K_\psi(y_b - y_a)} \left(\frac{1}{\sigma_d} \right) \left(\frac{[y_b - y_a]_d^2}{\sigma_d^2} - 1 \right), \tag{3.26}$$

siendo σ_d la desviación típica en cada dimensión. En la figura 3.4 se muestra el algoritmo que ajusta el ancho óptimo de los núcleos de Parzen. Sobre el conjunto inicial de muestras disponibles se crean dos subconjuntos con N_a y N_b elementos respectivamente. El primero se emplea para determinar la densidad de probabilidad y el segundo para estimar la entropía. Para garantizar que el algoritmo no se detenga en un mínimo local, se ha realizado una modificación de la propuesta original en [Viola *et al.*, 1996] consistente en emplear

un parámetro λ adaptativo. De esta forma, en posiciones alejadas del óptimo el parámetro es elevado para que la convergencia hacia el óptimo sea rápida. Cuando nos acercamos al óptimo las variaciones deben ser menores por lo que el parámetro se reduce en una cantidad ϵ y se itera de nuevo. El proceso se repite mientras λ sea superior a un valor mínimo λ_{min}. La única condición que debe cumplir ϵ es que su valor ha de estar comprendido entre 0 y 1.

AJUSTE VARIANZA NÚCLEOS PARZEN
Entrada: Valores iniciales ancho núcleos: σ_{inic}, λ_{inic}, λ_{min}
Salida: Ancho núcleos finales tras descenso
Seleccionar N_a muestras para Parzen
Seleccionar N_b muestras para el cálculo de la entropía
$\lambda = \lambda_{inic}$
$\sigma_d = \sigma_{inic}$
$H_0(Y) = \infty$
$t \leftarrow 1$
do
$\sigma_{new} = \sigma_d + \lambda \frac{\partial}{\partial \sigma_d} H^*(Y)$
Estimar $H_t(Y)$ para el ancho actual
if $H_t(Y) < H_{t-1}(Y)$ **then**
$\sigma_d = \sigma_{new}$
$\lambda = \lambda_{inic}$
else
$\lambda = \lambda\epsilon$
endif
$t \leftarrow t + 1$
while $\lambda > \lambda_{min}$
falgoritmo

Figura 3.4: Algoritmo para ajustar el ancho de los núcleos de Parzen.

De este modo, el único parámetro necesario para estimar la entropía por este método es el ancho inicial de los núcleos de Parzen. La cuestión a resolver es cuál es el ancho del núcleo apropiado para un problema determinado. En [Devijver y Kittler, 1982] se propone un método para determinar el ancho del núcleo, que depende del número de observaciones empleadas para de-

terminar la densidad de probabilidad. Un valor es considerado adecuado si satisface la condición:

$$\lim_{N_a \to \infty} \sigma^d(N_a) = 0. \tag{3.27}$$

En particular, podría tomarse:

$$\sigma(N_a) = N_a^{\frac{-\eta}{d}}, \tag{3.28}$$

con $\eta \in (0, 1)$. Aunque podría pensarse que el tipo de núcleo a adoptar es un factor determinante en la calidad de la estimación, diversos autores (véase [Spiegelhalter y Taylor, 1994], por ejemplo) justifican que el valor del ancho del núcleo es mucho más importante.

En la figura 3.5 representamos la estimación de la entropía obtenida para una muestra de una variable gausiana de dimensión 2, con matriz de covarianza $\psi = \text{Diag}(0{,}36, 0{,}09)$ para diferentes valores del ancho de los núcleos. Puesto que la variable es gausiana, el valor de la entropía viene dado por la ecuación 3.12, que para la distribución del ejemplo tiene un valor de $1{,}12307$.

De la forma de la función de la figura 3.5 se deduce que el intervalo de valores de anchura de núcleos que genera una estimación adecuada de la entropía es suficientemente amplio, por lo que la dependencia del ancho de núcleo elegido no es crítica. Por otra parte, a medida que el ancho de los núcleos tiende a cero en alguna de sus dimensiones, la densidad de los datos que no están en la muestra a tiende a cero y por consiguiente, la entropía tiende a $+\infty$ y la función deja de ser suave.

3.3.2. Método basado en MST (Minimal Spanning Trees)

Uno de los principales problemas del método anterior para la estimación de la entropía es que no es adecuado para problemas de dimensionalidad elevada. Debido a la necesidad de estimar previamente la densidad de probabilidad asociada a los datos, el número de observaciones necesarias para poder realizar una estimación precisa cuando el número de dimensiones es alto crecería exponencialmente, quedando algunas zonas del espacio sin prácticamente ningún representante y por tanto, no se obtendría una buena estimación. Este problema es conocido como *la maldición de la dimensionalidad*.

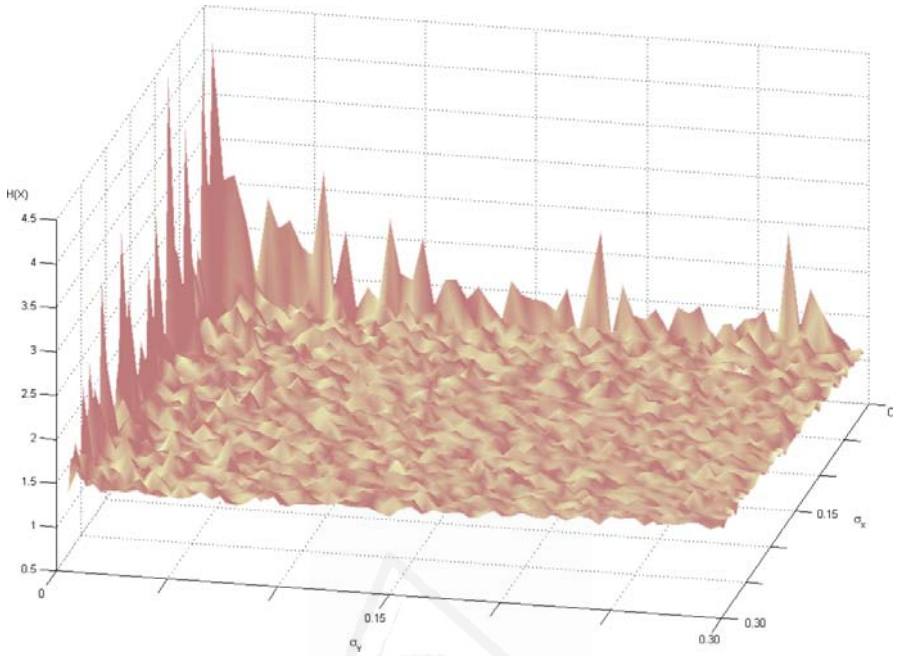

Figura 3.5: Representación de la estimación de la entropía en función del ancho de los núcleos de Parzen para dos dimensiones.

Una posible solución al problema es emplear métodos que realicen directamente una estimación de la entropía sin realizar una estimación previa de la densidad de probabilidad asociada a los datos.

Los *Entropic Spanning Graphs* unen un conjunto de vectores de características de tal forma que la longitud normalizada del grafo converge hacia la entropía de la distribución de probabilidad asociada a los datos a medida que se incrementa el número de vectores.

Este tipo de grafos [Hero y Michel, 2002], permiten realizar una estimación de la entropía de Rényi de orden α [Renyi, 1961] y pertenecen a los métodos de estimación denominados *non plug-in*. La entropía de Rényi de orden α de una función de densidad de probabilidad f es una generalización de la entropía de Shannon y se define como:

$$H_\alpha(f) = \frac{1}{1-\alpha} \ln \int_z f^\alpha(z) dz \tag{3.29}$$

para $\alpha \in (0, 1)$. La entropía de orden α converge a la entropía de Shannon $-\int f(z) \ln f(z) dz$ a medida que $\alpha \to 1$, por eso es posible obtener la segunda

a partir de la primera, aunque como veremos más adelante, la obtención no es directa.

Un grafo G se define por un conjunto de vértices $X_n = \{x_1, ..., x_n\}$, con $x_n \in R^d$ y aristas $\{e\}$ que conectan dos a dos los vértices del grafo: $e_{ij} = (x_i, x_j)$. Si denominamos $M(X_n)$ al conjunto de posibles aristas en la clase de los grafos acíclicos que expanden X_n, podemos definir el árbol de expansión mínima *Minimal Spanning Tree* (en adelante MST) en función de la distancia euclídea ponderada como:

$$L_\gamma^{MST}(X_n) = \min_{M(X_n)} \sum_{e \in M(X_n)} \mid e \mid^\gamma \tag{3.30}$$

con $\gamma \in (0, d)$ y $\mid e \mid$ la distancia euclídea entre los vértices del grafo.

El MST ha sido empleado con éxito para medir el grado de aleatoriedad de un conjunto de puntos. En la figura 3.6 mostramos el MST para dos conjuntos de puntos generados aleatoriamente en un espacio bidimensional. En la figura de la izquierda se muestra el resultado obtenido para una distribución uniforme, mientras que en el de la derecha se muestra el resultado obtenido para una distribución gausiana cuyos datos están más concentrados.

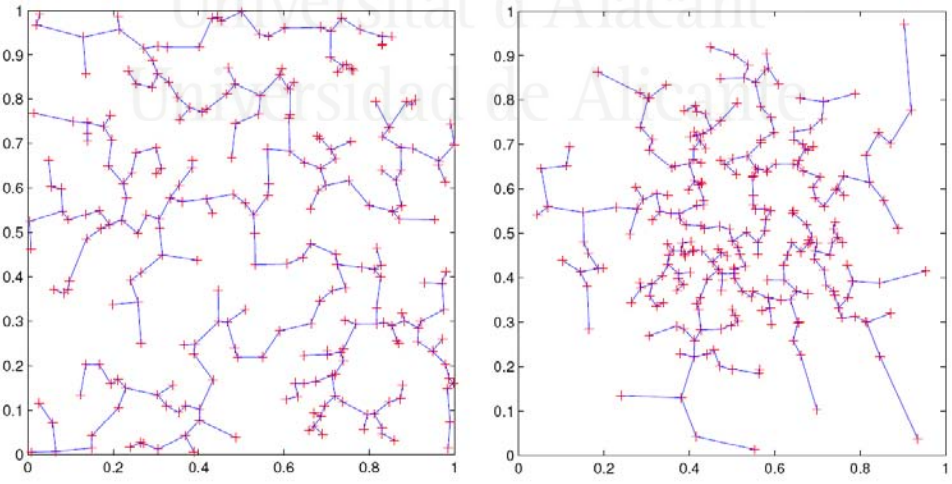

Figura 3.6: Comparación de la longitud del MST entre dos distribuciones de 256 puntos. Izquierda: Distribución uniforme. La longitud del MST es 10,5207; Derecha: Distribución gausiana. La longitud del MST es 8,4516.

La figura 3.7 muestra la longitud del MST como función del número de observaciones de la distribución. Es intuitivo pensar que la longitud del MST para el caso de la distribución no uniforme (más concentrada) se incrementa en menor medida en el caso de la distribución uniforme (más dispersa). Este hecho es el principal motivo de emplear el MST como forma de estimar el grado de aleatoriedad de un conjunto de puntos [Hoffman y Jain, 1983]. Incluso si se normaliza el MST con $\sqrt{n}$ y tomamos el logaritmo de esas funciones de longitud, se genera una secuencia que converge (con un factor constante) hacia la entropía de Rényi de orden $\alpha = 1/2$. Por último, si se cambia el valor de γ en la expresión 3.30, se puede lograr una secuencia convergente hacia valores de $\alpha = (d-\gamma)/d$, con $\gamma \in (0, d)$.

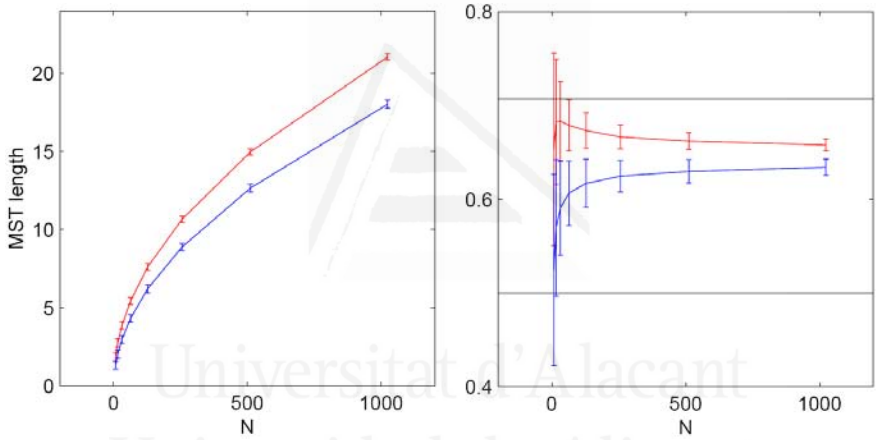

Figura 3.7: Izquierda: longitud del MST para la distribución uniforme (Rojo) y para la distribución gausiana (Azul). Derecha: MST dividido entre $\sqrt{n}$.

En [Hero y Michel, 1999a] se demuestra que en un espacio de características d-dimensional, con $d \geq 2$:

$$H_\alpha(X_n) = \frac{d}{\gamma}\left[\ln \frac{L_\gamma(X_n)}{n^\alpha} - \ln \beta_{L_\gamma,d}\right] \tag{3.31}$$

es un estimador asintótico no sesgado y casi absolutamente consistente de la entropía de orden α de f, con $\alpha = (d-\gamma)/d$ y $\beta_{L_\gamma,d}$ una constante correctora del sesgo, dependiente del criterio empleado para minimizar el grafo, pero independiente de la función densidad de probabilidad asociada a los datos f.

No existen expresiones cerradas para calcular $\ln(\beta_{L_\gamma,d})$, sólo aproximaciones y cotas:

- Simulación por métodos de Monte Carlo para un conjunto uniforme de observaciones aleatorias en un cubo de tamaño unidad $[0,1]^d$.
- Aproximación para valores altos de la dimensión:

$$\ln(\beta_{L_\gamma,d}) = \left(\frac{\gamma}{2}\right) \ln\left(\frac{d}{2\pi e}\right) \tag{3.32}$$

  propuesto en [Bertsimas y Ryzin, 1990].

A partir de la expresión 3.31, podemos estimar $H_\alpha(f)$ para valores diferentes de $\alpha = (d-\gamma)/d$ cambiando el exponente del peso de las aristas γ. Como γ modifica los pesos de las aristas de forma monótona, el grafo es el mismo para diferentes valores de γ y por tanto, sólo la longitud total de la expresión 3.31 debe ser recalculada.

3.3.3. Estimación de la entropía de Shannon a partir de la entropía de Rényi

Los *Entropic Spanning Graphs* son adecuados para la estimación de la entropía de orden α, con $\alpha \in [0,1[$, por eso la entropía de Shannon no puede ser directamente estimada con este método. En [Zyczkowski, 2003] se discute la relación existente entre la entropía de Shannon y la de Rényi de orden entero. Para cualquier distribución de probabilidad discreta, obtenida a partir de N puntos u observaciones de la misma, para la que se conozcan las entropías de Rényi de ordenes 2 y 3, se proporcionan una cota inferior y otra superior de la entropía de Shannon, aunque no se obtiene un valor concreto. En cualquier caso, dichas cotas no son de utilidad para nuestro propósito, pues la técnica de los MST sólo permite la estimación de entropías de $\alpha \in (0,1)$. En [Mokkadem, 1989], el autor propone la construcción de una estimación no paramétrica de la entropía de Shannon a partir de una secuencia convergente de estimaciones de entropías de orden α.

Por la relación existente entre γ y α, a medida que $\gamma \to 0$, $\alpha \to 1$, es decir, tiende a la entropía de Shannon. Sin embargo, la expresión 3.31 no permite asociar $\alpha = 1$, pues la primera parte de la expresión generaría una indeterminación (división por cero).

3.3.4. Método propuesto para la estimación de la entropía de Shannon

En esta sección proponemos una técnica para estimar la entropía de Shannon a partir de la estimación de la entropía de Rényi de orden α obtenida por el método expuesto anteriormente. Para ello, trataremos de averiguar el comportamiento de la función en el límite, es decir, cuando $\gamma \to 0+$ y por tanto, $\alpha \to 1-$.

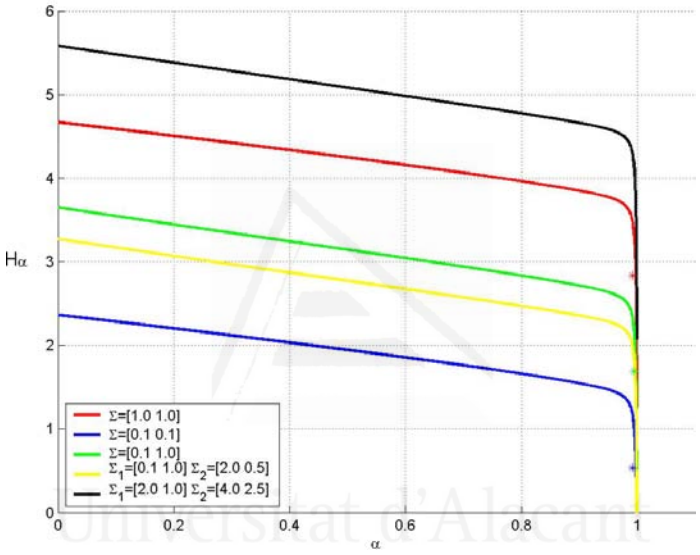

Figura 3.8: Representación de la función entropía de Rényi para diferentes valores de $\alpha \in [0, 1[$. En color negro y amarillo se muestran dos distribuciones bimodales. El resto de colores representan distribuciones gausianas con diferente matriz de covarianza.

Si representamos gráficamente la entropía de Rényi en función de α (figura 3.8), estimada mediante la ecuación 3.31, podemos observar que presenta una asíntota vertical para $\alpha = 1$. A medida que α tiende a 1, el valor de la función tiende a $-\infty$.

De la misma figura, se desprende que la forma de la función no depende de la naturaleza de los datos (gausianos o bimodales) ni del número de éstos empleados para la estimación de la entropía por el método del MST. A medida que el valor de α se aproxima a 1, la estimación de la entropía se hace cada vez más negativa, sobrepasando el valor teórico de la entropía de Shannon.

Por ello, es necesario encontrar un método que permita estimar cual sería el valor de la función para $\alpha = 1$.

Dado que no es posible calcular directamente el valor de H_α para $\alpha = 1$, aproximaremos dicho valor mediante una función continua que capture la tendencia de H_α en las inmediaciones de 1. A partir de un valor de $\alpha \in [0, 1[$, podemos calcular la recta $y = mx+b$ tangente a H_α en dicho punto, utilizando $m = H'_\alpha$, $x = \alpha$ y $y = H_\alpha$. H'_α representa la derivada de la entropía de orden α. Dada la complejidad de la expresión que permite el cálculo de la entropía a partir del MST, se ha optado por realizar la derivada numérica por el método clásico de los dos puntos [Faires y Burden, 2004]. En cualquier caso, la recta así calculada será continua y podremos calcular su valor para $x = 1$ (figura 3.9).

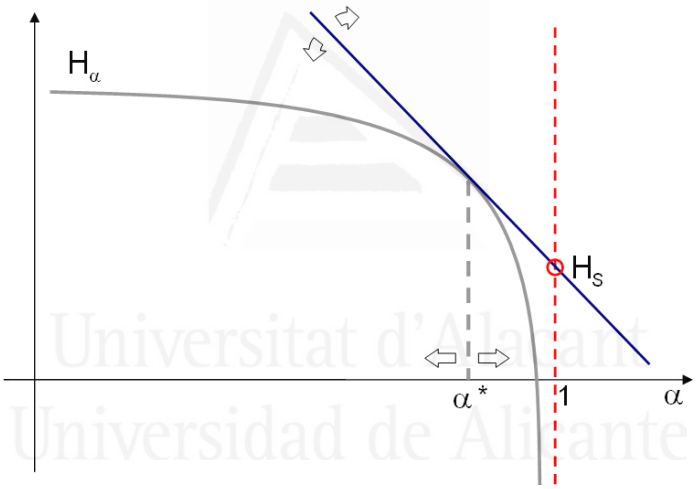

Figura 3.9: Representación de H_α y la recta tangente con la que trataremos de aprender el valor de α^* que nos permita averiguar el valor de la entropía para $\alpha = 1$.

El punto de corte de la recta generada en el valor 1 será distinto dependiendo del valor de α utilizado. En adelante, llamaremos α^* al valor de α tal que, siguiendo el procedimiento descrito, genera el valor correcto de la entropía en el valor 1. Por tanto, si conociéramos el valor de α^*, podríamos calcular el valor de b, como:

$$b = y - mx = H_{\alpha^*} - H'_{\alpha^*}\alpha^* \tag{3.33}$$

A partir de este valor, podríamos realizar la estimación de la entropía de

Shannon $H_s = H_1$ mediante la recta y con los parámetros definidos anteriormente, tomando $x = 1$ de la siguiente forma:

$$H_s = m + b = H'_{\alpha^*} + H_{\alpha^*} - H'_{\alpha^*}\alpha^* \tag{3.34}$$

Experimentalmente hemos comprobado que el valor de α^* no depende de la naturaleza de la distribución de probabilidad asociada a los datos, sino del número de observaciones empleadas para la estimación y de la dimensionalidad del problema.

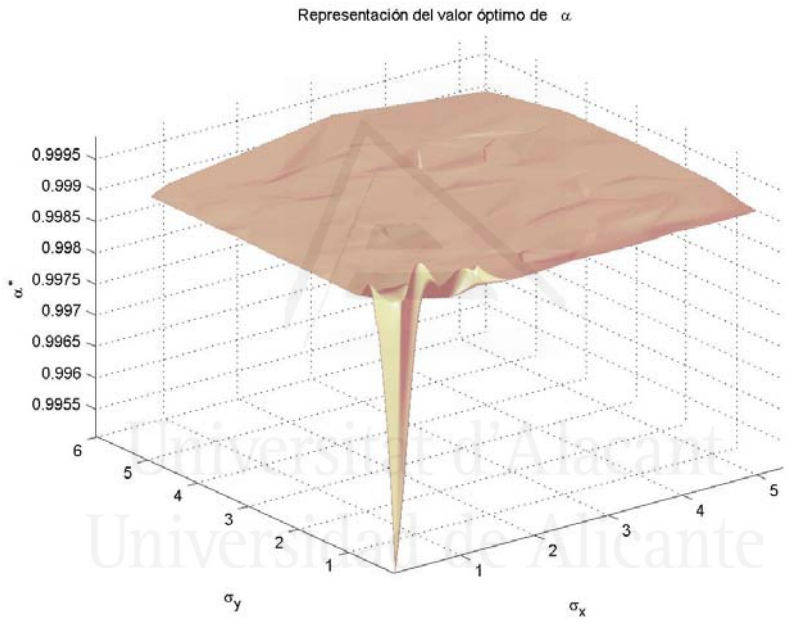

Figura 3.10: Representación de α óptimo 2D para diferentes valores de varianza y 400 observaciones.

Puesto que H_α es una función monótona decreciente y conocemos el valor de H_α en el caso gaussiano (estimada directamente mediante 3.12), podemos estimar el valor de α^* para distribuciones gausianas mediante una búsqueda dicotómica entre valores bien separados de α para un número constante de observaciones, dimensión del problema y diferentes matrices de covarianza. Experimentalmente hemos verificado que para una dimensionalidad y número de observaciones dado, α^* es casi constante para distribuciones de

probabilidad con matriz de covarianza diagonal y valores de varianza mayores de $0{,}5$. La figura 3.10 muestra la estimación de α^* para funciones de densidad de probabilidad en dos dimensiones con varianzas entre $0{,}1$ y $5{,}0$ y 400 observaciones de cada una de ellas, en la que se puede comprobar la afirmación anterior. Este hecho permite realizar una estimación precisa sin una sensibilidad excesiva al valor de α^* escogido.

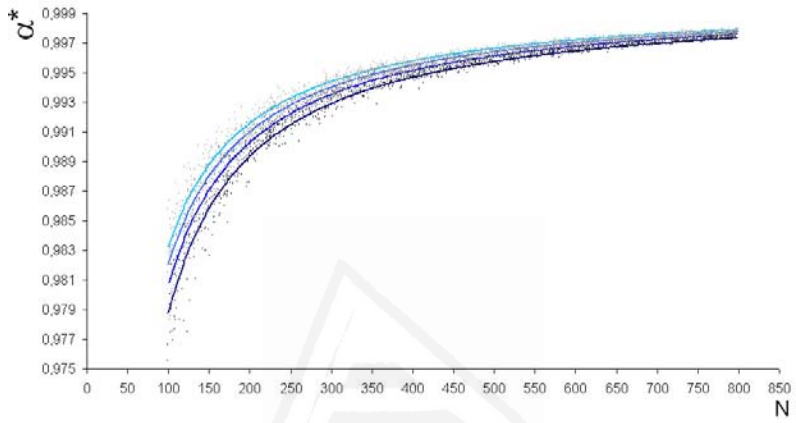

Figura 3.11: Representación gráfica del valor de α^* en función del número de observaciones disponibles para dimensiones entre 2 y 5. La curvas presentan la misma forma con independencia de la dimensión, pero se observa una convergencia acusada cuando el número de observaciones es elevado.

Por tanto, manteniendo constante el número de observaciones y la dimensionalidad, el α óptimo se puede aproximar por una constante. El problema ahora es generalizar el método para contemplar distintas dimensionalidades y número de observaciones. Para apreciar los efectos de estas dos variables del problema, generamos un experimento en el que, manteniendo el caso gausiano, calculamos el α óptimo para un conjunto de 1000 distribuciones, variando aleatoriamente la dimensionalidad del problema entre 2 y 5, el número de observaciones entre 50 y 1000 y las varianzas entre $0{,}5$ y 10. Experimentalmente hemos verificado que la forma de la curva resultante se ajusta adecuadamente a la siguiente expresión:

$$\alpha^* = 1 - \frac{a + b \exp^{cD}}{N}, \tag{3.35}$$

donde N es el número de observaciones, D es la dimensión del problema

a	b	c
1,271	1,3912	−0,2488

Tabla 3.2: Valores de las constantes a, b, y c obtenidas experimentalmente mediante simulación por métodos de Monte Carlo.

y a, b, c son tres constantes a estimar. Para estimar esos valores, llevamos a cabo una simulación por métodos de Monte Carlo, de modo que se minimice el error cuadrático medio entre la expresión y los datos. Tras el proceso se obtienen valores para el conjunto de parámetros a estimar (tabla 3.2).

La figura 3.11 muestra la forma de la función para diferentes dimensiones y número de observaciones. Se puede comprobar que el valor óptimo de α presenta una clara tendencia en cada dimensión. De este modo, es posible averiguar el valor de α^* a partir de la dimensionalidad del problema y del número de observaciones disponibles.

3.3.5. Comprobación de la calidad de la estimación

Para comprobar la calidad de la estimación realizada por el método descrito anteriormente se han realizado múltiples experimentos para diferente número de observaciones, dimensionalidad y matrices de covarianza. En la figura 3.12 se muestra el resultado de uno de los experimentos consistente en la generación aleatoria de 1000 distribuciones gausianas en 4-D con un número de observaciones entre 100 y 800 y matriz de covarianza diagonal con varianzas entre 0,5 y 10,5 en cada dimensión. Todos los parámetros de las distribuciones así generadas se obtienen también de forma aleatoria. Puesto que las distribuciones son gausianas, podemos aplicar la fórmula de la ecuación 3.12 y comparar el valor obtenido con la aplicación del método y el valor real obtenido a partir de la fórmula. Para la representación de la gráfica se ha seleccionado el tramo de entropías con valores comprendidos entre 7 y 10,5. Los resultados reflejan que la aproximación obtenida con nuestra técnica coincide con una precisión muy elevada con el valor real obtenido a partir de la fórmula.

Para comprobar la calidad del ajuste para distribuciones no gausianas se han realizado igualmente múltiples experimentos en los que se han genera-

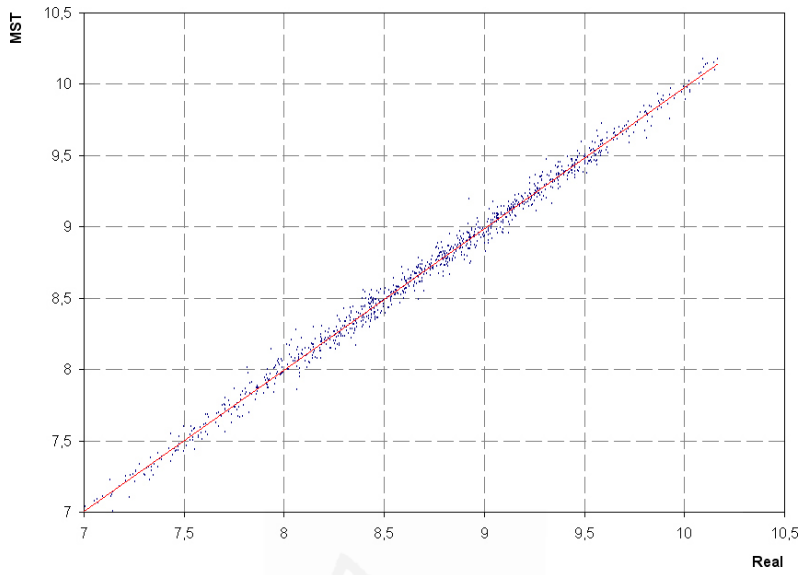

Figura 3.12: Comparativa entre la estimación de la entropía por el método del MST y el valor real obtenido a partir de la fórmula de la ecuación 3.12 en el caso de distribuciones gausianas. En el eje X se muestra el valor de la entropía real, mientras que el eje Y muestra el valor estimado de entropía por el método MST. La línea de color rojo representa la recta de regresión ajustada por mínimos cuadrados entre ambas variables. La relación existente es prácticamente lineal, según la recta $Y = 0{,}99X + 0{,}0727$.

do distribuciones bi-modales para un número de observaciones y varianzas también aleatorios. Puesto que en este caso no se dispone de formulación para calcular directamente el valor de la entropía real, se han comparado los resultados obtenidos por este método con por el método *plug-in* de las ventanas de Parzen. Para el experimento se han generado 1000 distribuciones de entre 100 y 1000 observaciones cada una. Las varianzas para cada una de las modas de las distribuciones oscilan entre 5 y 25.

La figura 3.13 muestra una comparación de las estimaciones obtenidas por ambos métodos para distribuciones de probabilidad de diferente naturaleza y número de observaciones. Siguiendo la misma idea que en la imagen anterior en el eje X se muestra la entropía estimada por el método MST, mientras que en el eje Y se representa la estimación obtenida mediante las ventanas de Parzen. En esta ocasión, la recta de regresión se encuentra desplazada por encima de la diagonal principal, indicando que el valor estimado por el méto-

do de las ventanas de Parzen está ligeramente por encima del MST. Además, en las pruebas realizadas observamos que para un número de datos reducido, el método del MST ofrece valores de entropía más bajos que el método de las ventanas de Parzen, obteniendo valores muy similares cuando el número de datos es superior a 800. Este hecho es explicable dado que la estimación por Parzen se realiza mediante un descenso por gradiente para calcular el ancho óptimo del núcleo. En [Viola, 1995] pág. 64 el autor apunta que la estimación obtenida por este método podría ser una cota superior de la entropía real.

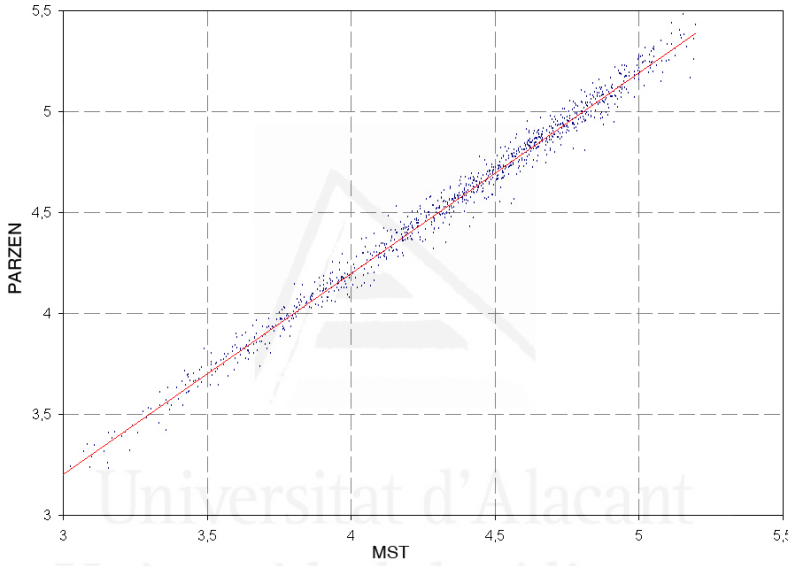

Figura 3.13: Comparativa entre los valores de entropía para diferentes distribuciones de probabilidad y número de observaciones. En el eje X se muestra la estimación por MST y en el eje Y la estimación por el método de las ventanas de Parzen. La recta de regresión lineal es $Y = 0{,}9955X + 0{,}2168$ reflejando claramente que Parzen genera estimaciones ligeramente superiores al MST.

Durante las pruebas realizadas para comprobar el comportamiento de las dos técnicas de estimación de entropía hemos verificado que el ajuste por la técnica MST es más preciso cuando el número de observaciones es reducido. Este hecho es debido a que no es necesario descomponer el conjunto de observaciones en dos para estimar previamente la densidad de probabilidad asociada al conjunto de datos. Relacionado con esto último y con la maldición de la dimensionalidad, también se obtienen mejores resultados que la técni-

ca *plug-in* cuando el número de dimensiones del problema es elevado. Por el contrario, la técnica basada en MST es mucho más sensible a la presencia de falsos positivos que la técnica basada en Ventanas de Parzen, por lo que no es adecuada para problemas en los que el grado de gausianidad de los datos es bajo, como los experimentos de segmentación de imágenes en color del capítulo siguiente. Este inconveniente podría ser resuelto empleando alguna de las técnicas que permiten una construcción robusta del árbol, como la propuesta en [Banks *et al.*, 1992] o la variación del algoritmo MST, denominada K-MST, propuesta en [Hero y Michel, 1998] que permite eliminar los falsos positivos durante el proceso de construcción del MST mediante una técnica voraz.

3.4. Algoritmo EM basado en máxima entropía

Si comparamos las estimaciones obtenidas a partir de las ecuaciones 3.12 con 3.23 y 3.31, tenemos una forma de cuantificar el grado de gausianidad de un núcleo determinado, lo que permitirá determinar cual de ellos ajusta en peor medida los datos de su vecindad. A partir de un conjunto de núcleos de la mezcla (inicialmente sólo uno) podemos evaluar la entropía global real $H(Y)$ y la entropía máxima teórica $H_{max}(Y)$ de la mezcla completa considerando los pares de entropías individuales de cada núcleo y sus correspondientes probabilidades a priori. De este modo conseguimos el primero de los criterios de parada del algoritmo propuestos en el presente trabajo.

3.4.1. Grado de gausianidad de la muestra completa

Esta medida está basada en el hecho de que para cualquier núcleo podemos obtener de forma directa la entropía máxima teórica obtenida, en el caso de que los datos de su vecindad fueran verdaderamente gausianos y la entropía real obtenida mediante alguno de los métodos de estimación anteriormente descritos:

$$H(Y) = \sum_{k=1}^{K} \pi_k H_k(Y). \tag{3.36}$$

$$H_{max}(Y) = \sum_{k=1}^{K} \pi_k H_{max}(k). \tag{3.37}$$

Una vez obtenidos estos valores podemos definir el *grado de gausianidad* G de la mezcla completa comparando ambas medidas. Son varios los criterios que pueden ser empleados en la comparación. En el siguiente capítulo se presentarán experimentos en los que se empleará la medida como criterio de parada del algoritmo:

$$G = H_{real}(Y)/H_{max}(Y) \tag{3.38}$$

$$\begin{aligned} G = H_{max}(Y) - H_{real}(Y) &= \sum_{k=1}^{K} \pi_k \left(\frac{H_{max}(k) - H_{real}(k)}{H_{max}(k)} \right) \\ &= \sum_{k=1}^{K} \pi_k \left(1 - \frac{H_{real}(k)}{H_{max}(k)} \right) \end{aligned} \tag{3.39}$$

Las diferentes expresiones empleadas como criterio de gausianidad de las ecuaciones 3.38 y 3.39 están acotadas y disponen de un valor óptimo que se debe tomar como valor máximo de gausianidad. En el primer caso $0 \leq G \leq 1$ y sólo se alcanzaría el hipotético valor 1 en el caso de que los datos fueran verdaderamente normales y no existiera ningún tipo de ruido. En el segundo caso, del mismo modo, $0 \leq G \leq 1$ y de forma análoga sólo se alcanzaría el valor 0 cuando los datos fueran completamente gausianos. El cociente entre H_{max} se lleva a cabo para que la medida sea adimensional y no dependa del rango de valores de entropía de los núcleos.

Si dicho valor se aproxima al valor óptimo teórico con una diferencia que no supere un valor umbral, consideramos que todos los núcleos están correctamente ajustados. Si por el contrario el valor queda por encima, existe alguna zona del espacio de datos que no está correctamente modelada y por tanto debemos introducir una nuevo componente al modelo en la zona peor ajustada.

Para ello, seleccionamos el núcleo con peor ratio individual y los sustituimos por dos nuevos núcleos que deben ser correctamente situados e inicializados. A continuación se lanza una nueva iteración del algoritmo EM con $K + 1$ núcleos.

Un valor bajo de G en el primer caso o alto en el segundo implica que existe multi-modalidad en alguna de las zonas del espacio de datos y por tanto el núcleo con peor valor individual de la medida debe ser reemplazado por otros dos que capten con más precisión la densidad de probabilidad en esa zona del espacio de datos.

3.4.2. Criterio de parada basado en MDL y MML

Como se ha comentado anteriormente, el criterio de máxima verosimilitud utilizado para la convergencia del algoritmo EM no puede ser empleado para la estimación del número óptimo de núcleos del modelo. A lo largo de la literatura se han presentado numerosos métodos para determinar lo que se conoce como *orden del modelo*. Entre ellos podríamos citar el *Bayesian Inference Criterion* (BIC) [Schwarz, 1978], *Akaike's Information Criterion* (AIC) [Akaike, 1973], el criterio MDL (*Minimum Description Length*) [Rissanen, 1983] o el criterio MML (*Minimum Message Length*) [Wallace y Freeman, 1987].

En el caso de MDL y sus variantes o MML (que serán posteriormente empleados como criterios de parada del algoritmo propuesto), la idea es seleccionar un modelo para la representación de los datos empleando un mensaje de la menor longitud posible, o lo que es lo mismo, con el menor número de parámetros, de entre un conjunto inicial de modelos. En teoría, el programa de ordenador más corto que genere un conjunto de datos y, proporcionaría la descripción más eficiente de tales datos. No obstante, en uno de sus teoremas relativos a complejidad, Kolmogorov ([Cover y Thomas, 1991], cap. 7) afirma que no existe ningún algoritmo capaz de encontrar el programa de ordenador más corto para representar un conjunto de datos, por lo que cualquier intento de averiguar la longitud de dicho programa de forma absoluta sería infructuoso. Lo que sí es posible es minimizar la longitud de descripción de los datos, a partir de un conjunto de modelos candidatos $\mathcal{M}$.

Desde este punto de vista, el mensaje estaría compuesto por tres partes: el modelo m, el conjunto de parámetros Θ y el conjunto de datos y, codificados a partir del modelo y de los parámetros anteriores. Según esto, la longitud total del mensaje sería:

$$\mathcal{L}(y, \Theta, m) = \mathcal{L}(y|\Theta, m) + \mathcal{L}(\Theta|m) + \mathcal{L}(m). \tag{3.40}$$

Si el número de modelos candidatos $|\mathcal{M}|$ es finito, el tercer sumando de la expresión sería constante, por lo que dicho término suele ser eliminado de la expresión anterior. De este modo, la longitud de código total estaría compuesta por una codificación en dos partes: primero codificamos los datos y dado el conjunto de parámetros Θ y después codificamos Θ.

Tanto el principio de longitud de descripción mínima como el de longitud mínima de mensaje para la selección del orden del modelo y estimación de parámetros, eligen valores de Θ y m que permiten minimizar la expresión de la ecuación 3.40 para el conjunto de datos y. No obstante, es necesario convertir la expresión anterior en una fórmula directamente utilizable en un problema de estimación en particular [Rissanen, 1989]. Según la teoría de Shannon, la longitud de código óptima para $\mathcal{L}(y|\Theta, m)$ coincide con el logaritmo de la verosimilitud del conjunto de datos dado el conjunto de parámetros multiplicado por -1.

$$\mathcal{L}(y|\Theta) = -L(\Theta, y) \tag{3.41}$$

Eliminando m de la expresión 3.40 y realizando la sustitución de 3.41 obtenemos:

$$\mathcal{L}(y, \Theta) = -L(\Theta, y) + \mathcal{L}(\Theta) \tag{3.42}$$

$\mathcal{L}(\Theta)$ es el resultado de realizar el siguiente razonamiento: para obtener una codificación de longitud finita para Θ, sus elementos (con valores reales) deben ser truncados a una precisión finita. Si la precisión es reducida, la longitud del término también lo será, pero la codificación de los parámetros puede estar lejos de la óptima y la primera parte de la expresión adquirir más peso. Por el contrario, con una precisión mayor, los parámetros codificados podrían estar cerca de los óptimos, pero a costa de una longitud de código mayor.

Como se demuestra en [Rissanen, 1983], la longitud de código óptima para cada parámetro real si el conjunto de datos tiene un tamaño elevado es $1/2 \log n$. Según esto, podríamos definir un criterio para la selección del orden del modelo, a partir de una función de coste obtenida como el logaritmo de la verosimilitud más un término adicional, cuya función sea la de penalizar valores excesivamente elevados del número de núcleos, de la siguiente

forma:

$$\mathcal{C}_{MDL}(\Theta_{(k)}, k) = -L(\Theta_{(k)}, y) + \frac{N(k)}{2} \log n, \tag{3.43}$$

donde $N(k)$ representa el número de parámetros para especificar un modelo de mezcla con k componentes: priors, medias y matrices de covarianza de cada uno de los componentes de la mezcla. Si el modelo permite medias y covarianzas de los núcleos sin ningún tipo de restricción, entonces $N(k)$ se obtiene de la siguiente forma:

$$N(k) = (k-1) + k\left(d + \frac{d(d+1)}{2}\right) \tag{3.44}$$

donde $k-1$ representa el número de priors a estimar para una mezcla con k núcleos. Debido a la restricción $\sum_i \pi_i = 1$, en términos de codificación el último valor se puede obtener a partir de los $k-1$ anteriores. En el segundo sumando $d + d(d+1)/2$ representa el número de parámetros de un núcleo $N(1)$: para un espacio d-dimensional, la media requiere d parámetros, mientras que la matriz de covarianza, que por definición es simétrica, requiere $d(d+1)/2$ parámetros. Si tenemos k núcleos, entonces el número de parámetros de todos ellos será $kN(1)$, dando lugar a la expresión 3.44.

Según diversos autores, tanto MDL como BIC tienden a estimar un número de núcleos inferior al real en el caso de modelos de mezclas [Kontkanen *et al.*, 1996] [Oliver *et al.*, 1996] [Smyth, 1996]. En ambos criterios, todas las observaciones tienen la misma importancia en en la estimación del conjunto de parámetros del modelo. Éste no es el caso de los modelos de mezclas, en los que para la estimación del conjunto de parámetros de cada núcleo se tienen en cuenta las observaciones que fueron generadas por dicho núcleo. Este hecho es observable si se calcula la matriz de información de Fisher para un vector de parámetros en un modelo de mezcla m (ver [Titterington *et al.*, 1985] para más detalles):

$$I(\theta_m) = n\alpha_m I_1(\theta_m), \tag{3.45}$$

siendo $I_1(\theta_m)$ la información de Fisher asociada a una observación que ha sido generada por el núcleo m del modelo según la expresión:

$$I_1(\theta_m) = -E\left[\frac{\delta^2}{\delta\theta_m^2}\log P(y|\theta_m)\right]. \tag{3.46}$$

Lo que representa la ecuación 3.45 es que un parámetro θ_m muestra un tamaño proporcional a $n\alpha_m$ y no a n, lo cual es lógico puesto que, como se comentaba anteriormente, para la estimación de θ_m se emplean observaciones que fueron generadas por el núcleo m de la mezcla y el valor esperado para ese número de observaciones es precisamente $n\alpha_m$. Por ello, en [Figueiredo *et al.*, 1999] se propone un nuevo criterio en el contexto de los modelos de mezclas gausianas, denominado MMDL (*Mixture Minimum Description Length*), que introduce una penalización menor mediante la inclusión de un término negativo que tiene en cuenta la importancia de cada observación en el cálculo de los parámetros de cada componente de la mezcla:

$$\mathcal{C}_{MMDL}(\Theta_{(k)}, k) = -L(\Theta_{(k)}, y) + \frac{N(k)}{2}\log n + \frac{N(1)}{2}\sum_{i=1}^{k}\log\alpha_i, \tag{3.47}$$

donde $N(1)$ representa al número de parámetros reales que definen cada componente de la mezcla. Según la expresión de la ecuación 3.44, sustituyendo k por 1, tenemos: $N(1) = d + d(d+1)/2$. El término añadido a la ecuación 3.43 para obtener la ecuación 3.47 es negativo, por tanto introduce una penalización menor que el criterio MDL clásico para tratar de evitar una estimación del orden del modelo inferior al real.

Otro planteamiento diferente para obtener el valor mínimo de la expresión de la ecuación 3.40 denominado MML o *Minimum Message Length* es propuesto por Wallace y Freeman en [Wallace y Freeman, 1987] [Wallace y Freeman, 1992]. A diferencia del criterio MDL, en MML no se asume un total desconocimiento a priori de las distribuciones de probabilidad asociadas al conjunto de parámetros del modelo. La expresión para el criterio de selección del orden del modelo bajo este principio propuesta en [Lanterman, 2001] es:

$$\begin{aligned}\mathcal{C}_{MML}(\Theta_{(k)}, k) &= -L(\Theta_{(k)}, y) - \log P(\Theta_{(k)}) + \\ &+ \frac{1}{2}\log|\mathbf{I}(\Theta_{(k)})| + \frac{d}{2}(1 + \log k_d),\end{aligned} \tag{3.48}$$

d	k_d
1	0,083333
2	0,080188
3	0,078743
4	0,076603
5	0,075625
6	0,074244
7	0,073116
8	0,071682
12	0,070100
16	0,068299
24	0,065771

Tabla 3.3: Valores de la constante k_d para las mejores rejillas de cuantización conocidas en varias dimensiones.

donde $\Theta_{(k)}$ representa el conjunto actual de parámetros para una mezcla con k núcleos, d es la dimensión del conjunto de parámetros, $\mathbf{I}(\Theta_{(k)}) \equiv -E\left[D^2_{\Theta_{(k)}} \log P(Y|\Theta_{(k)})\right]$ representa la matriz de información de Fisher, con $|.|$ su determinante, $D^2_{\Theta_{(k)}}$ la matriz de segundas derivadas o *Hessiana* y k_d una constante relativa a la rejilla de discretización para diferentes dimensiones. En la tabla 3.3 mostramos los mejores valores para algunas dimensiones [Conway y Sloane, 1993]. Cuando el número de dimensiones crece, muestra un comportamiento asintótico y $k_d \to 1/2\pi e \approx 0{,}05855$.

La matriz de Fisher $\mathbf{I}(\Theta_{(k)})$ no puede ser obtenida de forma analítica para el caso de modelos de mezclas [Titterington *et al.*, 1985] [McLachlan y Basford, 1988] [Oliver *et al.*, 1996]. Para solventar este inconveniente, en [Figueiredo y Jain, 2002] se propone reemplazar la matriz anterior por la matriz de información de Fisher de los datos completos $\mathbf{I}_c(\Theta_{(k)})$, que proporciona una cota superior de la anterior [Titterington *et al.*, 1985] y posee estructura diagonal por bloques:

$$\mathbf{I}_c(\Theta_{(k)}) = \text{n bloques-diag}\{\pi_1\mathbf{I}^{(1)}(\Theta_1), ..., \pi_k\mathbf{I}^{(k)}(\Theta_k), \mathbf{M}\}, \tag{3.49}$$

donde $\mathbf{I}^{(1)}(\Theta_m)$ es la matriz de Fisher para una observación en particular

que ha sido generada por la componente m de la mezcla que tiene actualmente k núcleos y $\mathbf{M}$ es la matriz de Fisher para una distribución multinomial con $|\mathbf{M}| = (\pi_1\pi_2...\pi_k)^{-1}$ [Titterington *et al.*, 1985].

El término relativo a la probabilidad a priori del conjunto de parámetros $P(\Theta_{(k)})$ en la expresión 3.48 se modela asumiendo un total desconocimiento sobre su distribución, aunque se supone independencia entre los parámetros de cada núcleo y sus probabilidades a priori:

$$P(\Theta_{(k)}) = P(\pi_1, ..., \pi_k) \prod_{i=1}^{k} P(\Theta_i). \tag{3.50}$$

Para modelar cada uno de los factores de la expresión anterior se adoptan las probabilidades a priori no informativas de Jeffrey (*non-informative Jeffrey's prior*) [Bernardo y Smith, 1994], cuyas expresiones son:

$$P(\Theta_i) \propto \sqrt{|\mathbf{I}^{(1)}(\Theta_i)|} \tag{3.51}$$

$$P(\pi_1, ..., \pi_k) \propto \sqrt{|\mathbf{M}|} = (\pi_1\pi_2...\pi_k)^{\frac{-1}{2}} \tag{3.52}$$

con $0 \leq \pi_1, \pi_2, ..., \pi_k \leq 1$ y $\sum_{i=1}^{k} \pi_i = 1$. Para una mezcla con k componentes, la dimensión d del conjunto completo de parámetros $\Theta_{(k)}$ se obtiene como $d = N(1)k + k$, con $N(1)$ el número de parámetros que define cada componente. Sustituyendo las expresiones anteriores en 3.48 y tomando $k_d = 1/12$ se obtiene una expresión computable para el criterio basado en MML:

$$\begin{aligned} \mathcal{C}_{MML}(\Theta_{(k)}, k) &= \frac{N(1)}{2} \sum_{i=1}^{k} \log\left(\frac{n\pi_i}{12}\right) + \\ &+ \frac{k}{2}\log(\frac{n}{12}) + \frac{k(N(1)+1)}{2} - L(\Theta_{(k)}, y) \end{aligned} \tag{3.53}$$

Puesto que k_d no varía demasiado se establece el valor de $1/12 \simeq 0{,}0833$ correspondiente a una rejilla de cuantización a partir de regiones hipercúbicas [Lanterman, 2001].

Para cualquiera de los 3 criterios descritos con anterioridad (MDL, MMDL y MML), la idea es encontrar un modelo de mezcla con un número de núcleos k, que genere un valor mínimo para las funciones criterio asociadas a cada una de ellos (3.43, 3.47 o 3.53). En principio, si el termino de

penalización añadido a la función de verosimilitud es adecuado, el valor mínimo de la función debería coincidir con el número óptimo de núcleos del modelo. En la figura 3.14 representamos la evolución de dicha función para diferente número de núcleos en tres problemas de estimación de densidad de probabilidad asociado a un conjunto de datos. En el eje horizontal mostramos el número de núcleos del modelo mientras que el vertical representa la función de energía sin normalizar. En los tres criterios, la función presenta un comportamiento similar con un rápido descenso a medida que el número de núcleos del modelo se acerca al valor óptimo y un suave y progresivo ascenso una vez que el algoritmo sobrepasa el óptimo. Este comportamiento de la función de energía permite detener la ejecución del algoritmo justo en el momento en el que la inclusión de un nuevo núcleo supone un crecimiento en el valor de la función.

En el apartado correspondiente a la especificación del algoritmo se describirá con detalle su utilización como criterio de parada y las principales ventajas con respecto a propuestas anteriores que hacen uso de criterios de longitud mínima (de descripción o de mensaje) para seleccionar el orden del modelo.

3.4.3. Introducción de un nuevo núcleo

El problema de descomponer un núcleo en otros dos es equivalente al de introducir un nuevo núcleo al modelo, pues el resultado obtenido es el de pasar de una mezcla inicial de K componentes a otra con $K+1$ componentes. Este proceso requiere una inicialización cuidadosa de los parámetros de los dos nuevos elementos introducidos en la mezcla. La división de un núcleo se lleva a cabo cuando una sola distribución gausiana no es suficiente para captar con precisión la densidad de probabilidad de su vecindad, lo cual implica existencia de multi-modalidad, en oposición a gausianidad, como quedaba reflejado en la figura 2.2 del capítulo 2. Parece por tanto lógico que la posición inicial de los nuevos núcleos, o lo que es lo mismo, sus medias, se obtengan a partir de un desplazamiento en sentidos opuestos, sobre la dirección de máxima variabilidad de los datos captados en la vecindad del núcleo original.

En [Peñalver *et al.*, 2003], realizamos una primera aproximación al pro-

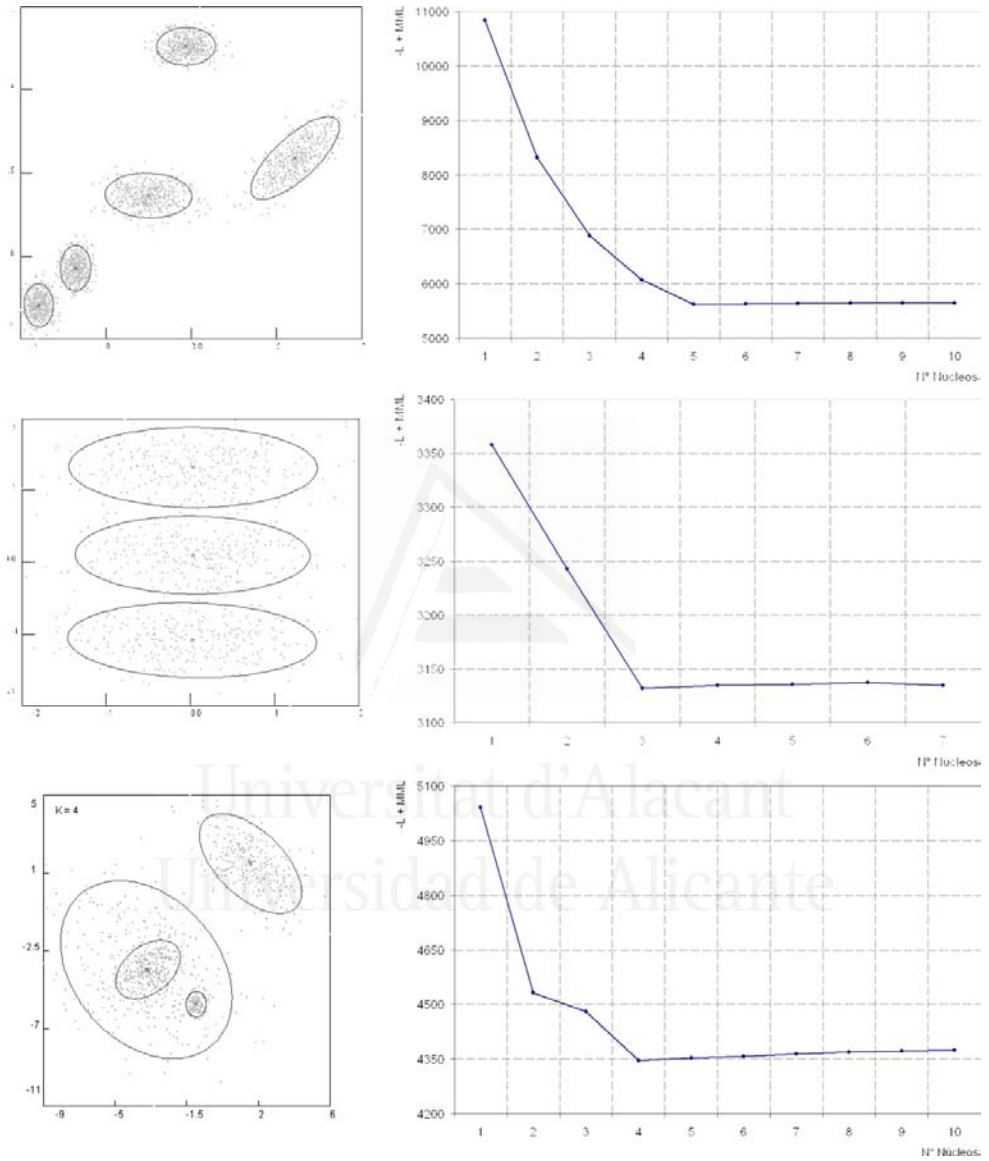

Figura 3.14: Representación de la función criterio basada en MML para tres mezclas con 5, 3 y 4 núcleos respectivamente. La función alcanza el mínimo precisamente en esos valores, para comenzar luego un ligero ascenso a medida que crece el número de núcleos.

blema de la inicialización de los nuevos núcleos. Aplicando PCA (*Principal Component Analysis*) al núcleo original, el principal autovector indicará la dirección de máxima variabilidad de los datos y podremos situar los dos nuevos núcleos en sentidos opuestos en esta dirección. Si k es el núcleo con peor grado de gausianidad, tras la división obtendremos los dos nuevos núcleos k_1 and k_2 con parámetros iniciales:

$$\Theta_{k_1} = (\mu_{k_1}, \Sigma_{k_1}) \quad \text{y} \quad \Theta_{k_2} = (\mu_{k_2}, \Sigma_{k_2}) \tag{3.54}$$

de los que las nuevas medias se obtienen de la siguiente forma:

$$\mu_{k_1} = \mu_k + \sqrt{\lambda_k}\mathbf{V} \quad \text{y} \quad \mu_{k_2} = \mu_k - \sqrt{\lambda_k}\mathbf{V}, \tag{3.55}$$

siendo λ_k el principal autovalor del núcleo k y $\mathbf{V}$ su autovector normalizado.

La inicialización de las nuevas matrices de covarianza se basaba en un criterio heurístico consistente en asociar a los nuevos núcleos la mitad de la anchura del núcleo original. Si $\lambda_k^{'}$ es el principal autovalor en ambos núcleos, entonces $\sqrt{\lambda_k^{'}} = \frac{\sqrt{\lambda_k}}{2}$ y por lo tanto, podemos expresar las nuevas matrices de covarianza de la siguiente forma:

$$\Sigma_{k_1} = \Sigma_{k_2} = \frac{1}{4}\Sigma_k. \tag{3.56}$$

Finalmente, las nuevas probabilidades a priori deben satisfacer la expresión $\sum_{k=1}^{K} \pi_k = 1$, por lo que podemos podemos repartir la probabilidad a priori del núcleo inicial en dos partes iguales según la expresión:

$$\pi_{k_1} = \pi_{k_2} = \frac{1}{2}\pi_k. \tag{3.57}$$

La figura 3.15 muestra el proceso de descomposición del núcleo según el criterio heurístico descrito con anterioridad. El desplazamiento de los núcleos se realiza únicamente en la dirección del vector de máxima variabilidad.

En [Richardson y Green, 1997], los autores desarrollaron una metodología para modelos de mezclas gausianas en problemas de una sola dimensión, en la que se utiliza el algoritmo RJMCMC basado en una serie de saltos del tipo combinar-dividir.

En la fase en la que se lleva a cabo la división, la matriz de covarianza del núcleo original debe dar lugar a dos nuevas matrices con dos restricciones:

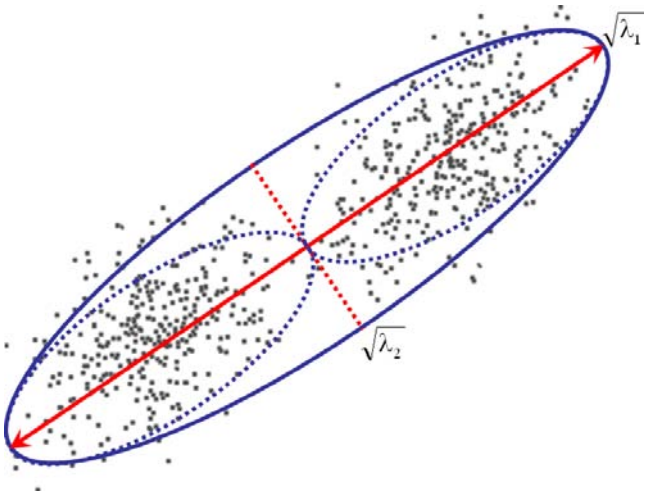

Figura 3.15: Representación gráfica de la descomposición de un núcleo en otros dos en la dirección de máxima variabilidad expresada por el autovector con mayor autovalor. Los datos no están originalmente bien ajustados, pues existe bi-modalidad.

- La dispersión promedio debe mantenerse casi constante después del proceso de división.
- Las nuevas matrices deben ser definidas positivas.

En la aproximación propuesta, se realizan tanto movimientos de combinación, seleccionando de forma aleatoria un par de núcleos que se fusionan en uno sólo; como movimientos de división, descomponiendo en dos un núcleo seleccionado igualmente de forma aleatoria. El planteamiento sugiere que los dos primeros momentos estadísticos deben mantenerse antes y después de llevar a cabo los movimientos de combinación y fusión.

A partir de la definición de mezcla gausiana de la ecuación 2.1, si consideramos que $K*$ es el componente con peor valor de gausianidad G, obtenido a partir de la expresión de la ecuación 3.39, entonces debe ser descompuesto en otros dos elementos a los que denominaremos K_1 y K_2, con sus correspondientes parámetros $\Theta_{k_1} = (\mu_{k_1}, \Sigma_{k_1})$ y $\Theta_{k_2} = (\mu_{k_2}, \Sigma_{k_2})$. En un contexto multi-variado, si seguimos las restricciones de conservación de los momentos de orden uno y dos, las correspondientes probabilidades a priori, vectores media y las matrices de covarianza deberían satisfacer las siguientes ecuaciones de división:

$$\begin{aligned} \pi_* &= \pi_1 + \pi_2 \\ \pi_*\mu_* &= \pi_1\mu_1 + \pi_2\mu_2 \\ \pi_*(\Sigma_* + \mu_*\mu_*^T) &= \pi_1(\Sigma_1 + \mu_1\mu_1^T) + \pi_2(\Sigma_2 + \mu_2\mu_2^T) \end{aligned} \tag{3.58}$$

De las ecuaciones anteriores se desprende que la operación de descomposición es un problema del tipo *ill-posed*, ya que el número de ecuaciones disponibles es menor que el número de incógnitas a determinar. Si además el espacio de observaciones pertenece a una dimensión alta, el problema se complica todavía más, debido a la necesidad de construir un número elevado de parámetros libres y a la restricción de que las nuevas matrices de covarianza deben ser definidas positivas.

Diferentes autores han propuesto soluciones a las ecuaciones anteriores en un contexto multi-dimensional. En [Zhang *et al.*, 2004] se lleva a cabo una representación espectral de la matriz de covarianza simétrica y definida positiva, que es posteriormente descompuesta en otras dos: una matriz de autovalores y otra de autovectores. A partir del carácter simétrico y definido positivo de la matriz de covarianza se proponen dos métodos para resolver las ecuaciones anteriores: Descomposición en Valores Singulares (SVD)[2] y la Descomposición de Cholesky, aunque con la importante restricción de que todos los componentes de la mezcla deben compartir la misma matriz de autovectores obtenida de esta forma.

Recientemente, en [Dellaportas y Papageorgiou, 2006] se propone una nueva descomposición espectral de la matriz de covarianza actual. De este modo, el problema original de estimar las nuevas matrices de covarianza es reemplazado por un nuevo problema consistente en la estimación de los nuevos autovalores y autovectores de las nuevas matrices de covarianza. A diferencia de la propuesta anterior, las matrices pueden ser distintas. El número de parámetros antes y después de la descomposición es $1 + d + d(d+1)/2$ y $2 + 2d + 2d(d+1)/2$ respectivamente, por lo tanto, el número adicional de parámetros a estimar en cada descomposición se incrementa cuadráticamente con la dimensión d.

Consideremos $\sum_* = V_*\Lambda_*V_*^T$ la descomposición espectral de la matriz de covarianza $\sum_*$, con $\Lambda_* = \text{diag}(\lambda j*^1, ..., \lambda j*^d)$ una matriz diagonal que contiene los autovalores de $\sum_*$ en orden creciente, $*$ la componente de la mezcla

[2] procedente del término inglés *Singular Value Decomposition*

con menor ratio de entropía G, π_*, π_1, π_2 las probabilidades a priori del componente original y los nuevos componentes respectivamente, μ_*, μ_1, μ_2 a las medias y $\sum_*, \sum_1, \sum_2$ a las matrices de covarianza. Denominemos además D a una matriz de rotación de tamaño $d \times d$ con columnas formadas por vectores ortonormales y unitarios. D se construye generando su matriz triangular inferior de forma independiente a partir de $d(d-1)/2$ distribuciones uniformes de la forma $U(0,1)$. La operación de descomposición propuesta se define como:

$$\begin{aligned}
\pi_1 &= u_1 \pi_* \\
\pi_2 &= (1 - u_1)\pi_* \\
\mu_1 &= \mu_* - (\textstyle\sum_{i=1}^{d} u_2^i \sqrt{\lambda_*^i} V_*^i) \sqrt{\frac{\pi_2}{\pi_1}} \\
\mu_2 &= \mu_* + (\textstyle\sum_{i=1}^{d} u_2^i \sqrt{\lambda_*^i} V_*^i) \sqrt{\frac{\pi_1}{\pi_2}} \\
\Lambda_1 &= \text{diag}(u_3)\text{diag}(\iota - u_2)\text{diag}(\iota + u_2)\Lambda_* \frac{\pi_*}{\pi_1} \\
\Lambda_2 &= \text{diag}(\iota - u_3)\text{diag}(\iota - u_2)\text{diag}(\iota + u_2)\Lambda_* \frac{\pi_*}{\pi_2} \\
V_1 &= DV_* \\
V_2 &= D^T V_*
\end{aligned} \tag{3.59}$$

donde, ι es un vector de unos de dimensión $d \times 1$, $u_1, u_2 = (u_2^1, u_2^2, ..., u_2^d)^T$ y $u_3 = (u_3^1, u_3^2, ..., u_3^d)^T$ son $2d + 1$ variables aleatorias adicionales necesarias para construir las nuevas probabilidades a priori, medias y autovalores para cada nuevo componente en la mezcla. Cada uno de los parámetros se obtiene de la siguiente forma:

$$\begin{aligned}
u_1 \sim \beta(2,2), u_2^1 \sim \beta(1,2d), \\
u_2^j \sim U(-1,1), u_3^1 \sim \beta(1,d), u_3^j \sim U(0,1)
\end{aligned} \tag{3.60}$$

con $j = 2, ..., d$ y $\beta(.)$ una distribución Beta.

En la figura 3.16 se muestra una descripción gráfica del proceso de descomposición de un núcleo para el caso bi-dimensional. Las direcciones y magnitudes de variabilidad se definen mediante los autovectores y autovalores de la matriz de covarianza. A diferencia de la propuesta de división inicial, el desplazamiento de los nuevos núcleos se produce en todas las direcciones del espacio de dimensiones del problema, en lugar de hacerlo únicamente en la dirección de mayor autovalor.

La descomposición así realizada es una solución al sistema de ecuaciones

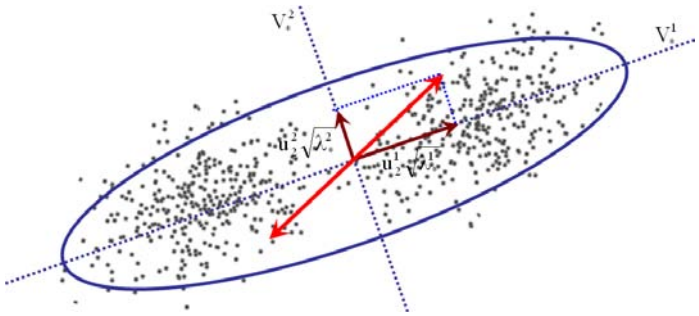

Figura 3.16: Ejemplo en 2-D de descomposición de un núcleo en dos nuevos núcleos. La necesidad de mantener u_2^1 positivo y permitir que u_2^2 varíe en el intervalo [-1, 1] es debido a que el desplazamiento a lo largo de la dirección del principal autovalor se consigue mediante los nuevos valores de μ_1 y μ_2, mientras que el desplazamiento a lo largo del eje más corto requiere valores positivos y negativos de u_2^2.

expuesto en 3.58 y permite una inicialización adecuada para los nuevos parámetros de la mezcla tras introducir un nuevo componente en la misma. Esta inicialización óptima evita la necesidad de realizar pasos EM parciales para ajustar los parámetros del modelo antes de continuar con la ejecución global del algoritmo EM como en el caso de [Vlassis y Likas, 2000].

3.4.4. Algoritmo

En este apartado se va a realizar una descripción detallada del algoritmo propuesto para el ajuste dinámico de un modelo de mezclas finitas a un conjunto de datos. Para ello se emplearán los métodos de estimación de entropía y la técnica de inicialización de los parámetros de los nuevos núcleos descrita en los apartados anteriores. Denominamos al algoritmo EBEM, siglas de la expresión en inglés *Entropy-based EM algorithm* o Algoritmo EM basado en entropía. La técnica emplea esta medida estadística como criterio para comprobar la calidad del ajuste de cada núcleo individual de los datos de su vecindad. Además uno de los criterios de parada se basa igualmente en la entropía global del modelo para un conjunto de núcleos determinado.

El algoritmo comienza con un sólo núcleo, cuyos parámetros de media y matriz de covarianza se corresponden con los de la media y covarianza muestral del total de observaciones puesto que, como se demuestra en [Mitchell, 1997], la hipótesis de máxima verosimilitud para la media en el ca-

so de una sola distribución normal a partir de un conjunto de observaciones de la misma $y_1, y_2, ..y_N$ es la que minimiza la suma de los errores cuadráticos sobre el conjunto de N observaciones, que en este caso es minimizada por la expresión de la media muestral:

$$\mu_1 = \frac{1}{N}\sum_{=1}^{N} y_i \tag{3.61}$$

Del mismo modo, se obtendría el valor inicial para la matriz de covarianza:

$$\Sigma_1 = [S_{jk}]_{d\times d}, \text{ y } S_{jk} = \frac{1}{N-1}\sum_{i=1}^{N}(y_i^j - \mu_i^j)(y_i^k - \mu_i^k) \tag{3.62}$$

con $j, k = 1, 2, ..d$ y d la dimensionalidad del problema. Por otro lado, el valor inicial para la probabilidad a priori π_1 tiene valor 1, pues partimos de un sólo núcleo que contiene la totalidad de la probabilidad de los datos. Con este conjunto de parámetros inicial se lleva a cabo el **paso E**, en el que se calcula la probabilidad de que cada observación y_n haya sido generada por un núcleo k, es decir $p(k|y_n)$, $\forall n, k$, según la expresión de la ecuación 2.7.

A continuación se ejecuta el **paso M** y se obtiene el nuevo conjunto de parámetros $\Theta^*(i+1)$, es decir los valores de medias μ_k, matrices de covarianza Σ_k y probabilidades a priori π_k para cada uno de los núcleos actuales de la mezcla (uno originalmente), según las expresiones de las ecuaciones 2.3.

Se calcula el logaritmo de la verosimilitud en la iteración actual, según la expresión de la ecuación 2.5 y se compara con el obtenido en la iteración anterior. El proceso se repite mientras el incremento en el logaritmo de la verosimilitud supere un umbral establecido a priori (CONVERGENCE_TH).

Una vez finalizado el proceso anterior debemos aplicar alguno de los dos criterios de parada expuestos en el apartado anterior:

- Si se ha escogido el criterio basado en la medida de gausianidad, se calcula la entropía máxima teórica de la mezcla $H(Y)$ y la entropía real ponderada con las probabilidades a priori de cada núcleo $H_{max}(Y)$, según las expresiones de las ecuaciones 3.36 y 3.37 por alguno de los dos métodos expuestos en el apartado anterior: *Ventanas de Parzen* o *Minimal Spanning Trees*. A continuación, se comparan ambas cantidades

según alguno de los criterios propuestos en la ecuación 3.39 para tener una medida del grado de gausianidad de la mezcla de modo global. Si el valor obtenido queda por encima de un umbral determinado (ENTROPY_TH), consideramos que los datos no están correctamente ajustados con el número actual de núcleos, por lo que se selecciona el núcleo k^* con peor grado de gausianidad individual de la mezcla actual y se descompone en otros dos cuyos valores iniciales se han especificado en el apartado anterior.

- Si el criterio ha sido el basado en el principio de longitud mínima (MDL, MMDL o MML), se dispone de una función de energía que puede ser evaluada cada vez que el algoritmo EM converge para el número actual de núcleos k. Puesto que dicha función de energía presenta un mínimo cuando el número de núcleos K es óptimo, debemos comparar el valor actual de la función con k núcleos con el valor obtenido para $k-1$. Si el valor es superior, entonces el algoritmo finaliza y el número óptimo de núcleos es $k-1$. De este modo el algoritmo necesita $k+1$ ejecuciones del algoritmo EM ($k = 1, 2, ..., K, K+1$), con K el número óptimo de núcleos.

  Este planteamiento supone una clara mejora respecto a otros basados en criterios similares [Figueiredo *et al.*, 1999] [Figueiredo y Jain, 2002] que parten de un número elevado de núcleos. Por un lado, el número de iteraciones necesarias es superior y por otro deben recorrer la totalidad del espacio de posibles valores de k hasta llegar a un sólo núcleo. Entonces se debe determinar cual de los modelos proporciona el valor mínimo de la función criterio. Este hecho es debido a que, si bien la función presenta un comportamiento monótonamente decreciente para $1 \leq k \leq k^*$, no ocurre lo mismo para $k^* \leq k \leq k_{max}$, existiendo la posibilidad de aparición de mínimos locales que obligan a realizar un recorrido completo para todos los valores de $1 \leq k \leq k_{max}$, con k_{max} un parámetro que representa el número de núcleos de partida.

El proceso de selección del núcleo que peor ajusta la densidad de probabilidad de su vecindad se lleva a cabo según la siguiente expresión:

$$k^* = \arg\max_k \left\{ \pi_k \frac{(H_{max}(k) - H_{real}(k))}{H_{max}(k)} \right\}. \tag{3.63}$$

El siguiente paso consiste en la sustitución del núcleo seleccionado en el paso anterior k^* por dos nuevos núcleos k_1 y k_2 cuyos conjuntos de parámetros iniciales Θ_1 y Θ_2 se obtienen mediante el proceso descrito en la sección 3.4.3. A continuación, comienza la iteración $i+1$ de los pasos E y M del algoritmo clásico con $k+1$ núcleos. Los valores iniciales para los $k-1$ núcleos que no han sido afectados por el proceso de división son los obtenidos tras la finalización de la iteración anterior i.

El proceso continúa mientras el valor obtenido de Gausianidad G no alcance un umbral ENTROPY_TH definido previamente o bien se alcance un mínimo en el valor de la función de energía obtenida a partir del principio de mínima descripción. En la figura 3.17 se muestra una descripción algorítmica detallada del proceso completo para criterio de parada mediante gausianidad. El algoritmo requiere como parámetros de entrada: el umbral de convergencia para el logaritmo de la verosimilitud CONVERGENCE_TH y el umbral de gausianidad mínimo para considerar bien ajustados los datos ENTROPY_TH. Los parámetros del núcleo inicial $\Theta(0) = \{\theta_1, \pi_1\}$, se obtienen a partir del conjunto completo de datos. Como resultado se obtiene el número óptimo de núcleos K y el conjunto de parámetros asociado a cada uno de ellos Θ^*.

En la figura 3.18 se muestra la variante del algoritmo para criterio de parada basado en mínima longitud (de descripción en sus dos variantes o de mensaje). El algoritmo requiere como parámetro de entrada únicamente el umbral de convergencia para el logaritmo de la verosimilitud CONVERGENCE_TH. Los parámetros del núcleo inicial $\Theta(0) = \{\theta_1, \pi_1\}$, son igualmente obtenidos a partir del conjunto completo de datos. Como resultado se obtiene el número óptimo de núcleos K y el conjunto de parámetros asociado a cada uno de ellos Θ^*. Cuando el algoritmo EM converge para un número k de núcleos, se calcula la función de energía $\mathcal{C}(\Theta(i))$ en la iteración i y se compara el valor obtenido en la iteración $i-1$. Si el nuevo valor es inferior al anterior, todavía no se ha alcanzado el mínimo y por tanto, debemos seleccionar el peor núcleo y descomponerlo en dos del mismo modo que en la versión con criterio de parada basado en gausianidad. Por el contrario, si el

ALGORITMO EBEM CRITERIO GAUSIANIDAD

Entrada: CONVERGENCE_TH, ENTROPY_TH.

Salida: Modelo de mezclas óptimo: K, Θ^*

$K \leftarrow 1$, $i \leftarrow 0$, $\pi_1 = 1$

$\Theta_1 \leftarrow \{\mu_1, \Sigma_1\}$ Obtenidos a partir del conjunto completo de datos.

$\mu_1 = \frac{1}{N}\sum_{i=1}^{N} y_i$

$\Sigma_1 = [S_{jk}]_{d\times d}$, $S_{jk} = \frac{1}{N-1}\sum_{i=1}^{N}(y_i^j - \mu_i^j)(y_i^k - \mu_i^k)$, con $j, k = 1, 2, ..d$

Final $\leftarrow$ false

repeat

$i \leftarrow i + 1$

repeat

$p(k|\mathbf{y}_n) = \frac{\pi_k p(\mathbf{y}^{(n)}|k)}{\sum_{j=1}^{K}\pi_j p(\mathbf{y}^{(n)}|k)}$

$\pi_k = \frac{1}{N}\sum_{n=1}^{N} p(k|\mathbf{y}_n)$, $\mu_k = \frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n)\mathbf{y}_n}{\sum_{n=1}^{N} p(k|\mathbf{y}_n)}$, $\Sigma_k = \frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n)(\mathbf{y}_n-\mu_k)(\mathbf{y}_n-\mu_k)^T}{\sum_{n=1}^{N} p(k|\mathbf{y}_n)}$

Calcular logaritmo verosimilitud en iteración i:

$\ell(Y|\Theta(i)) = \sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k p(y_k|\Theta(i)_k)$

until: $|\ell(Y|\Theta(i)) - \ell(Y|\Theta(i-1))| <$ CONVERGENCE_TH

Calcular $H_{real}(Y)$ y $H_{max}(Y)$ y Gausianidad G

$G = \sum_{k=1}^{K}\pi_k\left(1 - \frac{H_{real}(k)}{H_{max}(k)}\right)$

if ($G >$ ENTROPY_TH)

Seleccionar k^* con peor ratio individual

$k^* = \arg\max_k \{\pi_k (H_{max}(k) - H_{real}(k))/H_{max}(k)\}$

Descomponer k^* en k_1 y k_2

Inicializar parámetros Θ_1 y Θ_2

$\mu_1 = \mu_* - (\sum_{i=1}^{d} u_2^i\sqrt{\lambda_*^i}V_*^i)\sqrt{\frac{\pi_2}{\pi_1}}$, $\mu_2 = \mu_* + (\sum_{i=1}^{d} u_2^i\sqrt{\lambda_*^i}V_*^i)\sqrt{\frac{\pi_1}{\pi_2}}$

$\Lambda_1 = \text{diag}(u_3)\text{diag}(\iota - u_2)\text{diag}(\iota + u_2)\Lambda_*\frac{\pi_*}{\pi_1}$

$\Lambda_2 = \text{diag}(\iota - u_3)\text{diag}(\iota - u_2)\text{diag}(\iota + u_2)\Lambda_*\frac{\pi_*}{\pi_2}$

$V_1 = DV_*$, $V_2 = D^T V_*$

$\pi_1 = u_1\pi_*$, $\pi_2 = (1 - u_1)\pi_*$

$K \leftarrow K + 1$

else

Final $\leftarrow$ true

until: Final = true

Figura 3.17: Algoritmo EM basado en entropía con parada cuando el grado de gausianidad de la mezcla es inferior a un umbral.

valor obtenido es mayor, la naturaleza de la función implica que el valor mínimo se obtenía justo en la iteración anterior, por lo que el conjunto óptimo de parámetros coincide con el que se había obtenido en la iteración anterior: $K = K - 1$ y $\Theta^* = \Theta(i-1)$.

De este modo, a diferencia de los métodos con estrategia basada en la fusión de núcleos a partir de un número inicial relativamente elevado, en nuestra propuesta no es necesario almacenar el conjunto de parámetros para cada una de las iteraciones, siendo suficiente con el almacenamiento de los parámetros de las iteraciones actual y anterior. Esto es posible por el comportamiento monótono decreciente de la función de energía para un número de núcleos con intervalo comprendido entre uno y el óptimo.

Otra ventaja de partir de un sólo núcleo frente a las estrategias de fusión es que si k es demasiado elevado, podría ocurrir que ningún componente tuviera un soporte inicial suficiente $\sum_{n=1}^{N} p(k|\mathbf{y}_n) < N/2$ [Figueiredo y Jain, 2002] y por tanto no podría determinarse α_m, con $m = 1, 2, ..., k$. Para evitar este problema, en el trabajo citado se recurre a una variante del algoritmo EM, denominada CEM^2 [Celeux *et al.*, 1999], de un coste computacional mayor que el estándar al tener que realizar varios pasos **E** para recalcular $p(k|\mathbf{y})$. De este modo, si la probabilidad a priori de un núcleo alcanza el valor cero, su masa de probabilidad se redistribuye entre el resto de núcleos, incrementándose la probabilidad de que se mantenga.

ALGORITMO EBEM CRITERIO MÍNIMA LONGITUD

Entrada: CONVERGENCE_TH.

Salida: Modelo de mezclas óptimo: K, Θ^*

$K \leftarrow 1$, $i \leftarrow 0$, $\pi_1 = 1$, $\Theta_1 \leftarrow \{\mu_1, \Sigma_1\}$ Obtenidos a partir del conjunto completo de datos.

$\mu_1 = \frac{1}{N}\sum_{i=1}^{N} y_i$, $\Sigma_1 = [S_{jk}]_{d\times d}$, $S_{jk} = \frac{1}{N-1}\sum_{i=1}^{N}(y_i^j - \mu_i^j)(y_i^k - \mu_i^k)$, con $j, k = 1, 2, ..d$

Final $\leftarrow$ false

repeat

$i \leftarrow i + 1$

repeat

$p(k|\mathbf{y}_n) = \frac{\pi_k p(\mathbf{y}^{(n)}|k)}{\sum_{j=1}^{K}\pi_j p(\mathbf{y}^{(n)}|k)}$

$\pi_k = \frac{1}{N}\sum_{n=1}^{N} p(k|\mathbf{y}_n)$, $\mu_k = \frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n)\mathbf{y}_n}{\sum_{n=1}^{N} p(k|\mathbf{y}_n)}$, $\Sigma_k = \frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n)(\mathbf{y}_n-\mu_k)(\mathbf{y}_n-\mu_k)^T}{\sum_{n=1}^{N} p(k|\mathbf{y}_n)}$

Calcular logaritmo verosimilitud en iteración i:

$\ell(Y|\Theta(i)) = \sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k p(y_k|\Theta(i)_k)$

until: $|\ell(Y|\Theta(i)) - \ell(Y|\Theta(i-1))| <$ CONVERGENCE_TH

Seleccionar k^* con peor ratio individual

$k^* = \arg\max_k \{\pi_k (H_{max}(k) - H_{real}(k))/H_{max}(k)\}$

Calcular $\mathcal{C}_{MDL}$ ó $\mathcal{C}_{MMDL}$ ó $\mathcal{C}_{MML}$ para la iteración i

$\mathcal{C}_{MDL}(\Theta(i)) = -\ell(Y|\Theta(i)) + \frac{N(k)}{2}\log n$

$\mathcal{C}_{MMDL}(\Theta(i)) = -\ell(Y|\Theta(i)) + \frac{N(k)}{2}\log n + \frac{N(1)}{2}\sum_{i=1}^{k}\log\alpha_i$

$\mathcal{C}_{MML}(\Theta(i)) = \frac{N(1)}{2}\sum_{i=1}^{k}\log\left(\frac{n\alpha_i}{12}\right) + \frac{k}{2}\log(\frac{n}{12}) + \frac{k(N(1)+1)}{2} - \ell(Y|\Theta(i))$

if $(\mathcal{C}(\Theta(i)) \geq \mathcal{C}(\Theta(i-1)))$

Final $\leftarrow$ true

$K \leftarrow K - 1$, $\Theta^* \leftarrow \Theta(i-1)$

else

Descomponer k^* en k_1 y k_2

Inicializar parámetros Θ_1 y Θ_2

$\mu_1 = \mu_* - (\sum_{i=1}^{d} u_2^i\sqrt{\lambda_*^i}V_*^i)\sqrt{\frac{\pi_2}{\pi_1}}$, $\mu_2 = \mu_* + (\sum_{i=1}^{d} u_2^i\sqrt{\lambda_*^i}V_*^i)\sqrt{\frac{\pi_1}{\pi_2}}$

$\Lambda_1 = \text{diag}(u_3)\text{diag}(\iota - u_2)\text{diag}(\iota + u_2)\Lambda_*\frac{\pi_*}{\pi_1}$

$\Lambda_2 = \text{diag}(\iota - u_3)\text{diag}(\iota - u_2)\text{diag}(\iota + u_2)\Lambda_*\frac{\pi_*}{\pi_2}$

$V_1 = DV_*$, $V_2 = D^T V_*$

$\pi_1 = u_1\pi_*$, $\pi_2 = (1 - u_1)\pi_*$

$K \leftarrow K + 1$

until: Final = true

Figura 3.18: Algoritmo EM basado en entropía con criterio de parada empleando mínima longitud (de descripción o de mensaje). El algoritmo finaliza cuando se produce un incremento en la función de energía.

Capítulo 4

Experimentos y aplicaciones

Para comprobar el correcto funcionamiento del algoritmo se han llevado a cabo diferentes tipos de experimentos. Inicialmente mostramos los resultados obtenidos como técnica para estimar la densidad de probabilidad asociada a un conjunto de datos, tanto en comparación con el algoritmo EM clásico como con otras técnicas similares descritas en el capítulo 2. Posteriormente mostramos los resultados obtenidos en tareas de clasificación no supervisada de un conjunto de observaciones. A continuación, mostramos el comportamiento de la técnica en el contexto de la segmentación de color y finalizamos con una comparación entre los diferentes criterios de parada del algoritmo para la selección del orden del modelo.

4.1. Estimación de densidad de probabilidad

El primer experimento llevado a cabo consiste en la comparación entre el método propuesto y el algoritmo EM clásico. Para ello generamos 2500 observaciones pertenecientes 5 distribuciones gausianas en dos dimensiones, empleando diferentes probabilidades a priori, medias y matrices de covarianza. A continuación se ejecuta el algoritmo propuesto en el contexto de estimación

de densidad de probabilidad empleando gausianidad como criterio de parada, para posteriormente repetir el experimento con el algoritmo EM clásico y comparar los resultados obtenidos.

El conjunto de datos generado presenta 3 de los núcleos claramente separados y dos de ellos con medias muy próximas.

4.1.1. Resultados con EBEM

Los parámetros del modelo han sido: umbral de gausianidad de $0{,}1$ y umbral de convergencia de $0{,}001$ para el algoritmo EM. El algoritmo converge después de 30 iteraciones, tanto para el método de estimación de entropía *plug-in* como con el *non plug-in*, estimando correctamente el número óptimo de clases $k = 5$. Para comprobar la robustez del algoritmo propuesto, se han añadido múltiples falsos positivos al conjunto de datos generado previamente. El tamaño de muestra seleccionado para la estimación de la entropía por el método *plug-in* de las ventanas de Parzen ha sido 75. Este reducido tamaño permite encontrar la solución de forma rápida y estimar la entropía con la suficiente precisión. Los parámetros empleados para generar los datos artificiales han sido los siguientes:

$$\Sigma_1 = \begin{bmatrix} 0{,}20 & 0{,}00 \\ 0{,}00 & 0{,}30 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0{,}60 & 0{,}15 \\ 0{,}15 & 0{,}60 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 0{,}40 & 0{,}00 \\ 0{,}00 & 0{,}25 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0{,}60 & 0{,}00 \\ 0{,}00 & 0{,}30 \end{bmatrix},$$

$$\Sigma_5 = \begin{bmatrix} 0{,}20 & 0{,}00 \\ 0{,}00 & 0{,}30 \end{bmatrix}.$$

$$\begin{gathered} \pi_k = 0{,}2 \\ \mu_1 = [-1,-1]^T, \mu_2 = [6,3]^T, \mu_3 = [3,6]^T \\ \mu_4 = [2,2]^T, \mu_5 = [0,0]^T. \end{gathered} \tag{4.1}$$

En la figura 4.1 mostramos la evolución del algoritmo. Este comienza con un sólo núcleo con media y covarianza iniciales obtenidas a partir del conjunto de datos. En las sucesivas iteraciones del algoritmo se van introduciendo nuevos núcleos en las zonas en las que se ha realizado el peor ajuste, es decir, en las zonas representadas por núcleos para los que la expresión de la

ecuación 3.63 tiene un valor menor. El algoritmo se detiene correctamente al alcanzar el número óptimo de núcleos, fijado en 5 en este caso. En la 2ª imagen de la secuencia el núcleo inicial se descompone en dos. El primero de ellos representa a los dos núcleos de la parte superior mientras el segundo representa a los tres núcleos inferiores. En la 3ª imagen de la secuencia el núcleo superior se ha descompuesto en dos que ya ajustan perfectamente a los datos de su vecindad. En la 4ª imagen los tres núcleos superiores ya están correctamente ajustados. Los dos núcleos más próximos de la zona inferior de la imagen son los últimos en ajustarse. La 5ª imagen de la secuencia muestra el resultado final tras la ejecución del algoritmo.

4.1.2. Resultados con EM clásico

Además, realizamos el mismo experimento pero aplicando el algoritmo EM clásico fijando en 5 el número de núcleos. Dada la sensibilidad del método original a la inicialización, llevamos a cabo 20 ejecuciones del algoritmo con los datos anteriores, situando aleatoriamente entre el espacio de observaciones cada uno de los núcleos iniciales. En 18 de los 20 experimentos, el algoritmo clásico converge hacia máximos locales de la función de verosimilitud, no encontrando por tanto la solución óptima al problema. El número de iteraciones en promedio necesarias para lograr la convergencia es de 95 (siendo 250 el máximo y 23 el mínimo). Sólo en 2 casos, el algoritmo EM clásico encuentra el máximo global, empleando para ello 21 y 31 iteraciones respectivamente. Por lo tanto, nuestra propuesta trata dos problemas básicos del algoritmo EM clásico para la determinación de la densidad de probabilidad asociada a un conjunto de datos: la inicialización y la selección del orden del modelo.

En la figura 4.2 mostramos el resultado final obtenido mediante el algoritmo clásico para dos de las 20 ejecuciones. La elevada distancia entre la mayor parte de los núcleos, frente a la proximidad de los dos inferiores provoca que, salvo que en la inicialización aleatoria se generen dos núcleos en las proximidades de éstos, el resultado converge a un máximo local. En este caso, ambos núcleos quedan descritos por un solo componente de la mezcla, tendiendo uno de los núcleos sobrantes a representar observaciones claramente representadas por otro núcleo, con varianzas y probabilidad a priori próximas a

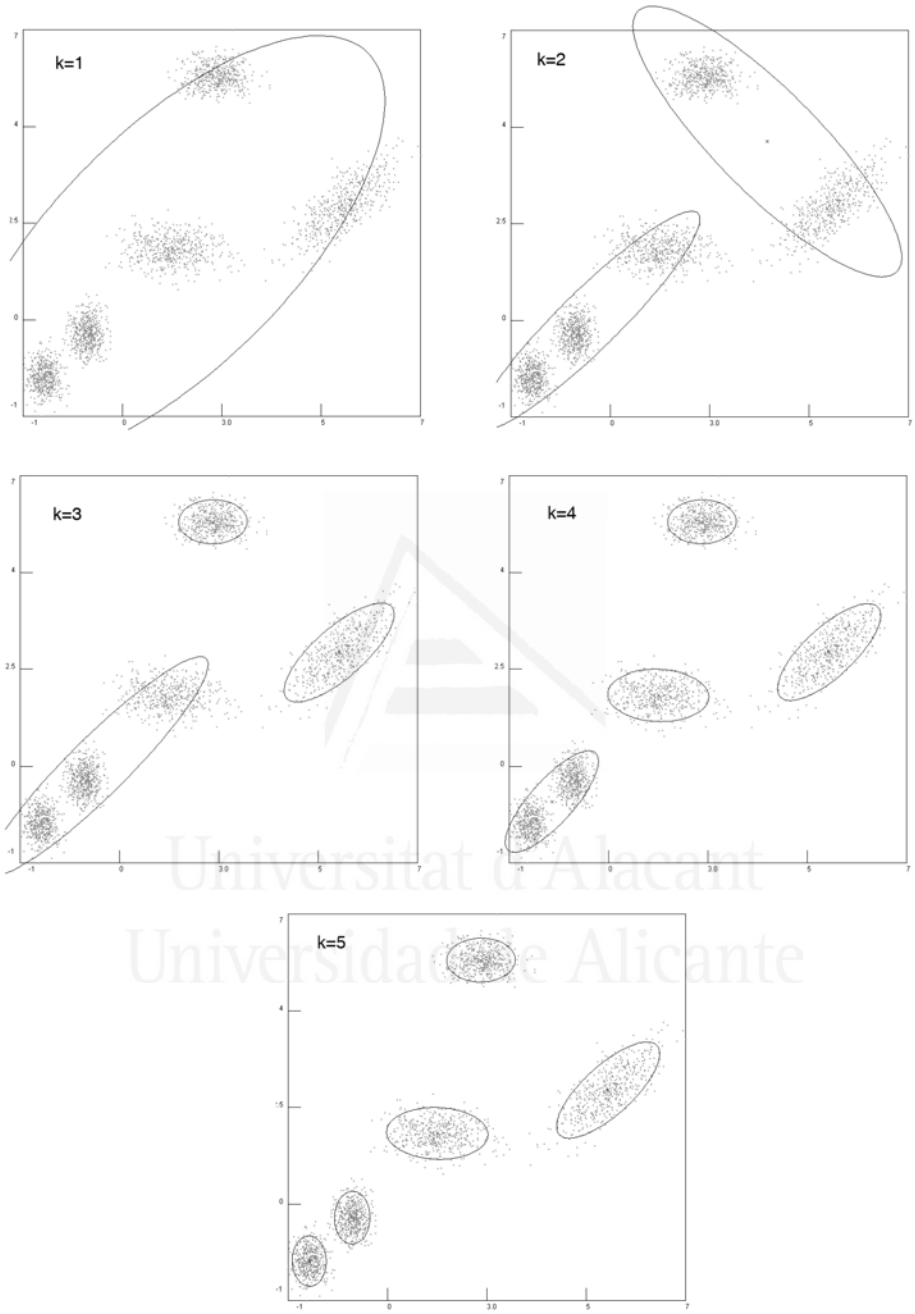

Figura 4.1: Evolución de nuestro algoritmo desde un núcleo inicial hasta la solución óptima compuesta por 5 núcleos. La secuencia muestra la posición de los núcleos tras la realización de los pasos E y M y alcanzar el umbral de convergencia para k núcleos.

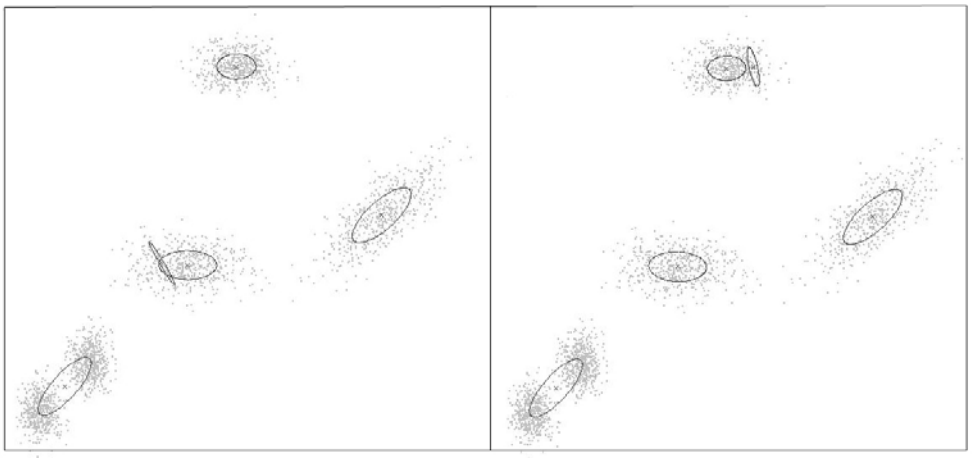

Figura 4.2: El algoritmo EM clásico es sensible a la inicialización de los parámetros del modelo. El algoritmo converge hacia un máximo local en 18 de las 20 ejecuciones con valores iniciales obtenidos de forma aleatoria.

cero.

En la mayoría de las ejecuciones que finalizan en máximos locales, el principal problema se debe a la gran distancia entre la mayor parte de los núcleos (salvo los dos inferiores). Si no hay un número suficiente de núcleos relativamente cercanos a los dos conjuntos de datos más próximos, el algoritmo no es capaz de determinar la bimodalidad inherente a esa zona del espacio de observaciones, empleando un sólo núcleo para representar observaciones pertenecientes a dos distribuciones gausianas. Por otra parte, la necesidad de mantener constante el número de componentes del modelo a 5, obliga a representar artificialmente con dos núcleos zonas en las que realmente sólo existe una distribución, forzando a que la varianza en alguna de las dimensiones tienda a cero, así como la probabilidad a priori de dicho núcleo *erróneo*.

4.1.3. Comparación con otros métodos

Una vez comprobado que el algoritmo mejora claramente el comportamiento del algoritmo EM clásico para problemas en los que los núcleos están claramente separados, comparamos los resultados con los obtenidos por otros métodos basados igualmente en EM y que han sido revisados con detalle en la sección 2. Hemos seleccionado [Figueiredo y Jain, 2002], puesto que

mejora significativamente al resto de métodos propuestos con anterioridad. De los múltiples experimentos que se llevan a cabo en el citado artículo, hemos escogido el mostrado en la figura 4.3. En ella puede observarse la evolución de nuestro algoritmo en un caso muy diferente al anterior, en el que los componentes de la mezcla están muy solapados.

Esta es, probablemente, una de las situaciones más complejas para ajustar correctamente el modelo, puesto que dos de los cuatro componentes de la mezcla comparten la misma media y poseen diferentes matrices de covarianza. Además, existe un núcleo de reducida probabilidad a priori y matriz de covarianza con varianzas igualmente reducidas, que se encuentra íntegramente incluido en otro núcleo de varianza mayor. El experimento se ha llevado a cabo a partir de la generación de 1000 muestras pertenecientes a 4 componentes con los siguientes parámetros:

$$\pi_1 = \pi_2 = \pi_3 = 0{,}3, \pi_4 = 0{,}1,$$

$$\mu_1 = \mu_2 = [-4, -4]^T, \mu_3 = [2, 2]^T, \mu_4 = [-1, -6]^T,$$

$$\Sigma_1 = \begin{bmatrix} 1 & 0{,}5 \\ 0{,}5 & 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0{,}125 & 0 \\ 0 & 0{,}125 \end{bmatrix}$$

En [Figueiredo y Jain, 2002] el algoritmo comienza con 20 núcleos inicializados aleatoriamente. Dependiendo de la ejecución realizada, el algoritmo converge en unas 200 iteraciones, aunque como veremos posteriormente, la solución no es siempre óptima. La figura 4.3 muestra la evolución de nuestro algoritmo, comenzando con un sólo núcleo inicial, hasta alcanzar los 4 núcleos existentes. Para la prueba empleamos nuevamente el método *plug-in* de estimación de entropía utilizando 75 muestras del total disponible para ello y la medida de gausianidad como criterio de parada. El umbral de convergencia del algoritmo es $0{,}001$ y el umbral de gausianidad igualmente $0{,}10$. El algoritmo converge en 141 iteraciones ajustando correctamente el modelo y estimando el orden del mismo en 4.

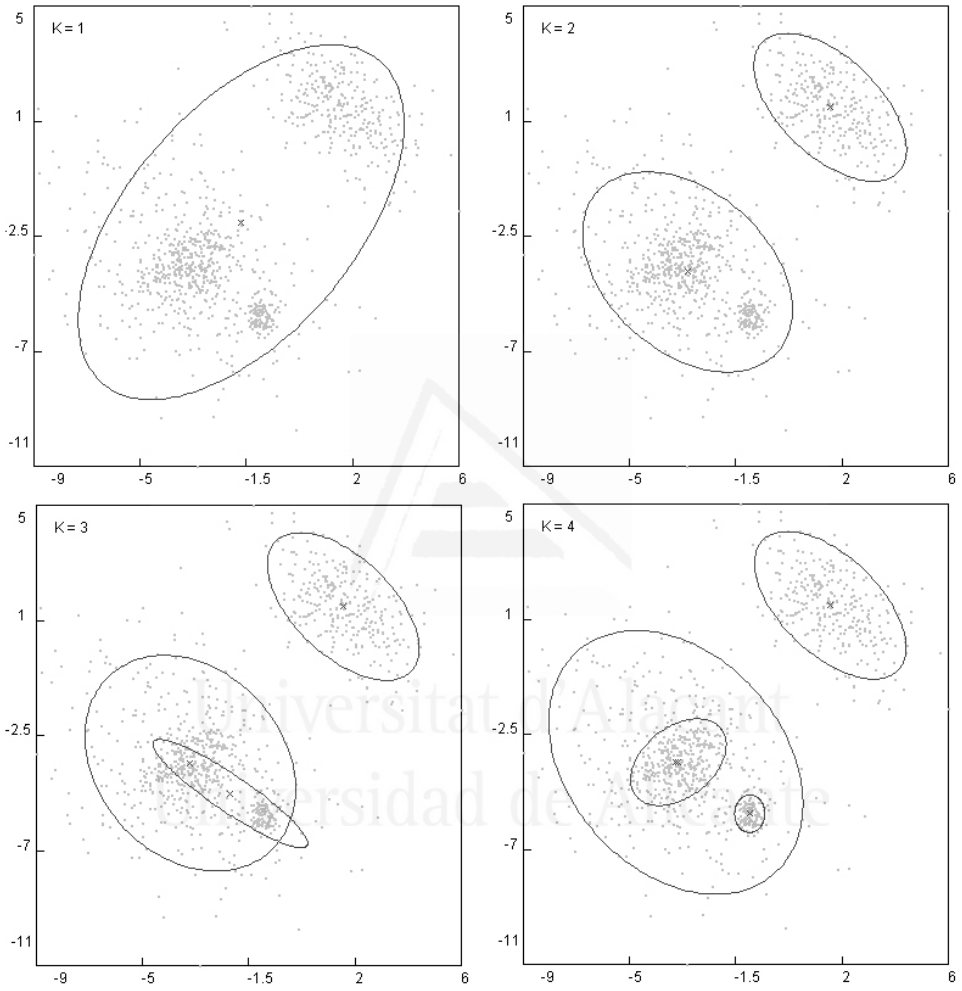

Figura 4.3: Ajuste de un modelo de mezcla gausiana con componentes que se solapan e incluso que contienen a otros de menor probabilidad y varianza. El algoritmo comienza con un sólo componente y selecciona correctamente el orden del modelo, fijado en 4.

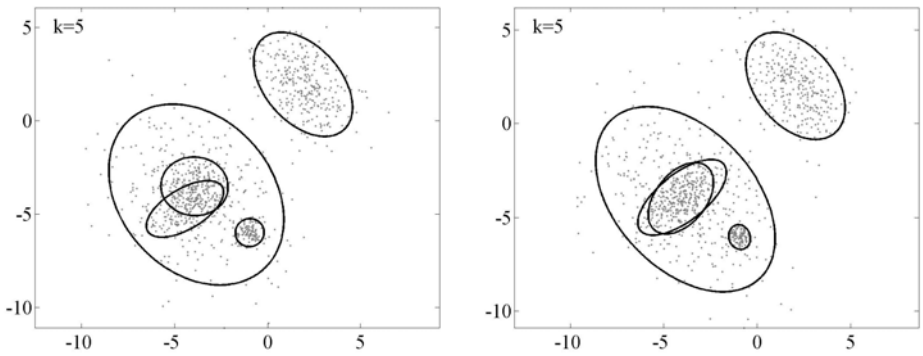

Figura 4.4: En el método de fusión de núcleos a partir del criterio basado en el principio de longitud de descripción mínima, la inicialización aleatoria del conjunto inicial de núcleos puede igualmente generar un óptimo local de la función de coste. En ambas imágenes el algoritmo genera un núcleo adicional en la zona del núcleo de amplia varianza que incluye a otros dos, seleccionando incorrectamente el orden del modelo.

En nuestras pruebas, el método propuesto en [Figueiredo y Jain, 2002], puede converger a un máximo local si el número inicial de núcleos no es suficientemente alto, pues la inicialización aleatoria de los componentes puede hacer que alguna zona del espacio no quede correctamente representada si no hay suficientes núcleos en sus proximidades. En la figura 4.4 mostramos el resultado de dos ejecuciones con un número inicial de núcleos de 25 y 20 respectivamente, que no logran alcanzar el óptimo global de la función de energía.

Por otra parte, el proceso de fusión de componentes requiere ejecutar el algoritmo desde los 20 núcleos iniciales hasta 1, seleccionando el número que mejor valor proporciona de la función de coste. Evidentemente, cuanto mayor sea el número inicial de núcleos, mayor tiempo de cómputo será necesario, con independencia de cual sea el número óptimo de componentes. En nuestro método, al comenzar por un sólo núcleo, el algoritmo no explora nuevas soluciones cuando se alcanza el umbral de entropía especificado, obteniendo tiempos de cómputo claramente inferiores si el número de componentes de la mezcla no es elevado.

4.2. Clasificación de patrones

En el segundo tipo de experimento comprobamos el funcionamiento del algoritmo en el contexto de clasificación no supervisada de un conjunto de observaciones. Para ello, hemos aplicado el método propuesto al conjunto de datos ampliamente conocido *Iris* [Blake y Merz, 1998], que contiene tres clases de 50 observaciones en un espacio de 4 dimensiones, referidas a tres clases de plantas del tipo Iris: *Versicolor*, *Virginica* y *Setosa*. 50 observaciones no son suficientes para construir la densidad de probabilidad empleando el método de las ventanas de Parzen en un espacio de 4 dimensiones por lo que, para comprobar el funcionamiento del algoritmo, se han generado 300 patrones de entrenamiento a partir de los valores de media y varianza de los patrones originales del conjunto y hemos probado el algoritmo con los 150 patrones originales. Comenzando del modo habitual con un sólo núcleo $K = 1$, el método selecciona correctamente $K = 3$ tras 20 iteraciones. A continuación, se construye un clasificador basado en el máximo a posteriori, con una tasa de acierto del $98\,\%$ (sólo 3 *Virgínica* son clasificadas como *Setosa*). Para ello obtenemos, para cada observación y_j perteneciente al conjunto inicial de observaciones y, cuál de los núcleos k obtenidos por el algoritmo genera un mayor valor para $P(k|\Theta_j)$. Si aplicamos el Teorema de Bayes a la probabilidad anterior y denominamos $m(y_j)$ a la clase a la que pertenece una observación y_j, tenemos:

$$m(y_j) = \arg \max_k \{\pi_k P(y_j|\Theta_k)\}\,. \tag{4.2}$$

Para la estimación de la densidad de probabilidad con el método de las ventanas de Parzen se han empleado 175 muestras. El incremento en el número de dimensiones requiere un mayor número de datos para la estimación de la densidad de probabilidad. Esta es la principal desventaja de los métodos *plug-in* de estimación de entropía frente al método basado en MST, que no requiere la estimación previa de la densidad de probabilidad. El umbral de gausianidad se ha establecido en $0{,}3$. Con la estimación de entropía basada en el cálculo del MST, en el que no es necesario llevar a cabo la estimación de la densidad de probabilidad, el algoritmo puede ser ejecutado con el conjunto de patrones original, obteniéndose la misma tasa de aciertos en el proceso de clasificación. La tabla 4.1 muestra un resumen de los parámetros y resultados

-	Parzen	MST
Umbral	0,3	0,3
Iteraciones	20	20
Muestras	300	150
Tasa acierto	98 %	98 %

Tabla 4.1: Experimento de clasificación empleando los métodos de estimación de entropía por el método de las ventanas de Parzen y MST. Con ambos métodos se obtiene la misma tasa de acierto, pero en el segundo caso es suficiente con el número reducido de observaciones disponibles.

tras la aplicación de las dos versiones de estimación de entropía.

4.3. Segmentación de color

Una utilidad diferente del algoritmo es su aplicación al problema de la segmentación de imágenes en color. La segmentación es un requisito previo para la resolución de muchos problemas de visión por computador, tales como clasificación de imágenes o recuperación y reconocimiento de objetos. Existen dos planteamientos principales para la segmentación de imágenes: supervisados y no supervisados.

El planteamiento no supervisado consiste en la división de una imagen en color en varias regiones homogéneas de forma automática, a partir de alguna medida de similitud. Este es el planteamiento más interesante, dentro del cual podemos encontrar modelos probabilísticos, como estadísticos bayesianos en [Zhu *et al.*, 2000] [Hornegger y Niemann, 2001] o modelos de Markov en [Li, 1995], etc.

En [Tu y Zhu, 2002], los modelos de mezclas gausianas son empleados para caracterizar colores con textura y segmentar imágenes mediante métodos Markov Chain Monte Carlo dirigidos por los datos. En [Zhang *et al.*, 2003] se emplea un método basado igualmente en el algoritmo EM que permite tanto la fusión como la división de componentes del modelo, aunque como se comentaba en el capítulo correspondiente al estado del arte, el número total de componentes (que representa el número de colores de la imagen) no cambia

durante la ejecución del algoritmo. La técnica propone emplear un criterio de inferencia bayesiana (BIC) para determinar el orden del modelo, pues el algoritmo por sí mismo no es capaz de determinarlo automáticamente. Además, es necesaria la utilización de alguna técnica adicional para inicializar los parámetros del modelo, como *k-means*. Nuestra técnica es próxima a ésta, pero empleando el algoritmo EBEM en lugar de SMEM, que como se vio en el capítulo 2 mantiene constante el número de núcleos en todo el proceso.

En el proceso de segmentación de color, cada punto de la imagen original se representa en el espacio de color RGB u otro diferente, prescindiendo por tanto de la información de posición i, j. Por tanto, se trata de un problema en un espacio de 3 dimensiones. En la figura 4.5 mostramos una representación de la imagen *Baboon* en el espacio RGB. Cada punto en la figura representa un píxel de la imagen original, que se coloca en una posición (X,Y,Z) del espacio 3D donde X es el valor de rojo, Y el valor de verde y Z el valor de azul. Los puntos que están cerca de la esquina inferior izquierda fondo representan tonos próximos al color negro, los de la derecha superior frente representan los tonos blancos y los de la diagonal principal representan los tonos grises. Para que se vea con facilidad el tono de cada punto, se han coloreado con su color original, por lo que se puede diferenciar las agrupaciones de puntos 3D que representan las diferentes clases que se obtendrán como resultado de la segmentación. La imagen original presenta claramente tonalidades rojas, azules amarillas y grises, que pueden ser observadas en la figura.

En el apartado siguiente realizamos una descripción detallada del modelo de imagen empleado para la ejecución de los experimentos de segmentación.

4.3.1. Modelo de imagen

Para comprobar el funcionamiento de nuestra técnica hemos realizado varios experimentos con imágenes en color. La representación de las imágenes se ha llevado a cabo de la siguiente forma: para cada píxel j de la imagen original hemos construido un vector de características tridimensional y_j con sus componentes en el espacio de color RGB normalizadas. Como resultado de ejecución del algoritmo, obtenemos el número de componentes desconocido a priori K y $y_j \in [1, 2, ..., K]$ para indicar a cual de las clases pertenece el píxel j. El proceso de segmentación así definido se convierte en un proceso

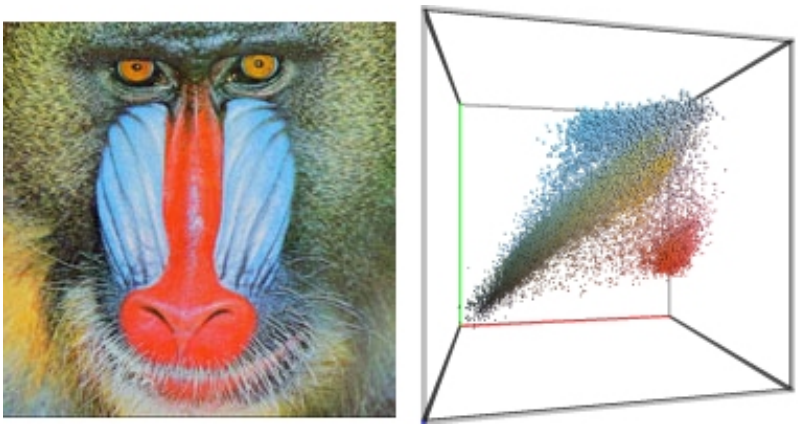

Figura 4.5: Representación gráfica de la imagen *Baboon* en el espacio de imagen (izquierda) y en el espacio RGB (derecha). Claramente se diferencian la nube de puntos roja separada del resto de puntos, así como los tonos azules y amarillos.

de etiquetado, en el que se establece que cada punto de la imagen ha sido generado por una de las distribuciones gausianas que forman parte del modelo de mezclas.

Para ello obtenemos, para cada punto de la imagen original y_j pertenecientes al conjunto inicial de observaciones y, cual de los núcleos k obtenidos por el algoritmo genera un mayor valor para $P(k|\Theta_j)$. Si aplicamos el Teorema de Bayes a la probabilidad anterior y denominamos $m(y_j)$ a la clase a la que pertenece una observación y_j, tenemos:

$$m(y_j) = \arg \max_k \{\pi_k P(y_j|\Theta_k)\} . \tag{4.3}$$

Para la obtención de los resultados se emplean diferentes umbrales de gausianidad y un umbral de convergencia para el algoritmo EM de $0{,}01$. El número de muestras empleado para la estimación de la entropía mediante el método de las ventanas de Parzen es de 1000. Dado el número de observaciones, es necesario emplear un número más elevado de muestras para la estimación precisa del valor de entropía de los núcleos que en los experimentos anteriores. El algoritmo converge tras unas pocas iteraciones (dependiendo del umbral de gausianidad especificado) generando un número creciente de componentes para la mezcla. A continuación mostramos los resultados obtenidos durante la ejecución de distintos experimentos.

4.3.2. Experimento 1

Realizamos este experimento con el objetivo de mostrar el funcionamiento del algoritmo para segmentación de color y el proceso detallado de descomposición de núcleos para una imagen en la que existe una gran variedad tonal. En la primera columna de la figura 4.6 mostramos la evolución del algoritmo para diferentes umbrales de gausianidad que dan lugar a imágenes segmentadas con número creciente de clases entre 2 y 5 representadas en cada una de las filas de la figura. En la imagen puede observarse como a medida que el umbral de gausianidad es más estricto, el núcleo con peor valor se descompone en otros dos, dando lugar a la aparición de dos nuevos colores en la imagen. Cuando un núcleo no está ajustando correctamente a los datos a los que representa, el color asociado presenta un color gris, que representa el valor promedio entre un conjunto de colores claramente diferenciados. A medida que el algoritmo evoluciona, el color gris original va dando lugar a la aparición de nuevos colores hasta que los datos quedan correctamente ajustados, o lo que es lo mismo, el umbral de gausianidad promedio de la imagen se aproxima a 0.

Cuando el número de núcleos es dos, el algoritmo ajusta correctamente el color rojo asociado a la nariz de *baboon* y el resto de tonalidades de la imagen pertenecen al otro núcleo, claramente mal ajustado y que se representa en gris. En la figura 4.6 se observa como la nube de puntos que representa los tonos rojos está claramente separada del resto de tonalidades. Tras calcular el valor de entropía y comprobar que no está ajustando correctamente al conjunto de observaciones, se descompone en otros dos, uno en tono verdoso que representa la cabeza y otro en tono gris azulado que representa a la nariz y otras tonalidades parecidas. En la cuarta secuencia de la imagen, el tono gris azulado da lugar a dos nuevos núcleos, uno claramente azul que representa correctamente la nariz y otro grisáceo que se descompone en la siguiente secuencia en el tono anaranjado que representa a los ojos y parte de la barbilla y otro grisáceo que representa al resto.

Las columnas 2 y 3 de la figura muestran una representación tridimensional en el espacio RGB de los núcleos obtenidos tras la ejecución del algoritmo. La imagen de la columna 2 muestra los núcleos resultantes. La imagen de la derecha muestra además la nube de puntos resultante coloreada con el tono

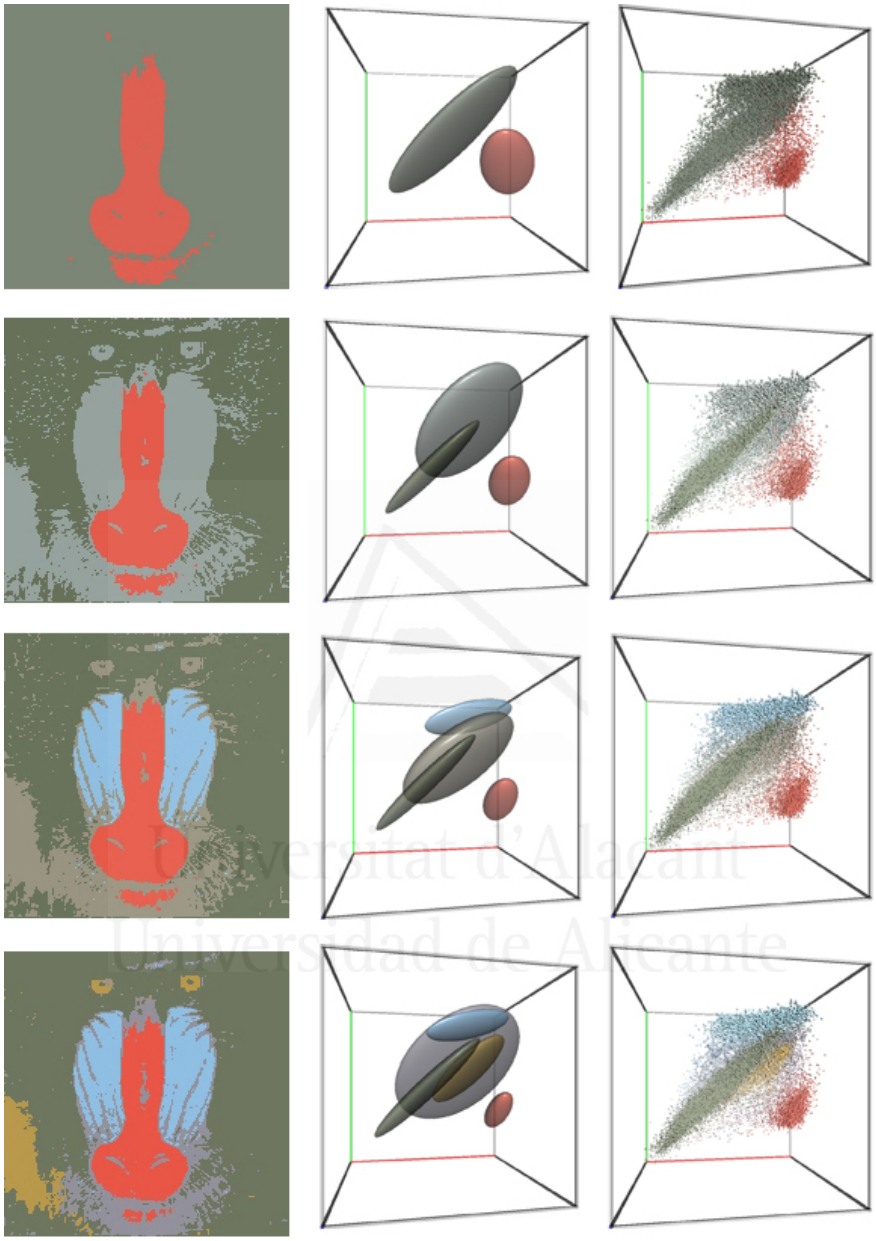

Figura 4.6: Evolución del algoritmo. La primera columna representa el resultado de la segmentación en el espacio de imagen para 2,3,4 y 5 clases (colores) respectivamente. La segunda columnas muestra la representación en el espacio RGB de los núcleos resultantes. La tercera columna representa además las observaciones de cada una de las clases resultantes coloreadas con el tono medio de la misma.

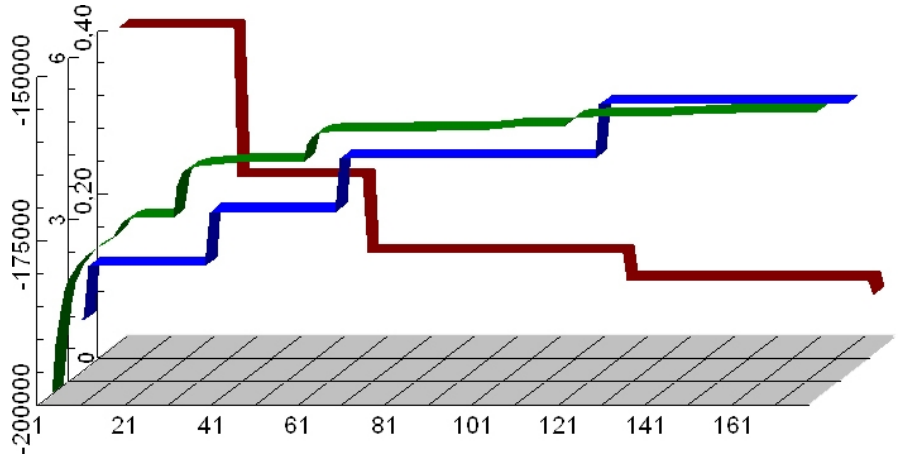

Figura 4.7: Representación gráfica de los valores de verosimilitud (verde), número de núcleos (azul) y valor de gausianidad (rojo) en cada paso del algoritmo para la imagen *Baboon*.

asociado al valor de la media del núcleo al que pertenece. Como en el caso de los experimentos 2D, el algoritmo es capaz de separar clases con un amplio grado de solapamiento.

En la figura 4.7 mostramos la evolución del experimento anterior. La serie en color verde representa el valor de la verosimilitud en cada iteración del algoritmo. Se observa claramente como ésta presenta una clara tendencia creciente. En color rojo se muestra el valor de gausianidad en cada paso del algoritmo. Puesto que el cálculo sólo se realiza tras alcanzar el umbral de verosimilitud, el valor permanece constante salvo cuando se lleva a cabo la descomposición de un núcleo en otros dos. Claramente la tendencia es descendente a medida que la introducción de nuevos núcleos permite llevar a cabo un mejor ajuste de las observaciones. La serie en color azul representa el número de núcleos de la muestra en cada iteración. Nuevamente esta permanece constante durante varias iteraciones hasta alcanzar el umbral de verosimilitud.

De la forma de la gráfica se desprende que los principales incrementos en verosimilitud coinciden con la introducción de un nuevo núcleo en la mezcla, hasta estabilizarse varias iteraciones más tarde y alcanzar la convergencia con el número de núcleos actual.

El número total de iteraciones es de 177, finalizando el algoritmo con un

Nº Iter.	K	G	L
1	1	0,394	−5,617
2	2	0,210	−4,833
31	3	0,118	−4,593
61	4	0,085	−4,516
121	5	0,068	−4,424

Tabla 4.2: Evolución de la segmentación de color de la imagen *Baboon* desde una hasta cinco clases.

valor de gausianidad para 5 clases de 0,068. En la tabla 4.2 se muestra un resumen de los parámetros del modelo para las diferentes iteraciones. La primera columna representa la iteración en la que se produce la descomposición del peor de los núcleos. La segunda columna representa el número de núcleos, la tercera el valor de Gausianidad y la última el valor de la verosimilitud de los datos normalizada con el número de píxeles de la imagen:

4.3.3. Experimento 2

En la figura 4.8 mostramos los resultados de segmentación obtenidos sobre una imagen en las clases están muy próximas, puesto que la imagen original presenta en general un aspecto *sepia* pero con una gran variedad de intensidades. Esto da lugar a que los núcleos se sitúen a lo largo de una de las diagonales del espacio RGB (fila inferior).

Para ello empleamos la conocida imagen Lenna. La imagen original se representa en la primera columna. La segunda imagen de la secuencia muestra el resultado de segmentación para dos clases. Los dos núcleos representan el promedio de tonos claros y oscuros respectivamente en la imagen. La tercera imagen representa la segmentación para tres clases. En esta ocasión el núcleo con peor valor de gausianidad es el que capturaba los tonos oscuros de la imagen, dando lugar a dos nuevos núcleos, uno claramente oscuro que captura principalmente el color del pelo y otro para el resto. La última imagen de la secuencia muestra el resultado de segmentación para cuatro clases. En este caso, el núcleo que originalmente representaba los tonos claros de la imagen se ha descompuesto en dos, uno representando los tonos más claros

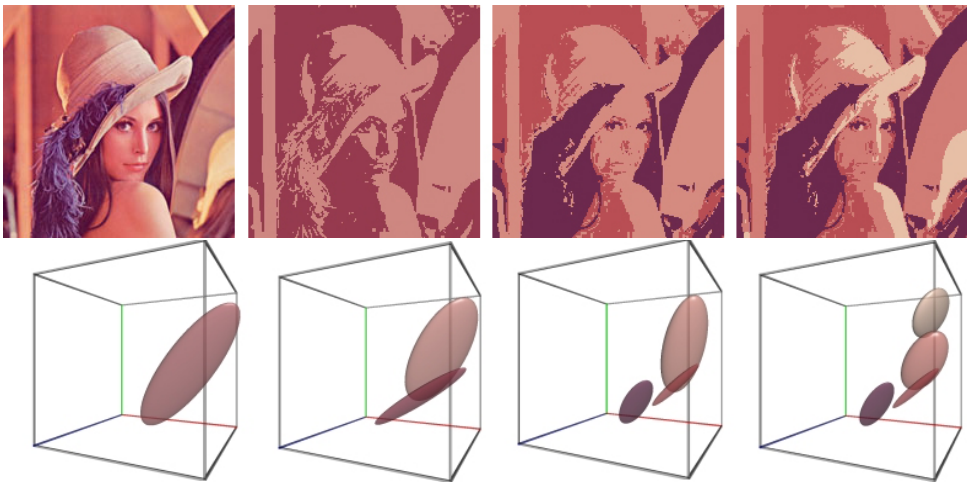

Figura 4.8: Evolución de la segmentación para la imagen Lenna en el espacio de imagen (fila superior) y en el espacio RGB (fila inferior). A pesar de que la imagen original presenta un tono *sepia*, el algoritmo separa los diferentes tonos a medida que se introducen nuevos núcleos al modelo.

de los brillos de la cara y del reflejo en el espejo y otro algo más oscuro representando al resto de tonos. El número total de iteraciones es de 230, el último valor de gausianidad calculado, para el caso de 4 es de $0{,}0955$ y los parámetros de ajuste del modelo los mismos del experimento anterior. En la tabla 4.3 se muestra un resumen de la evolución del algoritmo.

En la parte inferior de la figura se muestra la evolución del algoritmo en el espacio RGB. La primera imagen de la secuencia muestra el ajuste de la mezcla para un sólo núcleo, con una tonalidad que se corresponde con la media de la imagen. En la segunda secuencia el núcleo de tono más oscuro da

Nº Iter.	**K**	**G**	**L**
1	1	0,4239	−3,999
2	2	0,3003	−3,494
55	3	0,1543	−3,255
183	4	0,095	−3,167

Tabla 4.3: Traza de la segmentación de color para la imagen de Lenna desde una hasta cuatro clases.

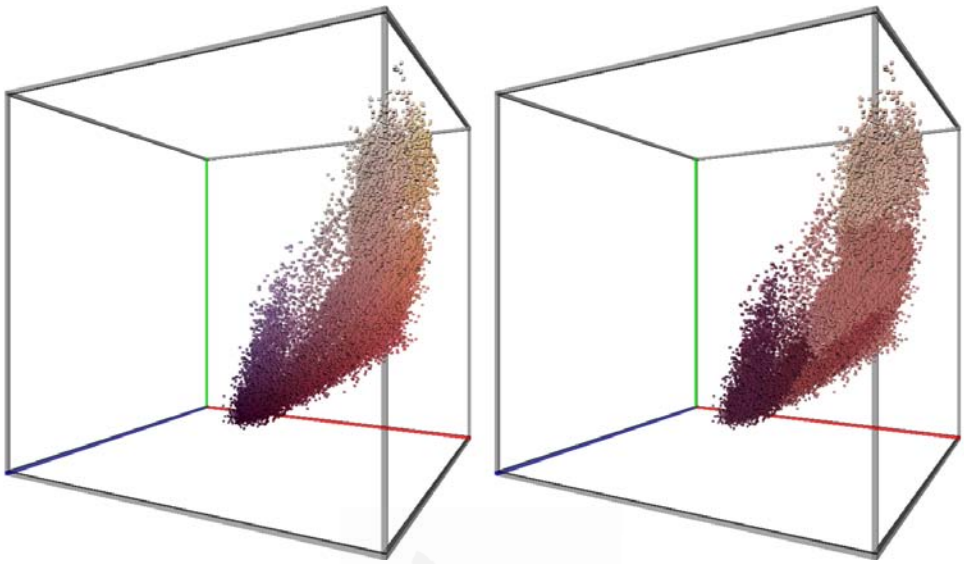

Figura 4.9: Representación en RGB de la imagen Lenna. La imagen de la izquierda muestra la nube de puntos original. La imagen de la derecha muestra la nube de puntos una vez segmentada en cuatro clases. A pesar de la proximidad de las clases el algoritmo separa correctamente cada uno de los planos de color.

lugar a otros dos, manteniéndose casi constante el núcleo de tono más claro. En la última imagen de la secuencia, este núcleo se descompone en otros dos, dando lugar a las cuatro clases resultantes de aplicar el algoritmo.

En la figura 4.9 se muestra la representación 3D de la segmentación con cuatro clases. En ambas se aprecia claramente la proximidad de las clases, que se extienden a lo largo de una de las diagonales del espacio. En la imagen de la izquierda se muestra el conjunto original de puntos; la de la derecha muestra los puntos tras la segmentación. Cada uno de ellos se ha coloreado con la tonalidad asociada al valor de la media de la clase a la que pertenece. A pesar de la proximidad de las clases, el algoritmo es capaz de separar correctamente cada una de ellas.

4.3.4. Experimento 3

Para finalizar los experimentos de segmentación de imágenes en color, en la figura 4.10 mostramos algunos resultados de para imágenes diferentes con tamaños de 189×189 píxeles, lo que supone un conjunto de $35{,}721$ ob-

Figura 4.10: Resultados en segmentación de imágenes en color. La primera columna representa las imágenes originales. El resto de columnas muestran los resultados obtenidos para diferentes umbrales de gausianidad.

servaciones. La primera columna muestra la imagen original, mientras que las columnas dos y tres muestran la imagen segmentada resultante con un umbral de gausianidad incremental.

Como en los casos de *Baboon* y *Lenna*, cuanto menor es el umbral, mayor es el número de núcleos generados por el algoritmo y por tanto, mayor el número de planos de color detectados en la imagen. Para la representación de las mismas no se ha empleado ningún tipo de post-proceso, mostrándose todas las imágenes tal cual han sido generadas por el algoritmo. Cada punto de la imagen original ha sido etiquetado con el color representado por la media de la clase a la que pertenece en el espacio RGB, es decir, los valores de rojo, verde y azul, resultando el color *promedio* de la totalidad de puntos de la imagen a los que representa.

Cada fila muestra la imagen original y dos resultados de segmentación con umbral de gausianidad decreciente. Durante la ejecución de las pruebas se ha empleado el método de estimación de entropía basado en ventanas de Parzen con una selección de 1000 observaciones para determinar el valor de entropía de cada núcleo y el umbral de verosimilitud se ha fijado en $0{,}01$. Las imágenes pertenecen a la base de datos de la Universidad de Berkeley para segmentación de imágenes y detección de bordes [Martin *et al.*, 2001]

4.4. Criterio de parada basado en longitud mínima

A continuación se muestran los resultados obtenidos con nuestra técnica empleando como criterio de selección del orden del modelo los principios de longitud de descripción mínima y mensaje mínimo introducidos en el tema anterior.

4.4.1. Mezcla artificial de cuatro clases solapadas

En el primero de los experimentos del apartado realizamos una comparación de los resultados obtenidos en el problema de estimación de densidad de probabilidad para cada uno de los tres criterios para la mezcla de cuatro núcleos con núcleos solapados. La imagen de la figura 4.11 muestra la evolución de la función de energía para diferente número de núcleos. La línea vertical discontinua muestra el número óptimo de núcleos (cuatro en esta ocasión).

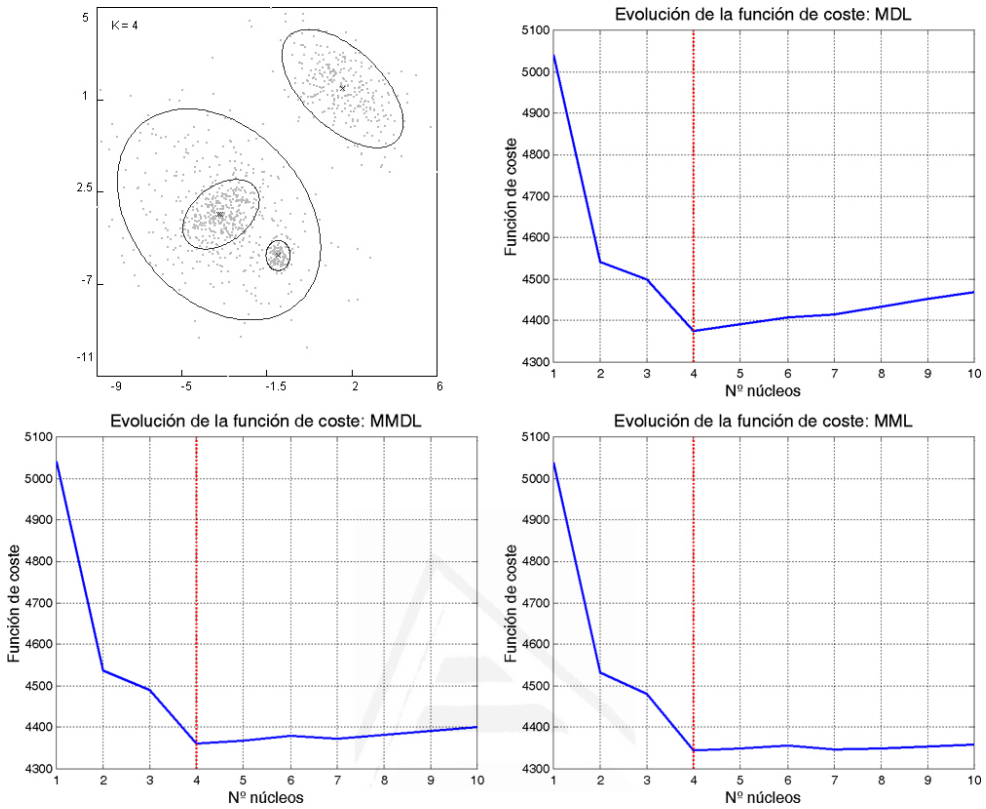

Figura 4.11: Representación de la evolución de la función de coste para la mezcla de 4 núcleos solapados (arriba izquierda). El criterio basado en MDL (arriba derecha) muestra un crecimiento mayor de la función de energía cuando se sobrepasa el número óptimo de núcleos que MMDL (abajo izquierda) y MML (abajo derecha).

Los tres criterios muestran un comportamiento similar mientras el número de núcleos es inferior al óptimo, con un pronunciado descenso. Sin embargo, una vez sobrepasado el valor óptimo este comportamiento cambia: El criterio MDL estándar muestra un ascenso monótono, mientras que MMDL y MML presentan un mínimo local para $k = 6$ y un crecimiento menos pronunciado. Este efecto es todavía más apreciable en el caso de MML.

En cualquier caso, puesto que el algoritmo comienza con un sólo núcleo y en los 3 criterios el descenso hasta el óptimo es monótono, los resultados obtenidos son idénticos, fijándose en todos ellos $K = 4$.

4.4.2. Mezcla artificial de cinco clases distanciadas

En la figura 4.12 mostramos la evolución de las 3 funciones de coste para un ejemplo de mezcla en el que existe gran distancia entre la mayor parte de las clases, salvo en dos de ellas que se encuentran relativamente próximas. En este caso, nuevamente existe un descenso pronunciado para un orden de modelo entre 1 y 5, para comenzar luego un ligero ascenso. Exceptuando el criterio MDL clásico, en los otros dos aparecen igualmente mínimos locales, llegando incluso el criterio MML a generar un mínimo absoluto de la función para $k = 10$. No obstante, el hecho de comenzar con un sólo núcleo, permite utilizar igualmente los 3 criterios obteniendo el mismo resultado para el orden del modelo: $K = 5$.

Tal y como apuntan otros autores [Kontkanen *et al.*, 1996] [Smyth, 1996], el criterio MDL penaliza en mayor medida que MMDL o MML la aparición de nuevos componentes en la mezcla, resultando valores de la función criterio mayores cuando el número de núcleos crece. No obstante, puesto que el algoritmo propuesto recorre la función criterio por la izquierda del óptimo y como en el caso anterior los 3 criterios presentan una tendencia claramente descendente, todos ellos pueden ser empleados como criterio para la selección del orden del modelo. Además, el coste espacial se reduce al almacenamiento únicamente de los dos últimos modelos de mezcla generados: el actual con k núcleos y el anterior, con $k - 1$ núcleos y que será el orden correcto del modelo cuando crezca el valor de la función.

4.4.3. Segmentación de imágenes en color

Por último, hemos comprobado el funcionamiento de los criterios basados en longitud mínima para el problema de la segmentación de imágenes en color. En las pruebas realizadas, ninguno de los tres criterios empleados introduce la suficiente penalización como para que el valor de la función de coste comience a incrementarse. Como en experimentos anteriores, se produce un rápido descenso inicial, pero a partir de ese momento el descenso es progresivo sin llegar a alcanzar un mínimo absoluto en ejecuciones de hasta 30 clases. La figura 4.13 mostramos la evolución de la función para el ejemplo de la imagen de las cebras, así como la imagen segmentada resultante con 30 clases.

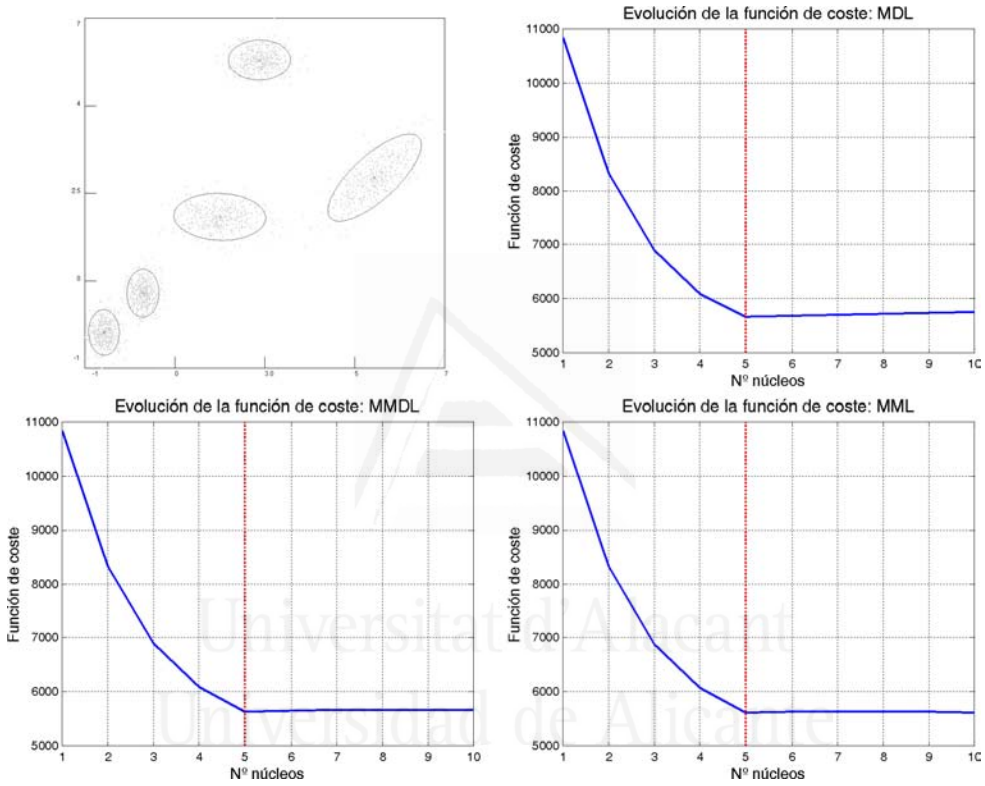

Figura 4.12: Representación de la evolución de la función de coste para una mezcla de 5 núcleos distanciados (arriba izquierda). El criterio basado en MML (abajo derecha) presenta un mínimo absoluto en $k = 10$, aunque puesto que el primer mínimo local se produce para $k = 5$, el resultado obtenido es el mismo que para MDL clásico (arriba derecha) y MMDL (abajo izquierda).

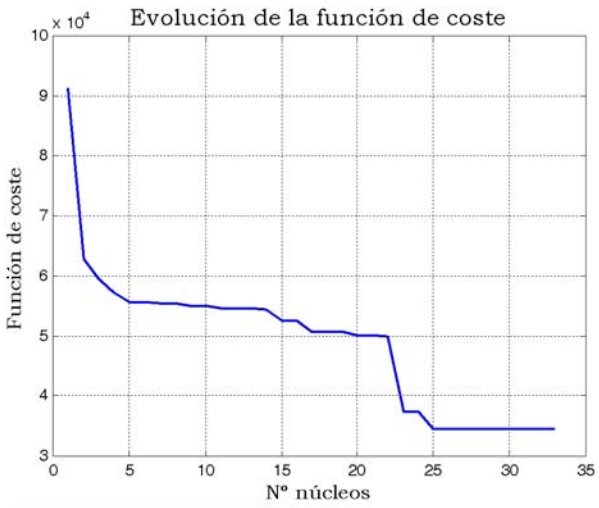

Figura 4.13: Segmentación de 30 clases de la imagen de las cebras (izquierda). Evolución de la función de coste con criterio MDL (el más restrictivo) (derecha). A pesar del elevado número de núcleos la función no inicia el ascenso.

La explicación a esta situación debemos encontrarla en que estamos representando con una mezcla gausiana un conjunto de datos cuya distribución no lo es. Por ello, a medida que se incrementa el número de componentes de la mezcla, el incremento de la verosimilitud tiene mayor peso que la componente de penalización introducida en la función criterio y la función no alcanza el mínimo a pesar de que el número de componentes es muy elevado.

Resultados similares se obtienen en [Liang *et al.*, 1992] en el contexto de la segmentación de imágenes radiológicas en gris 1-D. Aunque asintóticamente para valores elevados de n la longitud óptima de código para cada parámetro real es $1/2\log(n)$ [Schwarz, 1978], los autores comprueban experimentalmente que se obtienen mejores resultados para su problema en particular empleando $5/2$ como factor. En nuestro caso, en el que la dimensión es superior y los datos tienen una mayor variabilidad, ese factor tampoco es suficiente para que la función deje de decrecer con un número reducido de núcleos, obteniendo resultados más satisfactorios con un factor de 10.

La imagen de la figura 4.14 muestra la evolución de la función de coste para el criterio MDL original, con factor de $5/2$ y factor de 10 para el experimento realizado con la imagen de las cebras. Sólo en el caso del factor de

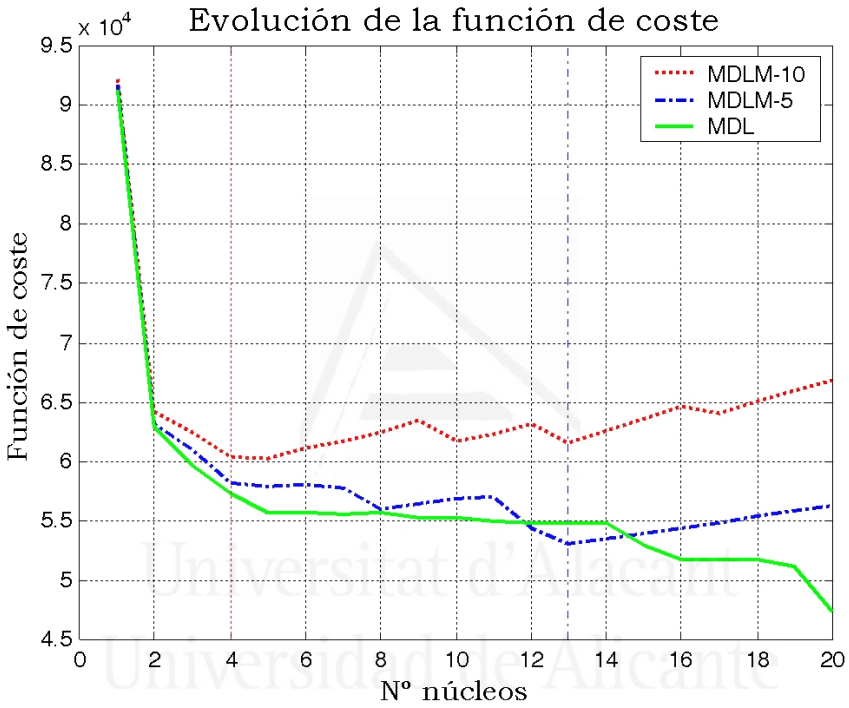

Figura 4.14: Representación de la evolución de la función criterio MDL estándar (verde), MDL factor $5/2$ (azul) y MDL factor 10 (rojo). Dependiendo del factor de penalización incluido el número óptimo de núcleos es diferente. Incluso éste no se alcanza para el factor $1/2$ de MDL estándar.

penalización de $5/2$ y 10 se produce un incremento de la función criterio tras alcanzar el mínimo. Este valor se obtiene para 13 y 4 núcleos respectivamente. Para la solución de cuatro núcleos, la segmentación resultante emplea dos variedades de gris para representar las cebras y dos de verde para el fondo, incrementándose progresivamente el número de tonalidades para la solución de 13 núcleos y para la de 20. Desde un punto de vista subjetivo, una solución de 4 núcleos sería suficiente para captar la mayor parte de la información de la escena, por lo que entendemos que es precisamente ese factor el que mejores resultados ofrece para nuestro problema en particular.

En cualquier caso, determinar el factor asociado al término que representa el peso en los criterios de longitud mínima es similar a especificar un umbral y depende de los datos. Por tanto, para la segmentación de imágenes a color, preferimos emplear como criterio de parada del algoritmo el grado de gausianidad en lugar de cualquiera de las funciones criterio descritas en el capítulo anterior. De este modo, se puede obtener un número de núcleos variable dependiendo del grado de gausianidad exigido al modelo y de la propia naturaleza de la imagen. Por supuesto, también sería posible especificar un número de núcleos a priori y detener el algoritmo cuando se alcance dicho número, obteniendo la solución óptima para el número de elementos fijado gracias a la forma en la que se introducen nuevos componentes a la mezcla.

Capítulo 5

Conclusiones y desarrollos futuros

Como finalización del trabajo llevamos a cabo una revisión de las principales conclusiones obtenidas tras los experimentos realizados así como las ampliaciones y los aspectos mejorables que forman parte de nuestro trabajo futuro.

5.1. Conclusiones

En este trabajo hemos presentado un nuevo algoritmo basado en EM para el ajuste de un modelo de mezcla gausiana a un conjunto de datos. La técnica permite además seleccionar el número óptimo de componentes del modelo empleando para ello dos criterios, uno basado en entropía y otro basado en criterios de información como el principio de *Longitud de Descripción Mínima* y sus variantes. El proceso comienza con un sólo núcleo y tras lograr la convergencia del algoritmo EM, se aplica un criterio basado en la entropía asociada a la densidad de probabilidad de cada uno de los núcleos que la forman, que permite decidir cual de ellos debe descomponerse en otros dos. La entropía se ha mostrado como una medida muy adecuada para estimar el

grado de normalidad o gausianidad asociado a un conjunto de datos. Según el segundo teorema de Gibbs, una distribución gausiana maximiza la entropía sobre todas las distribuciones no gausianas de igual varianza. Por otra parte, para una distribución normal podemos calcular de forma analítica el valor de la entropía. Por ello podemos averiguar el grado de normalidad de una distribución, comparando la medida real obtenida a partir de los datos con la máxima teórica, que se obtendría en el caso de que los datos hubieran sido generados por una distribución normal. Puesto que pretendemos ajustar los datos con un conjunto de núcleos gausianos, podemos comenzar con un conjunto reducido de ellos (normalmente uno) y emplear la medida anterior para determinar las zonas en las que aparece bimodalidad en oposición a normalidad, e incluir un nuevo núcleo que ajuste en mejor medida los datos de su vecindad.

El método converge en unas pocas iteraciones y se ha comprobado que es adecuado para estimación de densidad de probabilidad, tanto para distribuciones que se encuentran muy alejadas en el espacio como para las que presentan un importante grado de solapamiento, así como para reconocimiento de patrones y segmentación no supervisada de imágenes en color.

Puesto que el algoritmo comienza con un sólo núcleo, cuyos valores iniciales de media y covarianza vienen dados por el conjunto inicial de datos en su totalidad, no es sensible a la inicialización, evitando la posibilidad de convergencia a un máximo local del algoritmo EM clásico. Además, el criterio de introducción de nuevos núcleos de forma dinámica, en las zonas en las que los datos están peor ajustados, elimina otro de los problemas clásicos del algoritmo EM, derivado de la imposibilidad de que un núcleo se desplace por el espacio de observaciones cuando los datos están muy separados. La inicialización adecuada de los valores de media y covarianza de los nuevos núcleos, desplazándose en las direcciones de máxima variabilidad de los datos, permite obtener una rápida convergencia en los pasos EM ejecutados a continuación de la división.

Proponemos la utilización de dos técnicas diferentes para la estimación de la entropía: el método denominado *plug-in*, basado en Ventanas de Parzen emplea parte de los datos para estimar la densidad de probabilidad y el resto para estimar la entropía. Este método es adecuado para problemas de baja dimensionalidad con gran cantidad de datos y muestra una comporta-

miento robusto ante la presencia de falsos positivos. La segunda técnica se basa en las propiedades asintóticas de los *Entropic Spanning Graphs* que permiten realizar una estimación de la entropía de Renyi. El método no permite la estimación directa de la entropía de Shannon, por lo que hemos desarrollado una técnica que permite estimar esta última a partir de la primera. A este método de estimación de entropía se le denomina *non plug-in*, pues no requiere una estimación previa de la densidad de probabilidad. El método es adecuado para situaciones en las que el conjunto de observaciones es limitado, o bien perteneciente a un espacio de dimensionalidad elevado. En el primero de los casos, la escasez de observaciones no permite estimar de forma precisa la densidad de probabilidad, mientras que en el segundo, el curso de la dimensionalidad puede hacer igualmente insuficiente el número de observaciones para estimar dicha densidad. Por el contrario, la necesidad de calcular el *Minimal Spanning Tree* asociado al conjunto de datos, hace a esta técnica menos robusta ante la presencia de outliers. Este último problema podría solventarse empleando la variante K-MST del algoritmo que permite rechazar los puntos no pertenecientes a la distribución durante el proceso de construcción del árbol.

Como criterio de parada del algoritmo empleamos proponemos dos técnicas. La primera de ellas basada en la diferencia en promedio de las entropías reales y teóricas de los componentes de la mezcla a la que denominamos umbral de gausianidad. Esta medida está normalizada con valores entre cero (máxima gausianidad) y uno (ausencia absoluta).

El criterio de gausianidad es más versátil que fijar de antemano el número de clases. Por ejemplo, en un contexto de segmentación de imagen, se podría asumir que en una secuencia de imágenes del mismo entorno, el umbral de gausianidad debería permanecer casi constante, mientras que el número de núcleos podría ser diferente en cada secuencia. Aunque se podría usar el número de colores, la introducción de nuevos componentes de la técnica propuesta se realiza dinámicamente, evitando caer en mínimos locales.

El segundo criterio de parada está basado en el principio de longitud mínima, tanto de descripción (MDL) como de mensaje (MML), empleado tradicionalmente a lo largo de la literatura como criterio de selección del orden del modelo. A partir de una función de energía que penaliza la inclusión de un número excesivo de componentes a la mezcla es posible detener el proceso

de descomposición de núcleos en $K+1$ iteraciones, con K el número óptimo de núcleos. Ambos planteamientos muestran un descenso pronunciado de la función de energía en el intervalo comprendido entre el núcleo inicial y el número óptimo, para comenzar posteriormente un ascenso progresivo. Esta característica permite detener el proceso sin el inconveniente de recorrer todo el conjunto de valores posibles de k, que requieren otras técnicas anteriores. El criterio se ha mostrado adecuado para problemas de estimación de densidad de probabilidad asociada a un conjunto de datos, pero no así para la segmentación de imágenes en color, en la que las versiones estándar de ambos criterios no penalizan lo suficiente la función de energía y el incremento en verosimilitud tiene mayor peso que que la penalización por introducción de nuevos componentes. Debido a esto, la función no alcanza un mínimo en un número de iteraciones razonable. Variaciones del peso asociado al término de penalización proporcional a $\log n$ permiten alcanzar el mínimo, pero creemos que para este tipo de problemas es más adecuado emplear el grado de gausianidad como criterio de parada, puesto que los datos a ajustar no son verdaderamente gausianos. En este sentido, el umbral de gausianidad es, en cierto modo, el nivel de semejanza de los datos con unos verdaderamente gausianos.

5.2. Desarrollos futuros

Si bien el algoritmo se ha comportado de forma robusta en los experimentos realizados, hay algunos aspectos que podrían ser mejorados y que planteamos como desarrollos futuros.

En primer lugar, la estimación de la entropía por la técnica del MST presenta una sensibilidad a la presencia de falos positivos que provoca una estimación menos precisa de la entropía en situaciones de escasa gausianidad. Este problema puede ser solucionado con la implementación de la variante K-MST [Hero y Michel, 1999a] que permite la eliminación de los puntos no pertenecientes a la distribución.

En cuanto a la estimación de la entropía por el método de las Ventanas de Parzen, el ajuste del ancho óptimo requiere un descenso por gradiente para determinar el ancho más adecuado. Este proceso es costoso computacional-

mente y podría acelerarse si se encontrara una forma distinta de establecer este ancho que no precise un procedimiento iterativo y permitiera igualmente una estimación precisa de la entropía.

Los criterios de parada del algoritmo basados en Longitud Mínima se comportan adecuadamente para problemas de estimación de densidad de probabilidad o clasificación, pero no para segmentación de imágenes en color. Sería interesante derivar una modificación del principio MDL o MML que fuera adecuado para este tipo de problemas y presentara un mínimo claro de la función criterio para un número de núcleos no excesivamente elevado.

Por último, aunque el algoritmo es capaz de determinar el orden correcto del modelo y ajustar adecuadamente los parámetros para un problema determinado, lo hace para una dimensionalidad especificada previamente. Cuando el número de dimensiones es elevado, la complejidad del algoritmo aumenta, siendo más costoso tanto estimar la entropía como ajustar los parámetros del modelo. Una mejora interesante sería tratar de estimar además el conjunto de dimensiones que permite realizar un ajuste óptimo. La mayor parte de los algoritmos de selección de características se emplean en problemas de clasificación supervisada, en los que existe información acerca de las clases existentes. Para el caso no supervisado, han sido utilizados métodos basados en reducción de dimensionalidad o extracción de características, como PCA (*Principal Component Analysis*, Transformada de Karhunen-Loeve o SVD (*Singular Value Decomposition*) [Duda y Hart, 1973].

Más recientemente han aparecido propuestas basadas en la selección de las características más importantes a partir de algún criterio, como en [Cheng *et al.*, 1999] o [Dash y Liu, 2000] que llevan a cabo una ordenación de las características a partir de un criterio basado en entropía y la selección del mejor subconjunto a partir de una función criterio. Si consideramos cada característica como una variable aleatoria, su entropía será máxima cuando la densidad de probabilidad sea uniforme. En ese caso además, dicha característica será de escaso o nulo valor a la hora de clasificar los datos. La idea es sencilla, si denominamos C al conjunto completo de características y c_1 y c_2 a dos de las características pertenecientes al conjunto anterior, podemos comparar la entropía de $C - c_1$ con la entropía de $C - c_2$, es decir, eliminamos alternativamente del conjunto c_1 y c_2. Si la entropía resultante de eliminar c_1 es mayor que la obtenida tras eliminar c_2, entonces c_1 es más importante que

c_2 pues si la eliminamos los datos se distribuyen de manera más uniforme. Por tanto, disponemos de un criterio para ordenar las diferentes características según su importancia.

El problema radica en la introducción de este criterio en algoritmo desarrollado, pues aunque siguiendo con la filosofía del mismo, se podría plantear un proceso comenzando con el menor número posible de características, la comparación entre las soluciones obtenidas para diferentes dimensiones no es un problema trivial y requiere un importante trabajo de investigación.

Apéndice A

Producción científica

En este apartado resumiremos la producción científica de esta tesis. Distinguimos entre publicaciones internacionales y proyectos.

A.1. Publicaciones internacionales

A. Peñalver, J.M. Sáez, F. Escolano, *An Entropy Maximization Approach to Optimal Model Selection in Gaussian Mixtures*, In: Sanfeliu A., Ruiz-Shulcloper J. (eds.): Progress in Pattern Recognition, Speech and Image Analysis, 8th Iberoamerican Congress on Pattern Recognition (CIARP'03), Havana, Cuba, November 2003. Lecture Notes in Computer Science, Vol 2905, Springer-Verlag, Berlin Heidelberg New York (2003). 432-439.

En este artículo se propone el criterio basado en entropía para determinar la calidad del ajuste de una mezcla con un determinado número de núcleos. Si esta medida queda por debajo de un umbral, denominado de gausianidad, el núcleo con peor valor de la misma es descompuesto en otros dos. Para la determinación de los parámetros de los nuevos núcleos introducidos se emplea una técnica heurística. La estimación de la entropía se realiza mediante un método *plug-in* basado en ventanas de Parzen. Los resultados de la técnica propuesta son comparados con los obtenidos con el algoritmo EM clásico,

para el caso de estimación de densidad de probabilidad asociada a los datos y reconocimiento de patrones.

J.M. Sáez, A. Peñalver, F. Escolano, *Compact Mapping in Plane-Parallel Environments Using Stereo Vision*, In: Sanfeliu A., Ruiz-Shulcloper J. (eds.): Progress in Pattern Recognition, Speech and Image Analysis, 8th Iberoamerican Congress on Pattern Recognition (CIARP'03), Havana, Cuba, November 2003. Lecture Notes in Computer Science, Vol 2905, Springer-Verlag, Berlin Heidelberg New York (2003). 659-666.

En este artículo se utiliza una implementación básica de los modelos de mezcla gausiana en 1-D para estimar los principales planos 3D del entorno (paredes, suelo y techo) en tareas de navegación de robots. Asumiendo que el robot se mueve en un entorno plano-paralelo, a partir de la orientación de las normales y la posición relativa de los puntos captados con una cámara estéreo, se realiza una clasificación de los mismos como perteneciente a una de las clases anteriores. Posteriormente, se ajusta un plano 3D a cada grupo y se mapea la textura a partir de la información de apariencia.

A. Peñalver, F. Escolano, J.M. Sáez, *Color Image Segmentation Through Unsupervised Gaussian Mixture Models*, In: J.S. Sichman et al. (Eds.): Progress in Pattern Recognition, Speech and Image Analysis, 10th Ibero-American Artificial Intelligence Conference (IBERAMIA'06), Riberao Preto, Brazil, October 2006. Lecture Notes in Artificial Intelligence, Vol 4140, Springer-Verlag, Berlin Heidelberg New York (2006). 149-158.

En esta publicación se presenta la aplicación del algoritmo EBEM a la segmentación de imágenes en color. Para este tipo de problemas, el umbral de gausianidad puede ser utilizado para determinar la cantidad de clases o colores diferentes que aparecen en una imagen. Además, la técnica heurística inicial para la determinación de los valores de los nuevos parámetros tras descomponer un núcleo de la muestra es reemplazada por otra basada en la descomposición de las matrices de covarianza de cada núcleo, con la restricción de que los dos primeros momentos estadísticos asociados a cada núcleo se mantengan antes y después del proceso de división.

A. Peñalver, F.Escolano, J.M. Sáez, *EBEM: an Entropy-based EM Algorithm for Gaussian Mixtures*, Proceedings on IEEE International Conference on Pattern Recognition (ICPR 2006). Hong Kong, China, August 2006.

En esta publicación se presenta una nueva técnica para la estimación de la entropía catalogada dentro de los métodos *non plug-in*. A partir de la estimación del *Minimal Spanning Tree* asociado al *Entropic Spanning Graph* del conjunto de observaciones, se obtiene una estimación de la entropía de Renyi de orden α. La técnica original no permite la estimación directa de la entropía de Shannon ($\alpha = 1$) a partir de este valor, por lo que se desarrolla un nuevo método que, estudiando el comportamiento de la función en el límite (cuando $\alpha \rightarrow 1^{-}$), permite calcular el valor de la entropía de Shannon.

A. Peñalver, F. Escolano, J.M. Sáez, *Two Entropy-based Methods for Learning Unsupervised Gaussian Mixture Models*, In: D.-Y. Yeung et al. (Eds.): Progress in Pattern Recognition, Speech and Image Analysis, 6th International Workshop on Statistical Pattern Recognition (SPR-SSPR'06), Hong Kong, China, August 2006. Lecture Notes in Computer Science, Vol 4109, Springer-Verlag, Berlin Heidelberg New York (2006). 649-657.

En esta publicación se realiza una comparación del funcionamiento del algoritmo propuesto EBEM para las dos técnicas de estimación de entropía expuestas anteriormente: método *plug-in* basado en ventanas de Parzen y non plug-in basado en la estimación del *minimal spanning tree* asociado a los datos. Se realizan experimentos de estimación de densidad de probabilidad, reconocimiento de imágenes y segmentación de imágenes en color. De los experimentos realizados se concluye en que casos es más recomendable emplear una u otra técnica.

A.2. Proyectos

Grupo de investigación en Visión y Robótica (RVG), *Mapeado con Robots Móviles usando Técnicas de Visión Activa (MAP3D)*. Financiación: Ministerio de Ciencia y Tecnología (MCYT TIC 2002-62792). Investigador Principal: F. Escolano Ruiz. 01/12/2002-01/12/2005.

Los resultados del trabajo presentado en esta tesis doctoral se han empleado en el desarrollo de este proyecto de subvención pública, llevado a cabo por el Grupo de investigación en Visión y Robótica de la Universidad de Alicante. Cabe destacar que en esta tesis únicamente presentamos la investigación realizada por el autor de la misma en el ámbito del proyecto. El proyecto en sí es mucho más amplio, y ha sido desarrollado por diez investigadores a tiempo completo. En [Peñalver *et al.*, 2003] se aplican los resultados iniciales de selección del número óptimo de núcleos en la mezcla para problemas de clasificación no-supervisada. Posteriormente, en [Peñalver *et al.*, 2006a], [Peñalver *et al.*, 2006b] y [Peñalver *et al.*, 2006c] se aplican los resultados del algoritmo de clustering modificado para segmentación basada en color. El algoritmo se basa en modelos de mezclas gaussianas determinando de forma automática el número óptimo de clusters mediante aplicación de la teoría de la información y se realizan los primeros experimentos para la estimación de la pose del robot.

Bibliografía

[Agrawal *et al.*, 1998] R. Agrawal, J. Gehrke, D. Gunopulos, y P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Inter. Conf. Management of Data. ACM-SIGMOD*, Seattle, Washington, 1998.

[Akaike, 1973] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281, Budapest, 1973.

[Banfield y Raftery, 1993] J. Banfield y A. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49:803–821, 1993.

[Banks *et al.*, 1992] D. Banks, M. Lavine, y H.J. Newton. The minnimal spanning tree for nonparametric regresion and structure discovery. In *Computing Science and Statistics. 24 Th Symposium on the Interface*, pages 370–374, 1992.

[Beirlant *et al.*, 1996] E. Beirlant, E. Dudewicz, L. Gyorfi, y E. Van der Meulen. Nonparametric entropy estimation. *International Journal on Mathematical and Statistical Sciences*, 6(1):17–39, 1996.

[Bernardo y Smith, 1994] J. Bernardo y A. Smith. *Bayesian Theory*. J. Wiley and Sons, Chichester, UK, 1994.

[Bertsekas, 1999] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Mass., 1999.

[Bertsimas y Ryzin, 1990] D.J. Bertsimas y G. Van Ryzin. An asymptotic determination of the minimum spanning tree and minimum matching cons-

tants in geometrical probability. *Operations Research Letters*, 9(1):223–231, 1990.

[Biernacki *et al.*, 1999] C. Biernacki, G. Celeux, y G. Govaert. An improvement of the nec criterion for assessing the number of clusters in a mixture model. *Pattern Recognition Letters*, 20:267–272, 1999.

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

[Biernacki y Govaert, 1997] C. Biernacki y G. Govaert. Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics*, 29:451–457, 1997.

[Bishop, 1994] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1994.

[Blake y Merz, 1998] C.L Blake y C.J. Merz. Uci repository of machine learning databases. *University of California, Irvine, Dept. of Information and Computer Sciences*, 1998.

[Bottolo *et al.*, 2003] L. Bottolo, G. Dellaportas, y A. Lijoi. Bayesian analysis of extreme values by mixture modelling. *Extremes*, 6:25–47, 2003.

[Bozdogan, 1993] H. Bozdogan. Choosing the number of component clusters in the mixture model using a new informational complexity criterion of the inverse-fisher information matrix. *Information and Classification, Springer Verlag*, pages 40–54, 1993.

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

[Celeux *et al.*, 1999] G. Celeux, S. Chretien, F. Forbes, y A. Mikhadri. A component-wise em algorithm for mixtures. Technical Report 3746, INRIA Rhone-Alpes, France, 1999.

[Celeux y Soromenho, 1996] G. Celeux y G. Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Classification Journal*, 13:195–212, 1996.

[Cheng *et al.*, 1999] C. H. Cheng, A. W. Fu, y Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Knowledge Discovery and Data Mining*, pages 84–93, 1999.

[Chrétien y Hero, 2000] S. Chrétien y A. Hero. Kullback proximal algorithms for maximum likelihood estimation. *IEEE Transactions on Information Theory*, 46, 2000.

[Comon, 1994] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[Conway y Sloane, 1993] J. Conway y N. Sloane. *Sphere Packings, Lattices and Groups*. Springer Verlag, New York, 1993.

[Cover y Thomas, 1991] T. Cover y J. Thomas. *Elements of Information Theory*. J. Wiley and Sons, 1991.

[Dalal y Hall, 1983] S. Dalal y W. Hall. Approximating priors by mixtures of natural conjugate priors. *Journal of The Royal Statistical Society(B)*, 45(1), 1983.

[Dasgupta y Raftery, 1998] A. Dasgupta y A. Raftery. Detecting features in spatial point patterns with clutter via model-based clustering. *J. Am. Statistical Assoc.*, 93:294–302, 1998.

[Dash y Liu, 2000] M. Dash y H. Liu. Feature selection for clustering. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 110–121, 2000.

[Day, 1969] N. E. Day. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474, 1969.

[Dellaportas y Papageorgiou, 2006] P. Dellaportas y I. Papageorgiou. Multivariate mixtures of normals with unknown number of components. *Statistics and Computing*, 16(1):57–68, 2006.

[Dempster *et al.*, 1977] A. Dempster, N. Laird, y D. Rubin. Maximum likelihood estimation from incomplete data via the em algorithm. *Journal of The Royal Statistical Society*, 39(1):1–38, 1977.

[Devijver y Kittler, 1982] P.A. Devijver y J. Kittler. *Pattern Recognition: a Statistical Approach*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[Donoho, 1993] D.L. Donoho. One-sided inference about functionals of a density. *Ann. Statist.*, 16:1390–1420, 1993.

[Duda y Hart, 1973] R.O. Duda y P.E. Hart. *Pattern Classification and Scene Analysis. Chapter: Unsupervised Learning and Clustering*. John Willey and Sons, 1973.

[Erdogmus *et al.*, 2004] D. Erdogmus, K.E. Hild, J.C. Principe, M. Lazaro, y I. Santamaria. Adaptive blind deconvolution of linear channels using renyi's entropy with parzen window estimation. *IEEE Transactions on Signal Processing*, 52(6):1489–1498, 2004.

[Faires y Burden, 2004] J.D. Faires y R. Burden. *Numerical Methods*. Thomson Eds., 2004.

[Fernandez y Green, 2002] C. Fernandez y P.J. Green. Modelling spatially correlated data via mixtures: a bayesian approach. *Journal of the Royal Statistical Society B*, 64:805–826, 2002.

[Figueiredo *et al.*, 1999] M.A.T Figueiredo, J.M.N Leitao, y A.K. Jain. On fitting mixture models. *Energy Minimization Methods in Computer Vision and Pattern Recognition. Lecture Notes in Computer Science*, 1654(1):54–69, 1999.

[Figueiredo y Jain, 2000] M.A.T Figueiredo y A.K. Jain. Unsupervised selection and estimation of finite mixture models. In *International Conference on Pattern Recognition. ICPR2000*, Barcelona, Spain, 2000. IEEE.

[Figueiredo y Jain, 2002] M.A.T Figueiredo y A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–399, 2002.

[Fraley y Raftery, 1997] C. Fraley y A. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. Technical Report 329, Dept. Statistics, Washington University, Seattle. WA, 1997.

[Girolami y Fyfe, 1997] M. Girolami y C. Fyfe. Negentropy and kurtosis as projection pursuit indices provide generalised ica algorithms. Technical report, University of Paisley, Scotland, 1997.

[Goldberger *et al.*, 2006] J. Goldberger, S. Gordon, y H. Greenspan. Unsupervised image-set clustering using an information theoretic framework. *IEEE Transactions on Image Processing*, 15(2):449–458, 2006.

[Golub y Lan, 1996] G. H. Golub y C.F.V. Lan. *Matrix Computations, 3rd Edition*. The Johns Hopkins University Press, Baltimore, 1996.

[Green y Richardson, 2001] P. J. Green y S. Richardson. Modeling heterogeneity with and without the dirichlet process. *Scandinavian Journal of Statistics*, 28:355–376, 2001.

[Green, 1995] P. J. Green. Reversible jump markov chain monte carlo computaton and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[Hall y Morton, 1993] P. Hall y S.C. Morton. On the estimation of entropy. *Ann. Inst. Statist. Math.*, 45:69–88, 1993.

[Hastie y Tibshirani, 1996] T. Hastie y R. Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of The Royal Statistical Society(B)*, 58(1):155–176, 1996.

[Hero *et al.*,] A. O. Hero, J. A. Costa, y B. Ma. Convergence rates of minimal graphs with random vertices. Submited for publication. Available: http://citeseer.ist.psu.edu/hero03convergence.html.

[Hero y Michel, 1998] A.O. Hero y O. Michel. Asymptotic theory of greedy aproximations to minimal k-point random graphs. Technical Report 315, Communications and Signal Processing Laboratories (CCSPL), Dept. EECS. The University of Michigan. Ann Arbor, MI, 48109-2122 U.S.A., 1998.

[Hero y Michel, 1999a] A.O. Hero y O. Michel. Asymptotic theory of greedy aproximations to minimal k-point random graphs. *IEEE Transactions on Information Theory*, 45(6):1921–1939, 1999.

[Hero y Michel, 1999b] A.O. Hero y O. Michel. Estimation of rényi information divergence via pruned minimal spanning trees. In *Workshop on Higher Order Statistics*, Caessaria, Israel, 1999. IEEE.

[Hero y Michel, 2002] A.O. Hero y O. Michel. Applications of spanning entropic graphs. *IEEE Signal Processing Magazine*, 19(5):85–95, 2002.

[Hinton *et al.*, 1997] G. Hinton, P. Dayan, y M. Revow. Modeling the manifolds of images of handwriting digits. *IEEE Transactions On Neural Networks*, 8(1):65–74, 1997.

[Hoffman y Jain, 1983] R. Hoffman y A.K. Jain. A test of randomness based on the minimal spanning tree. *Pattern Recognition Letters*, 1(1):175–180, 1983.

[Hornegger y Niemann, 2001] J. Hornegger y H.Ñiemann. A novel probabilistic model for object recognition and pose estimation. *Pattern Recognition and Artificial Intelligence*, 15(2):241–253, 2001.

[Hyvarinen y Oja, 2000] A. Hyvarinen y E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[Hyvarinen, 1998] A. Hyvarinen. New approximations of differential entropy for independent component analysis and projection pursuit. *Advances in Neural Information Processing Systems*, 10:273–279, 1998.

[Jain *et al.*, 2000] A.K. Jain, R. Dubes, y J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–38, 2000.

[Jain y Dubes, 1988] A.K. Jain y R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, N.J., 1988.

[Joe, 1989] H. Joe. On the estimation of entropy and other functionals of a multivariate density. *Ann. Inst. Statist. Math.*, 41:683–697, 1989.

[Jones y Sibson, 1987] M.C. Jones y R. Sibson. *What is projection pursuit*. The Royal Statistical Society, 1987.

[Kloppenburg y Travan, 1997] M. Kloppenburg y P. Travan. Deterministic annealing for density estimation by multivariate normal mixtures. *Physical Rev. E*, 55:R2089–R2092, 1997.

[Kontkanen *et al.*, 1996] P. Kontkanen, P. Myllymaki, y H. Titrri. Comparing bayesian model class selection criteria in discrete finite mixtures. In *Information, Statistics and Induction in Science. ISIS'96*, Singapore, 1996. World Scientific.

[Lanterman, 2001] A. Lanterman. Schwarz, wallace, and rissanen: Intertwining themes in theories of model order estimation. *Intl Statistical Rev.*, 69:185–212, 2001.

[Law *et al.*, 2004] M.H.C. Law, M.A.T. Figueiredo, y A. K. Jain. Ssimultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[Li y Barron, 2000] J. Q. Li y A. R. Barron. Mixture density estimation. *Advances in Neural Information Processing Systems. MIT Press*, 12, 2000.

[Li, 1995] S.Z. Li. *Markov random field modeling in computer vision*. Springer-Verlag London, U.K., 1995.

[Liang *et al.*, 1992] Z. Liang, R. Jaszczak, y R. Coleman. Parameter estimation of finite mixtures using the em algorithm and information criteria with application to medical imaging processing. *IEEE Transactions on Information Theory*, 39(4):1126–1133, 1992.

[Lindsay, 1983] B. G. Lindsay. The geometry of mixture likelihoods: a general theory. *Ann. Statist.*, 11(1):86–94, 1983.

[Lloyd, 1982] S.P. Lloyd. Least squares quantization in pcm. *IEEE Transactions On Information Theory*, 28(2):129–136, 1982.

[Mardia, 1970] K.V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3):519–530, 1970.

[Martin *et al.*, 2001] D. Martin, C. Fowlkes, D. Tal, y J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Intertational Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[Martinez y Vitria, 2000] A. M. Martinez y J. Vitria. Learning mixture models using a genetic version of the em algorithm. *Pattern Recognition Letters*, 21(1):759–769, 2000.

[Martinez y Vitria, 2001] A. M. Martinez y J. Vitria. Clustering un image space for place recognition and visual annotations for human-roboter interaction. *IEEE Transactions on Systems, Man and Cybernetics B*, 31(5):669–682, 2001.

[McLachlan y Basford, 1988] G. McLachlan y K. Basford. *Mixture Models: Inference and Application to Clustering*. Marcel Dekker, New York, 1988.

[McLachlan y Krishnan, 1997] G. McLachlan y T. Krishnan. *The EM Algorithm and Extensions*. John Wiley and Sons, New York, 1997.

[McLachlan y Peel, 2000] G. McLachlan y D. Peel. *Finite Mixture Models*. John Wiley and Sons, New York, 2000.

[Miller y Fisher, 2003] E. G. Miller y J.W. Fisher. Ica using spacings estimates of entropy. *Journal of Machine Learning Research*, 4:1271–1295, 2003.

[Mitchell, 1997] T. M. Mitchell. *Machine Learning*. Mc Graw-Hill, Boston, Massachusetts, 1997.

[Mokkadem, 1989] A. Mokkadem. Estimation of the entropy and information of absolutely continuous random variables. *IEEE Transactions on Information Theory*, 35(1):193–196, 1989.

[Nobile y Green, 2000] A.Ñobile y P.J. Green. Bayesian analysis of factorial experiments by mixture modelling. *Biometrika*, 87:15–35, 2000.

[Oliver *et al.*, 1996] J. Oliver, R. Baxter, y C. Wallace. Unsupervised learning using mml. In *Proceedings of 13th International Conference on Machine Learning*, pages 364–372, 1996.

[Paninski, 2003] I. Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(1):1191–1253, 2003.

[Parzen, 1962] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(1):1065–1076, 1962.

[Peñalver *et al.*, 2003] A. Peñalver, J.M. Sáez, y F. Escolano. An entropy maximization approach to optimal model selection in gaussian mixtures. *Progress in Pattern Recognition, Speech and Image Analysis - CIARP 2003, Lecture Notes in Computer Science*, 2905:432–439, 2003.

[Peñalver *et al.*, 2006a] A. Peñalver, F. Escolano, y J.M. Sáez. Color image segmentation through unsupervised gaussian mixture models. *Image Processing, Computer Vision, Pattern Recognition, and Graphics - SPR/SSPR 2006, Lecture Notes in Computer Science*, 4109:649–657, 2006.

[Peñalver *et al.*, 2006b] A. Peñalver, F. Escolano, y J.M. Sáez. Ebem: An entropy-based em algorithm for gaussian mixture models. In *18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China*, pages 451–455, 2006.

[Peñalver *et al.*, 2006c] A. Peñalver, F. Escolano, y J.M. Sáez. Two entropy-based methods for learning unsupervised gaussian mixture models. *Advances in Artificial Intelligence - IBERAMIA-SBIA 2006, Lecture Notes in Computer Science*, 4140:149–158, 2006.

[Pernkopf y Bouchaffra, 2006] F. Pernkopf y D. Bouchaffra. Genetic-based em algorithm for learning gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1344–1348, 2006.

[Pudil *et al.*, 1995] P. Pudil, J.Ñovovicova, y J. Kittler. Feature selection based on the approximation of class densities by finite mixtures of the special type. *Pattern Recognition*, 28(9):1389–1398, 1995.

[Redner y Walker, 1984] R.A. Redner y H.F. Walker. Mixture densities, maximum likelihood, and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.

[Renyi, 1961] A. Renyi. On measures of entropy and information. *4th Berkeley Symp. Math. Stat. Prob.*, 1:547–561, 1961.

[Richardson y Green, 1997] S. Richardson y P.J. Green. On bayesian analysis of mixtures with unknown number of components (with discussion). *Journal of the Royal Statistical Society B*, 59(1):731–792, 1997.

[Rissanen, 1983] J. Rissanen. Stochastic complexity in statistical inquiry. *The Annals of Statistics*, 11(2):416–431, 1983.

[Rissanen, 1989] J. Rissanen. A universal prior for integers and estimation by minimum description length, 1989. World Scientific.

[Robert *et al.*, 2000] C. P. Robert, T. Rydén, y D.M. Titterington. Bayesian inference in hidden markov models through the reversible jump markov chain monte carlo method. *Journal of the Royal Statistical Society B*, 62:57–76, 2000.

[Roberts *et al.*, 1998] S. Roberts, D. Husmeier, I. Rezek, y W. Penny. Bayesian approaches to gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, 1998.

[Schwarz, 1978] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[Sáez *et al.*, 2003] J.M. Sáez, A. Peñalver, y F. Escolano. Compact mapping in plane-parallel environments using stereo vision. *Progress in Pattern Recognition, Speech and Image Analysis, CIARP 2003, Lecture Notes in Computer Science*, 2905:659–666, 2003.

[Shannon, 1948] C. Shannon. A mathematical theory of comunication. *The Bell System Technical Journal*, 27(1):379–423, 1948.

[Smyth, 1996] P. Smyth. Clustering using monte-carlo cross-validation. In *Second International Conference on Knowledge Discovery and Data Minning*, Menlo Park, CA, 1996. AAAI Press.

[Spiegelhalter y Taylor, 1994] D. Michie D.J. Spiegelhalter y D.J. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Series in Artificial Intelligence, 1994.

[Srebro *et al.*, 2006] N. Srebro, G. Shakhnarovich, y S. Roweis. An investigation of computational and informational limits in gaussian mixture cluste-

ring. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 865–872, New York, NY, USA, 2006. ACM Press.

[Streit y Luginbuhl, 1994] R. Streit y T. Luginbuhl. Maximum likelihood training of probabilistic neural networks. *IEEE Transactions On Neural Networks*, 5(5):764–783, 1994.

[Titterington *et al.*, 1985] D Titterington, A. Smith, y U. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley and Sons, Chichester, UK, 1985.

[Tu y Zhu, 2002] Z. W. Tu y S. C. Zhu. Image segmentation by datadriven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 24(1):657–673, 2002.

[Ueda *et al.*, 2000] N. Ueda, R.Ñakano, Z. Ghahramani, y G. E. Hinton. Smem algorithm for mixture models. *Neural Computation*, 12(1):2109–2128, 2000.

[Viola *et al.*, 1996] P. Viola, N.Ñ. Schraudolph, y T. J. Sejnowski. Empirical entropy manipulation for real-world problems. *Advances in Neural Information Processing Systems*, 8(1), 1996.

[Viola y Wells, 1995] P. Viola y W. M. Wells. Alignment by maximization of mutual information. In *5th International Conference on Computer Vision*. IEEE, 1995.

[Viola, 1995] P. Viola. Alignment by maximization of mutual information. Technical Report 1548, Massachusetts Institute of Technology. Artificial Intelligence Laboratory, Massachusetts, 1995.

[Vlassis *et al.*, 2000] N. Vlassis, A. Likas, y B. Krose. A multivariate kurtosis-based dynamic approach to gaussian mixture modeling, 2000. Inteligent Autonomous Systems Technical Report.

[Vlassis y Likas, 1999] N. Vlassis y A. Likas. A kurtosis-based dynamic approach to gaussian mixture modeling. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(4):393–399, 1999.

[Vlassis y Likas, 2000] N. Vlassis y A. Likas. A greedy em algorithm for gaussian mixture learning. *Neural Processing Letters*, 15(1):77–87, 2000.

[Wallace y Dowe, 1999] C. Wallace y D. Dowe. Minimum message length and kolgomorov complexity. *The Computer J.*, 42(4):270–283, 1999.

[Wallace y Freeman, 1987] C. Wallace y P. Freeman. Estimation and inference via compact coding. *Journal of The Royal Statistical Soc. (B)*, 49(3):241–252, 1987.

[Wallace y Freeman, 1992] C. Wallace y P. Freeman. Single-factor analysis by minimum message length estimation. *Journal of The Royal Statistical Soc. (B)*, 54(1):195–209, 1992.

[Wand, 1994] M.P. Wand. Fast computation of multivariate kernel estimators. *J. Comp. and Graph. Statistics*, 3(4):433–445, 1994.

[Whindham y Cutler, 1992] M. Whindham y A. Cutler. Information ratios for validating mixture analysis. *Journal Am. Statistical Association*, 87:1188–1192, 1992.

[Wolpert y Wolf, 1995] D. Wolpert y D. Wolf. Estimating function of probability distribution from a finite set of samples. *Phisical Review E*, 52(6), 1995.

[Xu y Jordan, 1996] L. Xu y M. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.

[Xu, 1997] L. Xu. Bayesian ying-yang machine, clustering and number of clusters. *Pattern Recognition Letters*, 18(1):1167–1178, 1997.

[Xu, 2002] L. Xu. Byy harmony learning, structural rpcl, and topological self-organizing on mixture models. *Neural Networks*, 15(1):1125–1151, 2002.

[Zhang *et al.*, 2003] Z. Zhang, C. Chen, J. Sun, y K.L. Chan. Em algorithms for gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36(1):1973–1983, 2003.

[Zhang *et al.*, 2004] Z. Zhang, K.L. Chan, Y. Wu, y C. Chen. Learning a multivariate gaussian mixture models with the reversible jump mcmc algorithm. *Statistics and Computing*, 14(1):343–355, 2004.

[Zhang y Ma, 2004] J. Zhang y D. Ma. Non linear prediction for gaussian mixture image models. *IEEE Transactions on Image Processing*, 13(6):836–847, 2004.

[Zhu *et al.*, 2000] S.C. Zhu, R. Zhang, y Z. Tu. Integrating bottom-up for object recognition by data driven markov chain montecarlo. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000.

[Zyczkowski, 2003] K. Zyczkowski. Renyi extrapolation of shannon entropy. *Open Systems and Information Dynamics*, 10(3):297–310, 2003.