ESTIMACIÓN DE DENSIDADES MULTIVARIADAS EN FLUJO DE DATOS USANDO MEZCLAS ADAPTATIVAS DE COMPONENTES GAUSSIANAS.

Por
Héctor Javier Moyano Niño

Tesis sometida en cumplimiento parcial de los requerimientos para el grado de
MAESTRÍA EN CIENCIAS
en
MATEMÁTICAS (ESTADÍSTICA)

UNIVERSIDAD DE PUERTO RICO
RECINTO MAYAGÜEZ
2012

Aprobada por:

Edgar Acuña, Ph.D. Presidente, Comité Graduado	Fecha
Edgardo Lorenzo, Ph.D. Miembro, Comité Graduado	Fecha
Raúl Macchiavelli, Ph.D. Miembro, Comité Graduado	Fecha
Mario Córdova, Ph.D. Representante de Estudios Graduados	Fecha
Omar Colón, Ph.D. Director del Departamento	Fecha

Abstract of Thesis Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Master of Science

MULTIVARIATE DENSITY ESTIMATION IN DATA STREAM USING ADAPTIVE MIXTURES OF GAUSSIAN COMPONENTS.

By

Héctor Javier Moyano Niño

2012

Chair: Edgar Acuña.

Major Department: Department of Mathematical Sciences

In the current world of science and technology the data arrive continuously over time, this type of data is called data stream and is impractical to store all of the data. The data mining and traditional techniques of analysis aren't efficient enough to work with problems that have data stream. Then it is necessary to have statistical models for data stream.

The adaptive mixtures (AM) is an estimation method that combines Gaussian mixture modeling and estimation via kernel. Also has as one of its main features, constant updating with the arrival sequence data. Therefore the adaptive mixtures (AM) are very attractive for modeling the data stream. To adapt the idea adaptive mixtures to data streams presents some problems such as creating models of mixtures with too many components, slight changes in the estimated model parameters due to ordering in

the arrival of new data and little applicability to space of high dimension. Many of these problems have been treated recently with the adequacy of expectation-maximization algorithm online (oEM) to the process of adaptive mixtures for data stream (oAM). The thesis presents the study of adaptive mixtures for modeling multidimensional data flow using Gaussian components. Also, it presentes an experimental study with artificial data to control the growth in the number of components and improve the estimation of model components using what I call graphs adjustment components.

All the theoretical framework and the algorithms presented here are directed to estimate multivariate densities, but the experimental part was carried out and implemented in R statistical programming language for data in two and three dimensions.

Resumen de Tesis Presentada a Escuela Graduada
de la Universidad de Puerto Rico como Requisito Parcial de los
Requerimientos para el Grado de Maestría en Ciencias

ESTIMACIÓN DE DENSIDADES MULTIVARIADAS EN FLUJO DE DATOS USANDO MEZCLAS ADAPTATIVAS DE COMPONENTES GAUSSIANAS.

Por

Héctor Javier Moyano Niño

2012

Consejero: Edgar Acuña.

Departamento: Departamento de Ciencias Matemáticas

En muchas aplicaciones de ciencia y tecnología de la actualidad los datos llegan en forma continua en el tiempo y es poco práctico almacenar la totalidad de éstos, por lo que técnicas tradicionales de análisis y minería de datos son pocos eficientes para tratar con problemas que relacionen esta clase de datos. Se hace necesario tener modelos estadísticos para el flujo de datos. Las mezclas adaptativas (AM) es un método de estimación que combina el modelado con mezclas gaussianas y la estimación tipo núcleo, y además tiene como una de sus principales características su constante actualización con la llegada secuencial de datos. Por lo tanto las mezclas adaptativas (AM) son muy atractivas para modelar la clase de datos en cuestión. Adecuar la idea de mezclas adaptativas a flujos de datos presenta algunos problemas tales como la creación de

modelos de mezclas con demasiadas componentes, ligeros cambios en los parámetros de los modelos estimados debido al ordenamiento en la llegada de un nuevo dato y la poca aplicabilidad a espacios de alta dimensión. Gran parte de estos problemas han sido tratados recientemente con la adecuación del algoritmo de esperanza-maximización en linea (oEM) al proceso de mezclas adaptativas para flujo de datos (oAM). En esta tesis se presenta el estudio de mezclas adaptativas para el modelado multidimensional de flujo de datos usando componentes gaussianas y se presenta al final un estudio experimental con datos artificiales para controlar el crecimiento en el número de componentes y mejorar la estimación de las componentes del modelo usando lo que aquí se ha denominado ajuste de componentes con grafos. Todo el marco teórico y los algoritmos aquí presentados están orientados para la estimación de densidades multivariadas, pero la parte experimental fue realizada y ejecutada en lenguaje de programación estadístico R para datos en dos y tres dimensiones.

Dedicado a:

Mis padres Margarita (q.e.p.d) y Esteban. A mis Hermanos Jhon, Luisa y Felipe.

AGRADECIMIENTOS

A Dios, por darme fortaleza y sabiduría para seguir luchando.

Al Dr. Edgar Acuña, por su paciencia y dedicación en el desarrollo de este trabajo.

Al Dr. Raúl Macchiavelli, por sus enseñanzas y amabilidad durante toda mi estadía en Puerto Rico.

Al Dr. Edgardo Lorenzo, por sus aportes en este trabajo.

Al Dr. Daniel McGee, por darme la oportunidad de ser parte de su equipo.

Al Dr. Pedro Vasquez, por la confianza depositada en mi.

A María Isabel, por regalarme una ilusión de vida.

A Yuri, Ricela, Omaira, Shirley, Greichaly, Widad, Claudia, Milena, Belsi, Glorimar, Isnardo, Edwin, Abner, Carlos, Jesus R, Jesus C, Jairo, Roberto, Jose, Yovani, Alex, por su apoyo, compañía y amistad durante estos tres años y medio.

A todos y cada una de las personas que me extendieron su mano mientras mi estadía en la Isla del Encanto.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

LISTA DE ALGORITMOS

ÍNDICE DE TABLAS

CAPÍTULO 1

INTRODUCCIÓN

En los últimos años, los avances tecnológicos han hecho que muchas de las nuevas aplicaciones funcionen con grandes volúmenes de datos, como por ejemplo los registros de clientes en operaciones en línea, los registros de llamadas en las empresas de telecomunicaciones, los conjuntos grandes de páginas Web, los datos multimedia, las transacciones bancarias, etc. A menudo, los datos llegan de forma continua y su naturaleza es transitoria (es decir que se hace innecesario o poco práctico almacenar los datos), refiriéndose a éstos como flujo de datos. El volumen de los flujos de datos, así como la naturaleza rápida de llegada, hacen que técnicas tradicionales de análisis y minería de datos no sean eficientes, pues los datos se acumulan más rápido de lo que pueden ser procesados. Esto produce problemas de costo y de procesamiento compu-

tacional, lo que hace necesario el modelado de este tipo especial de datos.

La estimación de la función de probabilidad de una variable aleatoria $\mathbf{x}$ de dimensión d, para una muestra $x_1, x_2, \ldots, x_n$, es un método muy importante y utilizado en minería y análisis de datos. El objetivo principal en la estimación de densidad es modelar la función de densidad de probabilidad de una distribución desconocida solamente usando una muestra representativa de los datos. Por lo tanto, un estimador de densidad bien definido permite tratar problemas estadísticos tales como problemas de agrupamiento (clustering), clasificación, regresión, análisis de series de tiempo, etc. Esto hace necesario la adaptación de las estimaciones de densidades a los flujos de datos. En está tesis se estudia un método híbrido (con diferentes mejoras) de otros ya existentes para la estimación de densidades, pero con la novedad de ser adaptados a flujos de datos. La tesis está dividida en seis capítulos cuyos contenidos se describen a continuación.

El segundo capítulo presenta los conceptos básicos necesarios para el entendimiento y desarrollo de esta tesis. Se inicia el capítulo con la sección 2.1 donde se definen conceptos básicos de teoría de grafos. En esta sección se presenta el concepto de *grafo* y la *matriz de adyacencia*, necesarios en los últimos capítulos de la tesis. La sección 2.2 trata sobre *la descomposción espectral de una matriz* simétrica. En la sección 2.3 se presenta la densidad gaussiana multivariada, la cual será nuestra distribución base en el desarrollo de toda la tesis. El capítulo dos finaliza en la sección **??** con el estudio de las *errores de estimación en una densidad de probabilidad*.

El capítulo tres presenta algunos métodos para la *estimación de densidades de probabilidad*, la medición entre las densidades objetivo y las estimadas, así como las agrupaciones que se pueden hacer con las componentes de la densidad estimada. Las técnicas parámetricas de estimación de densidades consideran muestras que tienen distribucio-

nes conocidas para así luego calcular los estimados de los parámetros. Dicha suposición usualmente se basa en informaciones sobre el vector aleatorio d-dimesional que son externas a la muestra, pero cuya validez puede ser comprobada con posterioridad mediante pruebas de bondad de ajuste. Las técnicas no-parámetricas no predeterminan ningún modelo para la distribución de probabilidad y dejan que la función de densidad pueda adoptar cualquier forma.

La discusión sobre el uso de una estimación parámetrica o no-parámetrica no ha cesado a lo largo de los años. La eficiencia que proporciona la estimación con técnicas paramétricas se contraponen al riesgo de las *malas suposiciones* adoptadas para determinar el modelo, conduciendo esto a errores de interpretación y proporcionando una pérdida mayor que la ganancia proporcionada por la eficacia de la estimación.

La sección 3.1 de este capítulo inicia con la presentación del método no-parámetrico denominado *estimación de densidad núcleo (KDE)* el cual para un mejor entendimiento se presenta tanto para el caso unidimensional como para el caso d-dimesional. Para propósitos de esta tesis nos centraremos en *núcleos gaussianos* d-dimensionales. La sección 3.2 describe un método parámetrico para el modelado de densidades de probabilidad generado por la suma finita de M-componentes en forma de densidades. Igual que antes, para propósitos de esta tesis nos centraremos en *densidades componentes gaussianas* d-dimensionales. Dicha técnica se denomina *mezclas finitas de gaussianas*. En la sección 3.3 se estudian las *medidas de similaridad entre dos mezclas de gaussianas d-dimensionales*. Cabe rescatar que en esta sección se estudia la transformación *unscented* de la distancia de Hellinger para medir la similaridad entre mezclas gaussianas d-dimensionales , la cual juega un importante papel en la medición del error para estimaciones de flujos de datos. Se finaliza este capítulo en la sección 3.4 con la

presentación de un método de agrupamiento (clustering) jerárquico para componentes en Mezclas Gaussianas.

En el capítulo cuatro se adaptan los métodos vistos en el capítulo tres para el estudio de flujos de datos. Debido a que en la mayoría de aplicaciones del mundo real donde surgen estos flujos de datos, no hay un conocimiento a priori sobre sus características y mucho menos sobre la función de densidad de probabilidad (pdf) asociadas a ellos, podríamos considerar herramientas de la estadística matemática para estimar la pdf correspondiente. La combinación de los estimadores de densidad núcleo con los modelos de mezclas gaussianas es una buena estrategia para hacer frente al problema del modelado de flujos de datos. El capítulo inicia con la sección 4.1, donde se describe un método híbrido entre los presentados en la secciones 3.1 y 3.2. En la sección 4.2 se presentan las características y los requisitos de los datos y modelos a trabajar en el resto de la tesis. En la sección 4.3 se estudia la adecuación de las *mezclas adaptativas* vistas en la sección 4.1 para el caso de datos en línea. Este paso es posible gracias a la versión en línea del algoritmo (EM) denominado (oEM). Al final de esta sección se propone un ajuste al método de *mezclas adaptativas* para el caso de datos en línea usando agrupaciones con grafos. En el capítulo cinco se presenta un estudio experimental para validar el ajuste a los métodos anteriormente presentados en el capítulo cuatro, usando datos simulados de dimensión dos y tres. Se finaliza esta tesis en el capítulo seis donde se exponen las conclusiones de los experimentos, así como los trabajos futuros.

CAPÍTULO 2

CONCEPTOS PRELIMINARES

2.1. Conceptos Básicos de Teoría de Grafos

En términos generales, un grafo consiste en un conjunto de puntos, que llamaremos *vértices*, y líneas que unen los vértices, que denominaremos *aristas*. En la Figura 2.1 se observa un ejemplo de un grafo con 6 vértices y 7 aristas.

Definición 2.1. *Un* ***grafo*** *está formado por un par de conjuntos finitos, y se denota por* $G = (V, A)$*, donde* V *es el conjunto de* ***vértices*** *y* A *es el conjunto de* ***aristas****.*

Cada arista de $a \in A$ conecta dos vértices de V, que llamaremos *extremos de la arista*, y escribiremos $a = \{x, y\}$ para indicar que a conecta o une los vértices x e y. Diremos entonces que x e y son adyacentes por a. Para el ejemplo mostrado en la

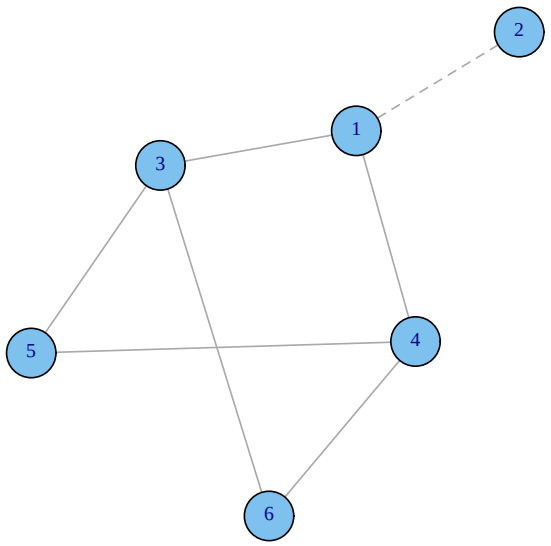

Figura 2.1: Ejemplo de un grafo con 6 vértices y 7 aristas.

Figura 2.1 se tiene:

$$V = \{1, 2, 3, 4, 5, 6\}$$
$$A = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 5\}, \{3, 6\}, \{4, 5\}, \{4, 6\}\} \tag{2.1}$$

En un grafo podemos encontrar*lazos* (aristas cuyos extremos coinciden), aristas múltiples (más de una arista conectando los mismos vértices) y vértices aislados (no están conectados a ningún otro vértice). Pero también podemos hablar de *grafos dirigidos*, donde cada arista tiene una dirección de recorrido; modelos para una distribución de agua por la red de tuberias de la ciudad, la red vial con calles de sentido único, etc., son ejemplos de grafos dirigidos.

Definición 2.2. *Un **digrafo** o **grafo dirigido** está formado por un par de conjuntos finitos, y se denota por $D = (V, A)$, donde V es el conjunto de **vértices** y A es el conjunto de arcos o **aristas dirigidas** entre los vértices.*

Cada arco $a \in A$ conecta dos vértices de V, que llamaremos respectivamente *extremo inicial* y *extremo final* del arco, y escribiremos $a = (x, y)$ para indicar que a conecta o une el vértice x con el vértice y. Diremos también que x es adyacente a y y que a incide en y. Los grafos se representan con puntos y líneas que los unen, los digrafos se representan con puntos y flechas entre ellos.

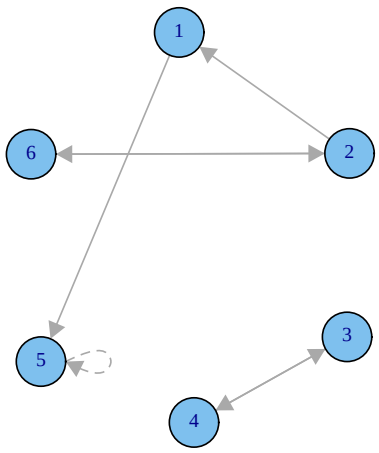

Figura 2.2: Ejemplo de un grafo dirigido con 6 vértices, 6 aristas dirigidas (arcos) y un lazo.

Un digrafo con dos arcos de sentidos contrarios puede considerarse como un grafo

(no dirigido). En la Figura 2.2 se observa el ejemplo de un digrafo con 6 vértices, 6 aristas dirigidas (arcos) y un lazo. Para este caso se tiene el siguiente conjunto de vértices y conjunto de arcos:

$$V = \{1, 2, 3, 4, 5, 6\}$$
$$A = \{(1, 5), (2, 1), (2, 6), (3, 4), (4, 3), (5, 5), (6, 2)\} \tag{2.2}$$

Definición 2.3. *Un **subgrafo** o **subdigrafo** de un grafo (digrafo), es un grafo (digrafo) formado con vértices y aristas (arcos) del inicial.*

Es decir, se obtienen eliminando aristas y/o vértices del inicial (si se elimina un vértice, también deben eliminarse todas las aristas incidentes en él).

2.1.1. Matriz de Adyacencia de un Grafo

Un grafo o un digrafo $D = (V, A)$ puede también describirse mediante una tabla o matriz que indique las conexiones:

Definición 2.4. *Si $\boldsymbol{D}$ tiene d vértices, se llama **Matriz de Adyacencia** de D a la matriz cuadrada de orden d, expresada por $M_{Ady} = (m_{ij})_{d\times d}$ donde $m_{ij} = 1$ si el arco $(v_i, v_j) \in A$ y $m_{ij} = 0$ en otro caso.*

La matriz de adyacencia para la Figura 2.2 está dada por:

$$\mathbf{M}_{Ady} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Si G es no dirigido, su matriz de **adyacencia** es simétrica (si la arista $\{v_i, v_j\}$ está en A, también está $\{v_j, v_i\}$). En esta tesis se trabajan matrices de adyacencia simétricas con valores en la diagonal cero (los digrafos no tienen lazos). En un digrafo, el número de unos de cada fila de la matriz de adyacencia corresponde a el número de arcos salientes desde ese vértice y el número de unos de cada columna indica el número de arcos que llegan a ese vértice.

2.2. Descomposición Espectral de una Matriz.

Sea $\mathbf{\Sigma}$ una matriz simétrica definida positiva de orden d. Por álgebra matricial se sabe que para esta clase de matrices los valores propios son números reales y los vectores propios son ortogonales. Luego la matriz $\mathbf{\Sigma}$ puede descomponerse como:

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^T, \tag{2.3}$$

donde $\mathbf{D}$ es una matriz diagonal formada por los valores propios de $\mathbf{\Sigma}$ y $\mathbf{U}$ es una matriz ortogonal cuyas columnas son los vectores propios unitarios asociados con los

elementos de la diagonal de la matriz **D**. Esta propiedad se conoce con el nombre de **la descomposición espectral**. Llamando $\lambda_1, \ldots, \lambda_d$ a los valores propios de la matriz $\mathbf{\Sigma}$ y $\mathbf{u}_1, \ldots, \mathbf{u}_d$ a sus respectivos vectores propios, la descomposición dada en (2.3) puede escribirse:

$$\mathbf{\Sigma} = \sum_{i=1}^{d} \lambda_i \mathbf{u}_i \mathbf{u}_i^T, \tag{2.4}$$

que descompone la matriz $\mathbf{\Sigma}$ como la suma de d matrices de rango uno, $\mathbf{u}_i \mathbf{u}_i^T$, con coeficientes λ_i.

La importancia de esta descomposición es que si algunos valores propios son muy pequeños, podemos reconstruir aproximadamente $\mathbf{\Sigma}$ utilizando los restantes valores y vectores propios. Observemos también que la descomposición espectral de $\mathbf{\Sigma}^{-1}$ es

$$\mathbf{\Sigma}^{-1} = \sum_{i=1}^{d} \lambda_i^{-1} \mathbf{u}_i \mathbf{u}_i^T,$$

ya que $\mathbf{\Sigma}^{-1}$ tiene los mismos vectores propios de $\mathbf{\Sigma}$ y valores propios λ_i^{-1}.

2.3. La Densidad Gaussiana Multivariada

Una *densidad gaussiana d-dimensional* para una variable aleatoria vectorial $\mathbf{x}$, viene dada por:

$$\phi(\mathbf{x}, \mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) \right\}, \tag{2.5}$$

donde $\mu = (\mu_1, \ldots, \mu_d)^T$, $\mathbf{\Sigma}$ es una matriz simétrica definida positiva de dimensión $(d \times d)$ denominada la matriz de covarianzas y $|\mathbf{\Sigma}|$ denota el determinante de $\mathbf{\Sigma}$.

La Figura 2.3 muestra la densidad gaussiana estandar bivariada, la cual tiene vector de medias $\mu = \mathbf{0}_{2\times 1}$ y matriz de covarianzas $I_{2\times 2}$

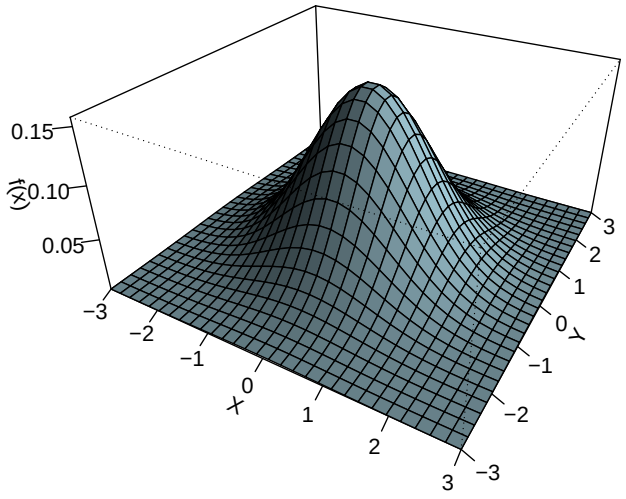

Figura 2.3: Densidad gaussiana estándar bivariada.

2.3.1. Geometría de la Densidad Gaussiana Multivariada.

El exponente $(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)$ de la función de densidad gaussiana multivariada dado en la expresión (2.5), corresponde a la ecuación de un elipsoide en el espacio d-dimensional cuando este es igual a una constante $\mathcal{C}$; este elipsoide se obtiene al cortar con un hiperplano, paralelo al definido por las d-variables que forman la variable

vectorial $\mathbf{x}$, la densidad gaussiana d-dimensional. Luego la ecuación:

$$(\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) = \mathcal{C} \tag{2.6}$$

define las curvas de nivel y éstas a su vez una medida de la distancia de un punto x al centro de la densidad. Para el caso $d = 2$, los planos cortan la densidad gaussiana formando elipses. La Figura 2.4 muestra una densidad gaussiana bivariada con $\mu = (3, 4)^T$ y $\Sigma_{11} = \Sigma_{22} = 1, \Sigma_{12} = \Sigma_{21} = 0.5$. También se observan las elipses que forman las curvas de nivel de la densidad gaussiana multivariada.

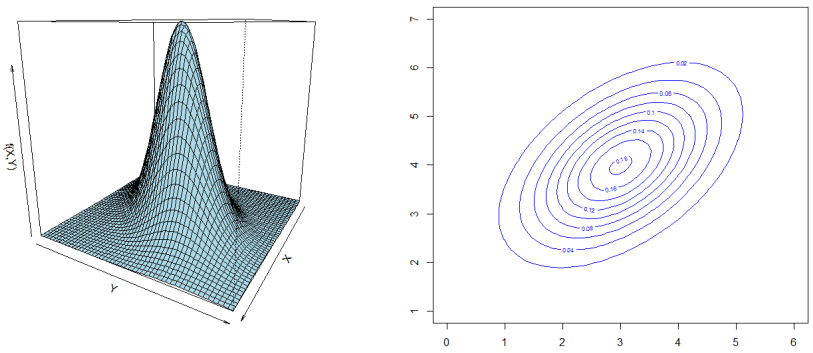

Figura 2.4: Densidad gaussiana bivariada con $\mu = (3, 4)^T$ y $\Sigma_{11} = \Sigma_{22} = 1$, $\Sigma_{12} = \Sigma_{21} = 0.5$ y sus respectivas curvas de nivel.

La ecuación (2.6) determina una medida que se denomina *distancia de Mahalanobis* y la representamos por:

$$D_M^2(\mathbf{x_i}, \mu; \mathbf{\Sigma}) = (\mathbf{x_i} - \mu)' \mathbf{\Sigma}^{-1} (\mathbf{x_i} - \mu), \tag{2.7}$$

donde $\mathbf{x_i} = (x_{i1}, \ldots, x_{id})^T$ representa un individuo particular, seleccionado aleatoriamente de una población con centro en $\mu = (\mu_1, \ldots, \mu_d)^T$ y matriz de covarianzas $\mathbf{\Sigma}$.

Proposición 2.1. *Sea* $\mathbf{x} \sim \phi(\mu, \mathbf{\Sigma})$*, la distancia de Mahalanobis*

$$D_M^2(\mathbf{x}, \mu; \mathbf{\Sigma}) = (\boldsymbol{x} - \mu)^T \mathbf{\Sigma}^{-1} (\boldsymbol{x} - \mu)$$

se distribuye como una χ^2 *con* d *grados de libertad*

$$D_M^2 \sim \chi_d^2$$

Demostración. Realizando la transformación lineal o *estandarización* del vector $\mathbf{x}$ así:

$$\mathbf{z} = \mathbf{\Sigma}^{-1/2}(\mathbf{x} - \mu), \tag{2.8}$$

se tiene que $\mathbf{z}$ se distribuye como una gaussiana d-dimensional con vector de medias cero y matriz de varianzas y covarianzas I_d, donde $\mathbf{\Sigma}^{-1/2} = (\mathbf{\Sigma}^{-1})^{1/2}$. Luego si tomamos $\mathbf{\Sigma}^{-1} = \mathbf{\Sigma}^{-1/2}\mathbf{\Sigma}^{-1/2}$ se obtiene que

$$D_M^2 = \mathbf{z}^T \mathbf{z} = \sum \mathbf{z}_i^2,$$

donde cada $\mathbf{z}_i \sim \phi(0, 1)$ y $\mathbf{z}_i, \mathbf{z}_j$ son independientes $\forall i \neq j$. Puesto que $\mathbf{z}_i^2 \sim \chi_d^2$ y $\sum \mathbf{z}_i^2 \sim \chi_d^2$, se tiene: $D_M^2 \sim \chi_d^2$ ■

La Figura 2.5 muestra la relación entre la Distancia de Mahalanobis y la distribución χ_d^2 para $d = 2, 3, 4, 5$. (Tomado de la ayuda del comando *mahalanobis* del programa `R`.)

Llamaremos *elipsoide de predicción* con $(1 - \alpha) \times 100\,\%$ de confianza, al elipsoide que cumple la relación:

$$(\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) = \chi_{\alpha,d}^2 \tag{2.9}$$

En Johnson y Wichern [30], se muestra cómo la geometría de la distribución gaussiana d-dimensional puede ser estudiada teniendo en cuenta la orientación y la forma

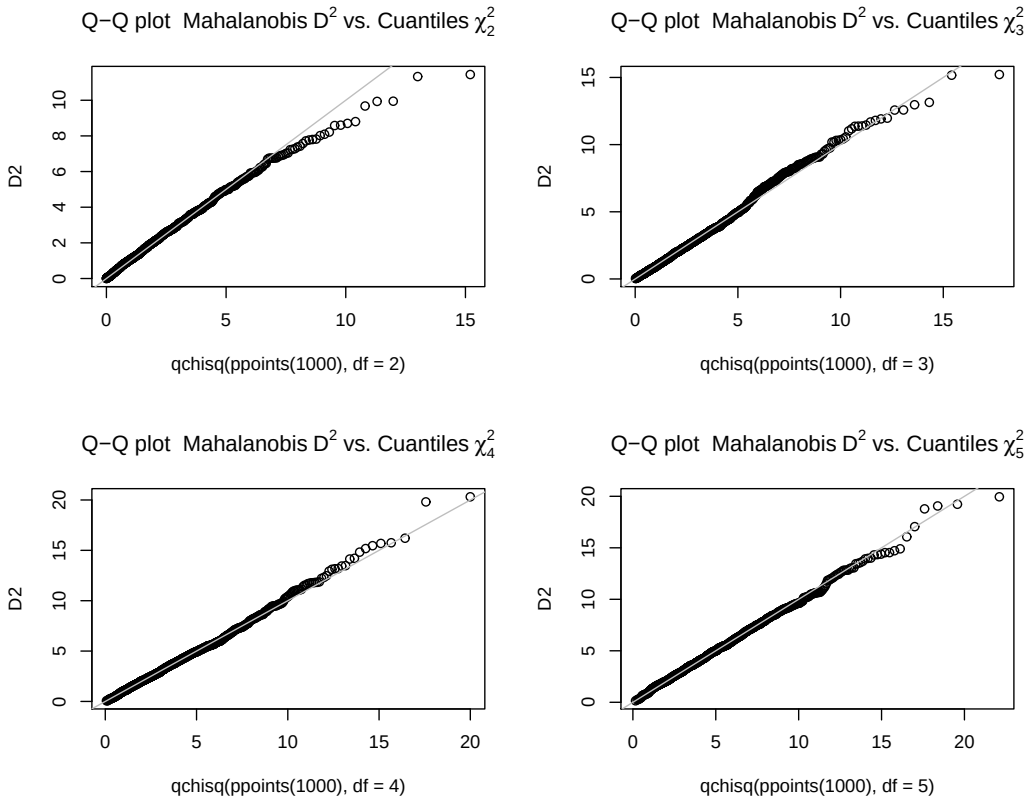

Figura 2.5: Q-Q Plot Distancia de Mahalanobis D_M^2 Vs cuantiles χ_d^2.

del elipsoide de predicción, la cual está determinada por la matriz de covarianzas $\mathbf{\Sigma}$. La descomposición espectral de $\mathbf{\Sigma}$ dada en la Sección 2.2, permite hallar los valores y vectores propios de la matriz $\mathbf{\Sigma}$, los cuales especifican la orientación y la longitud de los semiejes del elipsoide dado por la ecuación (2.9). En la Figura 2.6 se observa la relación de la elipse de confianza del $(1-\alpha)\times 100\,\%$, para una densidad gaussiana bivariada, de vector de medias $\mu = (\mu_1, \mu_2)$ y matriz de covarianzas $\mathbf{\Sigma}$, con los valores y vectores propios de la descomposición espectral de la matriz de covarianzas $\mathbf{\Sigma}$.

Los elipsoides tienen las longitudes de los semiejes determinada por los valores

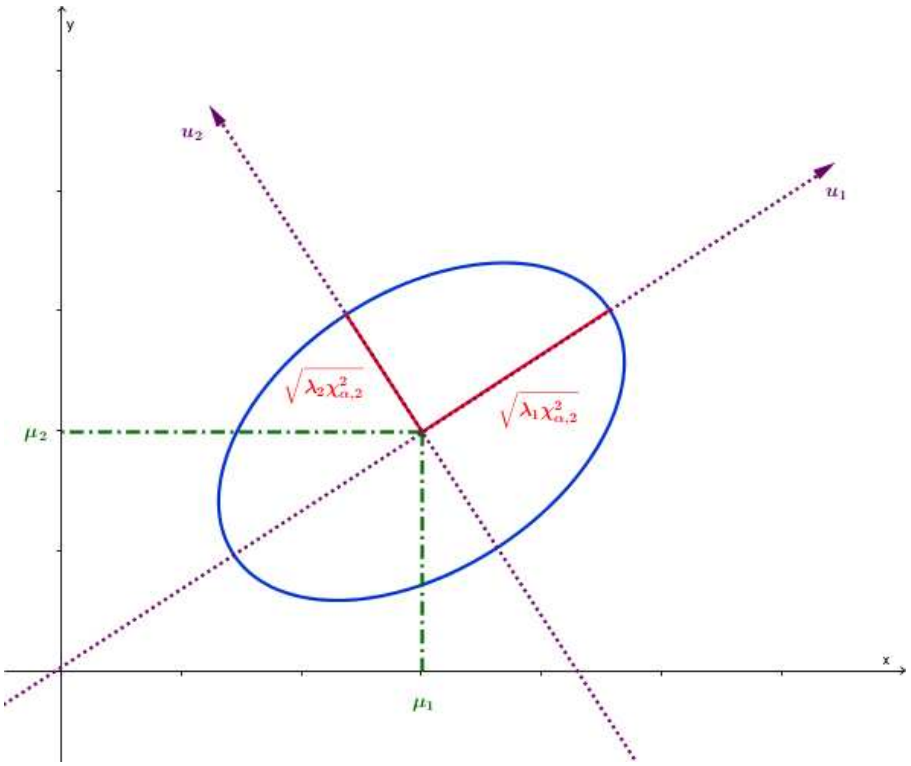

Figura 2.6: Relación de la elipse de confianza de una densidad gaussiana bivariada, de vector de medias $\mu = (\mu_1, \mu_2)$ y matriz de covarianzas $\mathbf{\Sigma}$, con los valores y vectores propios de la descomposición espectral de la matriz de covarianzas $\mathbf{\Sigma}$.

propios $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ en las direcciones de los vectores propios $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$. En la Figura 2.6 para la gaussiana bivariada, el semieje mayor de la elipse con longitud proporcional a $\sqrt{\lambda_1}$ en la dirección del primer vector propio $\mathbf{u}_1$; el semieje más corto es proporcional a $\sqrt{\lambda_2}$, es perpendicular al primero, en la dirección del vector propio $\mathbf{u}_2$. Las longitudes correspondientes a los semiejes se obtienen mediante la siguiente expresión:

$$l_j = \sqrt{\lambda_j \chi^2_{\alpha,d}} \quad con \quad j = 1, 2, \ldots, d. \tag{2.10}$$

CAPÍTULO 3

ESTIMACIÓN, DISTANCIAS Y AGRUPACIONES EN DENSIDADES MULTIVARIADAS.

3.1. Estimación de Densidades con Núcleos Gaussianos

Las aproximaciones a funciones de densidad conocidas (χ^2- chi cuadrado, Beta, distribución t, gaussiana, etc.) presentan problemas en la práctica tales como *el ajuste* y el número de modas (distribuciones multimodales). Como hemos mencionado antes,

las estimaciones no-parámetricas no hacen ningún supuesto sobre la forma de la distribución de la densidad y solo se basan en los datos muestrales. Un estimador clásico no-parámetrico de la función de densidad es el *histograma normalizado* [1]. En general, la estimación con histogramas realiza una partición del espacio muestral en un número de celdas de igual tamaño (*bins*). Luego la estimación de la función de densidad con histogramas estará dada por:

$$\hat{f}(\mathbf{x}) = \frac{1}{n}\left(\frac{\#\ de\ puntos\ en\ el\ bin\ conteniendo\ a\ \mathbf{x}}{volumen\ del\ bin}\right) \tag{3.1}$$

Los histogramas como estimadores no-parámetricos de una función de densidad presentan varios problemas, entre ellos que la función resultante no es una *función suave* [2] ya que genera una función escalonada y además depende de los anchos, así como de los puntos terminales de cada *bin*.

Una forma de resolver estos problemas es considerar *estimaciones de densidad tipo núcleo*. Estas estimaciones usan funciones continuas centradas en cada dato, para construir de forma dinámica las componentes (que llamaremos núcleos), que al promediarse dan forma a la densidad objetivo. En la Figura 3.1 se realiza la estimación de la densidad núcleo bivariada para datos simulados con $\mu = (0, 0)$, $\Sigma_{11} = \Sigma_{22} = 1$ y $\Sigma_{12} = \Sigma_{21} = 0.60$.

En Silverman [32] se pueden consultar en forma detallada los diversos tipos de núcleos existentes. A continuación, discutiremos los estimadores tipo núcleo para el caso de

[1] Debe estar normalizado para integrar a uno. Para más información ver en Miñarro [20].

[2] El principal objetivo en las estimaciones de densidades no-parámetricas es *suavizar* un conjunto de datos creando una función que intente capturar los patrones importantes de dichos datos. Para más información ver Klemela [14].

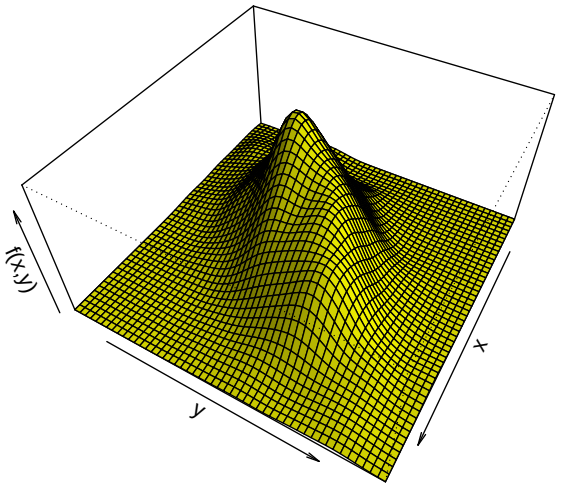

Figura 3.1: Estimación de la densidad bivariada con núcleos gaussianos, para datos simulados de una distribución gaussiana bivariada con $\mu = (0, 0)$, $\Sigma_{11} = \Sigma_{22} = 1$ y $\Sigma_{12} = \Sigma_{21} = 0.60$.

núcleos gaussianos.

3.1.1. Núcleos Gaussianos Univariados

Definición 3.1. *Dada una muestra de n observaciones* $X_1, \ldots, X_n \in \mathbb{R}$*, la fórmula para un estimador de densidad núcleo está dada por:*

$$\hat{f}(\mathbf{x}; h) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - X_i}{h}\right) \tag{3.2}$$

donde:

$\mathbf{x}$ = *Punto en el cual se trata de estimar la densidad.*

X_i = *Valor de la variable en el caso* $i = 1, \ldots, n$.

$K(\cdot)$ = *Función núcleo.*

h = *Número positivo denominado ancho de banda o parámetro de suavizado.*

Una fórmula ligeramente más compacta para el estimador de densidad núcleo puede obtenerse mediante la introducción de la notación de reescalado $K_h(u) = h^{-1}K(u/h)$, lo que nos permite reescribir la ecuación (3.2) como

$$\hat{f}(\mathbf{x}; h) = \frac{1}{n} \sum_{i=1}^{n} K_h(\mathbf{x} - X_i). \tag{3.3}$$

En la Figura 3.2 se muestra una función de densidad núcleo construida usando cinco obervaciones $X_1 = 3,\ X_2 = 4.5,\ X_3 = 5,\ X_4 = 8,\ X_5 = 9$ con núcleos $\phi(0, 1)$. En la práctica se usan muchas más observaciones para construir la estimación de la densidad. Note que la estimación es construida por el promedio de los núcleos centrados en cada observación.

Las funciones de densidad núcleo univariadas más comunes son: Epanechnikov, Gaussiana, Triangular, Rectangular, Biweight, Triweight y Arco coseno. En particular, los núcleos gaussianos para el caso univariado se notan $K_h \equiv \phi_\sigma$ y tienen la forma:

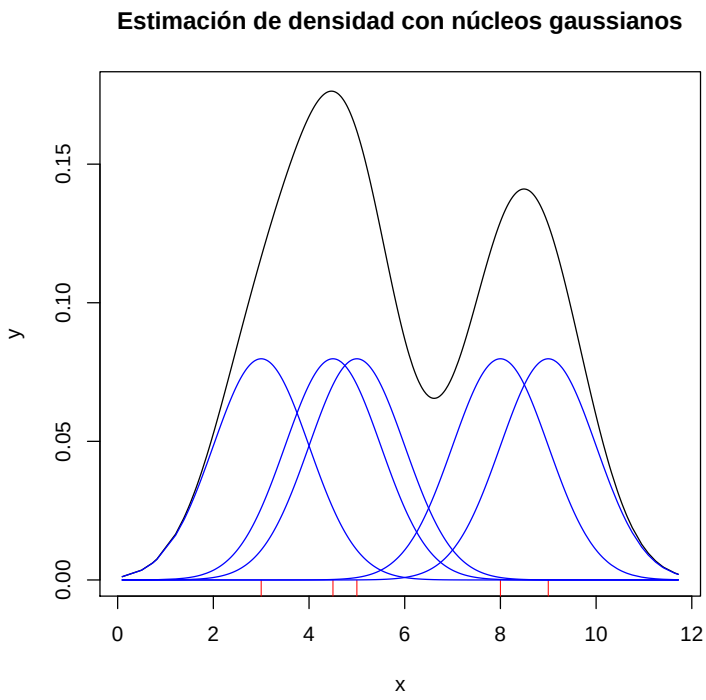

Figura 3.2: Estimación de densidad con núcleos gaussianos $N(0,1)$ para las observaciones $X_1 = 3,\ X_2 = 4.5,\ X_3 = 5,\ X_4 = 8,\ X_5 = 9.$

$$\phi_\sigma(\mathbf{x} - \mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-(\mathbf{x} - \mu)^2/(2\sigma^2)\}. \tag{3.4}$$

En este caso, la desviación estándar σ desempeña el papel de ancho de banda.

3.1.2. Núcleos Gaussianos Multivariados

Definición 3.2. *Para muestras aleatorias d-dimensionales* $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ *de una densidad* $f(x)$*, el estimador de densidad núcleo está dado por:*

$$\hat{f}(\mathbf{x}; H) = \frac{1}{n} \sum_{i=1}^{n} K_H (\mathbf{x} - X_i) \tag{3.5}$$

donde $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$ *y* $X_i = (X_{i1}, X_{i2}, \ldots, X_{id})^T$ *con* $i = 1, 2, \ldots, n$.

Aquí $K_H(\mathbf{x})$ es la *función núcleo*, la cual corresponde a una densidad de probabilidad simétrica y H la *matriz ancho de banda* (o suavizador), con dimensiones $d \times d$, simétrica y definida positiva. Igual que el caso univariado se tiene la notación de escalado

$$K_H(\mathbf{x}) = |H|^{-1/2} K(H^{-1/2}\mathbf{x}) \tag{3.6}$$

En la Figura 3.3 se realiza la estimación de la densidad núcleo bivariada para 272 registros con dos mediciones del tiempo de duración de una erupción (minutos) y el tiempo de espera hasta la próxima erupción (minutos) del *Geyser Old Faithful* en el Parque Nacional de Yellowstone, EE.UU [3]. Se presentan el gráfico de dispersión y el diagrama de contornos de los datos.

Los núcleos multivariados K_H más comunes son: Gaussiano y Bartlett-Epanechnikov. Los diversos núcleos multivariados se pueden consultar en [22]. En particular los siguientes núcleos gaussianos multivariados con $\mu = 0_{1\times d}$ son de uso común:

$$\phi_I(\mathbf{x}) = (2\pi)^{-d/2} \exp\left\{-\frac{1}{2}\mathbf{x}^T\mathbf{x}\right\}, \quad y \quad \phi_\Sigma(\mathbf{x}) = (2\pi)^{-d/2}|\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}\right\} \tag{3.7}$$

[3] Este conjunto de datos está en las bases de datos del programa `R`

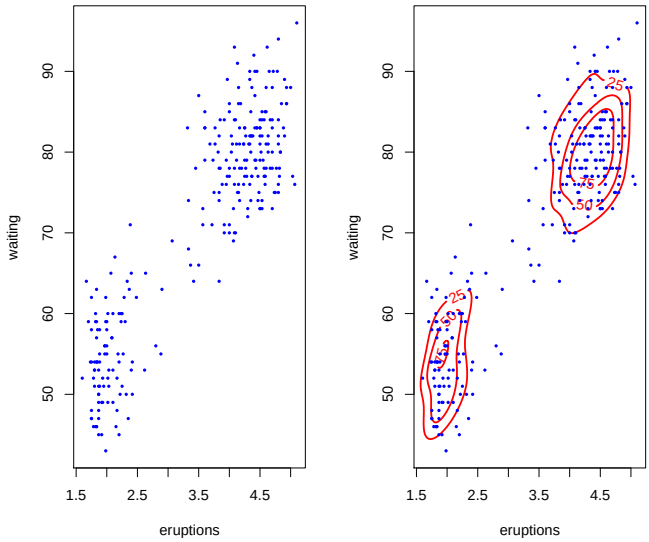

Figura 3.3: Gráfico de dispersión y el diagrama de contornos en la estimación de la densidad núcleo bivariado para 272 registros del *Geyser Old Faithful* en el Parque Nacional de Yellowstone, EE.UU. (fuente: dataset `faithful` programa `R`)

Igual que el caso univariado, lo crucial aquí es la estimación de *la matriz ancho de banda*, que se representa por

$$H = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1d} \\ h_{21} & h_{22} & \dots & h_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ h_{d1} & h_{d2} & \dots & h_{dd} \end{pmatrix}, \tag{3.8}$$

donde $h_{ij} = h_{ji}$ con $i, j = 1, \dots, d$. Luego existen $\frac{1}{2}d(d+1)$ parámetros a estimar. Wand y Jones (1993) proponen tres categorías para el estudio de la matriz H. A continuación se presentan estas categorías.

Clasificación y Transformaciones de la Matriz del Ancho de Banda

La elección de la matriz de ancho de banda H afecta la precisión de la estimación, ya que controla *la orientación* de los núcleos usados. Ésta es una diferencia básica entre la estimación de densidad núcleo multivariada con su análoga univariada, ya que la orientación no está definida para los núcleos con $d = 1$. Luego es natural pensar en *parametrizar* la elección de H. Las tres clases principales de parametrizaciones para H (en orden creciente de complejidad) son:

1. La clase $\mathcal{S}$ a la que pertenecen las matrices de la forma $H = h^2 I$ con $h > 0$. Esta clase tiene el mismo valor de *suavizado* h que se aplica en todas las direcciones de coordenadas, lo cual puede llegar a ser una desventaja en la estimación de la función objetivo.

2. La clase $\mathcal{D}$ a la que pertenecen matrices de la forma $H = diag(h_1^2, h_2^2, \ldots, h_d^2)$. Esta clase permite distintas cantidades de *suavizado* h_i en cada una de las coordenadas.

3. La clase $\mathcal{F}$ a la que pertenecen matrices sin restricciones, simétricas y definidas positivas. Esta clase permite valores y orientaciones arbitrarias de H.

Históricamente, las clasés $\mathcal{S}$ y $\mathcal{D}$ son las más estudiadas debido a razones de cálculo, pero las investigaciones de los últimos años indican que hay importantes ganancias en la precisión utilizando la clase $\mathcal{F}$. Wand y Jones [22], indican que la forma más sencilla de obtener una matriz de ancho de banda con orientacion arbitraria es con:

$$H = h^2 S \tag{3.9}$$

donde S es la matriz de covarianza muestral y h constante.

Estrechamente relacionado con la parametrización de ancho de banda, está la trans-

formación previa de los datos. En lugar de centrar nuestra atención en la selección de ancho de banda en los datos originales $X_1, \ldots, X_n$, podemos utilizar datos transformados $X_1^*, \ldots, X_n^*$. Los datos transformados X_i^* están más *alineados* con los ejes coordenados, lo que nos puede llevar a pensar que las parametrizaciones restringidas (matrices diagonales) pueden ser las más adecuadas. La transformación de datos se puede hacer de dos formas:

1. *Transformaciones esféricas:*

$$X_i^* = S^{-1/2} X_i, \tag{3.10}$$

donde S es la matriz de covarianza muestral. Luego la matriz de ancho de banda con los datos transformados de esta manera la denotamos por H^*. La relación entre H y H^* está dada por:

$$H = S^{1/2} H^* S^{1/2}.$$

La parametrización de H^* puede pensarse así:

Si $H^* = h^2 I$ entonces $H = (h^*)^2 S$.

Si $H^* = diag((h^*)_1^2, \ldots, (h^*)_d^2)$ entonces H es una matriz no-diagonal. Los elementos de H fuera de la diagonal tienen que ser construidos con la varianza muestral.

2. *Transformaciones por escalado:*

$$X_i^* = S_D^{-1/2} X_i, \tag{3.11}$$

donde $S_D = diag(s_1^2, \ldots, s_d^2)$ y $s_1^2, \ldots, s_d^2$ son las varianzas muestrales marginales. De forma análoga, la matriz de ancho de banda con los datos transformados de

esta manera la denotamos por H^*. La relación entre H y H^* está dada por:

$$H = S_D^{1/2} H^* S_D^{1/2}.$$

La parametrización de H^* puede pensarse así:

Si $H^* = h^2 I$ entonces $H = (h^*)^2 S_D$.

Si $H^* = diag((h^*)_1^2, \ldots, (h^*)_d^2)$ entonces $H^* = diag(s_1^2 h_1^{*2}, \ldots, s_d^2 h_d^{*2})$. Claramente H sigue siendo una matriz diagonal, por lo que no se ve ninguna ventaja sobre la matriz H sin pre-escalado.

Luego, debido a las limitaciones de combinar parametrizaciones de H con matrices diagonales y transformaciones de datos X^*, el desarrollo de esta tesis se centra en selecciones de matrices de ancho de banda tipo $\mathcal{F}$ (sin restricciones).

3.2. Estimación con una Mezcla de Distribuciones Gaussianas Multivaridas.

Una mezcla finita de distribuciones para una variable aleatoria $\mathbf{x}$ se define como la suma ponderada de componentes con densidades multivariadas:

$$f(\mathbf{x}, \alpha, \theta) = \sum_{i=1}^{M} \alpha_i f_i(\mathbf{x}; \theta_i) \tag{3.12}$$

donde el peso $\alpha_i > 0$, $\sum_{i=1}^{M} \alpha_i = 1$ y el componente f_i es una densidad con vector de parámetros d-dimensional $\theta_i = (\mu_i, \Sigma_i)$. Además se tiene que:

$$\alpha_{M\times 1} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{pmatrix} \qquad \mu_{M\times d} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_M \end{pmatrix} \qquad \mathbf{\Sigma}_{Md\times d} = \begin{pmatrix} \Sigma_1 \\ \vdots \\ \Sigma_M \end{pmatrix}$$

En nuestro trabajo de tesis, f_i es una distribución gaussiana d-dimensional dada por:

$$f_i(\mathbf{x};\theta_i) = \phi(\mathbf{x};\mu_i,\Sigma_i) = (2\pi)^{-\frac{d}{2}}|\Sigma_i|^{-\frac{1}{2}}e^{\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right)}. \tag{3.13}$$

Una mezcla con distribuciones gaussianas de variable aleatoria unidimensional $\mathbf{x}$ vendrá dada por:

$$f_i(\mathbf{x};\theta_i) = \alpha_1\phi(\mathbf{x};\mu_1,\sigma_1) + \cdots + \alpha_M\phi(\mathbf{x};\mu_M,\sigma_M), \tag{3.14}$$

donde $\phi(\mathbf{x};\mu_i,\sigma_i)$ denota una distribución gaussiana con media μ_i y desviación estándar σ_i, para $i = 1, ..., M$. En una mezcla univariada se deben estimar $3M$ parámetros (M-medias, M-varianzas y M-pesos). Como la suma de los pesos es 1, el número de parámetros a estimar se reduce a $3M - 1$.

El paquete **ks** del lenguaje de programación **R** permite simular una mezcla de distribuciones gaussianas. En la Figura 3.4 se presenta una mezcla de 4 distribuciones gaussianas unidimensionales simuladas con medias ubicadas en $-2, 0, 2, 4$, desviaciones estándar dadas por $0.5, 1.0, 1.5, 2.0$ y pesos $0.3, 0.25, 0.1, 0.35$ respectivamente.

Para problemas de mayor dimensión ($d > 1$), asumimos en la mezcla la variable aleatoria vectorial $\mathbf{x} = (x_1, ..., x_d)$ con las densidades componentes distribuidas como gaussianas d-dimensionales $\phi(\mathbf{x};\mu_i,\mathbf{\Sigma}_i)$, cuyo vector de medias es μ_i y matrices de varianzas y covarianzas $\mathbf{\Sigma}_i$, para $i = 1, \ldots, M$. En este tipo de mezcla se deben estimar

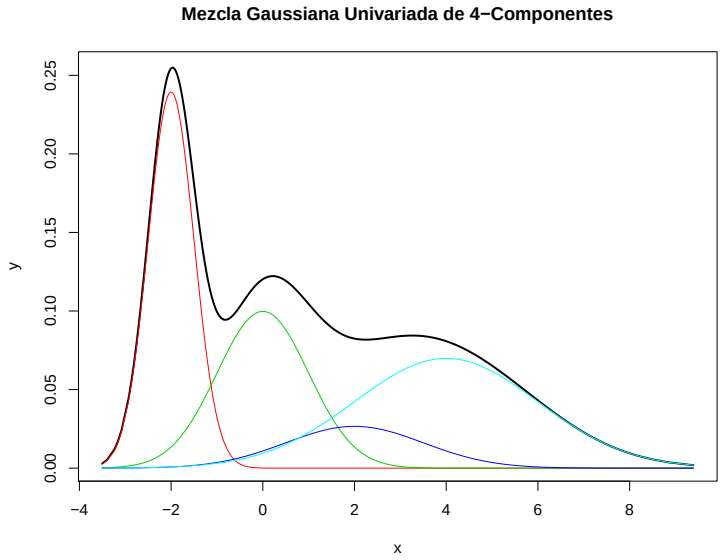

Figura 3.4: Mezcla de cuatro distribuciones gaussianas unidimensionales con medias ubicadas en $-2, 0, 2, 4$, desviaciones estándar dadas por $0.5, 1.0, 1.5, 2.0$ y pesos $0.3, 0.25, 0.1, 0.35$ respectivamente

$M-1$ valores para los pesos, $(d \times M)$ valores para los vectores medias y $\frac{d\times(d+1)}{2} \times M$ valores para los componentes de las matrices de covarianzas.

En la Figura 3.5 se muestra la gráfica generada con datos simulados de una mezcla de seis distribuciones gaussianas bivariadas con matrices de covarianzas iguales a la matriz identidad I, medias en $(-4, 0), (-2.5, 2.5), (-1, -1), (1, 1), (2.5, -2.5), (4, 0)$ y pesos $0.1, 0.18, 0.2, 0.23, 0.16, 0.13$ respectivamente.

Las mezclas han demostrado ser muy útiles en el modelado de densidades más complejas. Al usar un número pequeño de componentes gaussianas, se pueden modelar distribuciones que están lejos de ser gaussianas. Los métodos estándar para ajustar mezclas implican la selección de un número de componentes M y utilizan el algoritmo

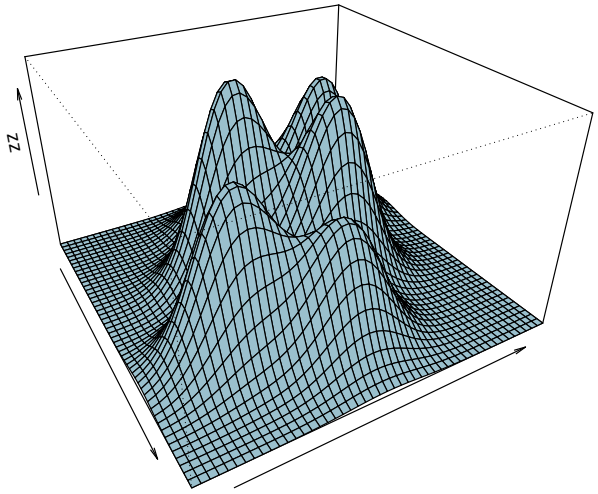

Figura 3.5: Mezcla de seis distribuciones gaussianas bivariadas con matrices de covarianzas I, medias en $(-4,0),(-2.5,2.5),(-1,-1),(1,1),(2.5,-2.5),(4,0)$ y pesos $0.1, 0.18, 0.2, 0.23, 0.16, 0.13$ respectivamente.

esperanza - maximización (EM) para estimar los parámetros.

3.2.1. Estimación de Parámetros en una Mezcla de Distribuciones Gaussianas.

Para observaciones que provienen de una mezcla de distribuciones gaussianas, es posible estimar conjuntamente los parámetros de las distribuciones que forman la mezcla y las *probabilidades a posteriori* de cada dato que pertenezca a cada una de las

componentes de la mezcla. Para ello se presentan los siguientes conceptos básicos en el proceso de estimación.

Ecuaciones de Máxima Verosimilitud para una Mezcla Gaussiana

Peña [27] describe en forma clara y resumida las ecuaciones de máxima verosimilitud para la mezcla d-dimensional de distribuciones gaussianas. Supongamos que se tienen n datos observados que provienen de una mezcla de distribuciones gaussianas

$$f(\mathbf{x};\alpha,\mu,\mathbf{\Sigma}) = \sum_{i=1}^{M} \alpha_i \phi(\mathbf{x};\mu_i,\mathbf{\Sigma}_i), \tag{3.15}$$

luego la función de verosimilitud estará dada por

$$l(\alpha,\mu,\mathbf{\Sigma}|\mathbf{x}) = \prod_{k=1}^{n} f(\mathbf{x}_k;\alpha,\mu,\mathbf{\Sigma}) = \prod_{k=1}^{n} \left(\sum_{i=1}^{M} \alpha_i \phi(\mathbf{x}_k;\mu_i,\mathbf{\Sigma}_i) \right). \tag{3.16}$$

Los estimadores de máxima verosimilitud para los parámetros $\alpha_i, \mu_i, \mathbf{\Sigma_i}$, son los valores $\hat{\alpha_i}, \hat{\mu_i}, \hat{\mathbf{\Sigma}}_i$ que maximizan la densidad de aparición de los valores observados y los cuales obtenemos al calcular el valor máximo de la función $l(\alpha,\mu,\mathbf{\Sigma}|X)$.

En la práctica suele ser más cómodo obtener el máximo del logaritmo de la función de verosimilitud:

$$\ln\left[l(\alpha,\mu,\mathbf{\Sigma}|\mathbf{x})\right] = \sum_{k=1}^{n} \ln\left[f(\mathbf{x}_k;\alpha,\mu,\mathbf{\Sigma})\right] = \sum_{k=1}^{n} \ln\left[\sum_{i=1}^{M} \alpha_i \phi(\mathbf{x}_k;\mu_i,\mathbf{\Sigma}_i)\right]. \tag{3.17}$$

Si sustituimos (3.13) en (3.17) se obtiene:

$$\ln\left[l(\alpha,\mu,\mathbf{\Sigma}|\mathbf{x})\right]=\sum_{k=1}^{n}\ln\left[\sum_{i=1}^{M}\alpha_i(2\pi)^{-\frac{d}{2}}|\mathbf{\Sigma}_i|^{-\frac{1}{2}}\ exp\left\{\left(-\frac{1}{2}\left(\mathbf{x}_k-\mu_i\right)^T\mathbf{\Sigma}_i^{-1}\left(\mathbf{x}_k-\mu_i\right)\right)\right\}\right]. \tag{3.18}$$

Moyano [21] muestra de forma detallada dicho proceso de maximización, el cual consiste en igualar a cero las derivadas de la ecuación (3.18) respecto a los parámetros α,μ,Σ. En este proceso surge la denominada *probabilidad a posteriori*, que es la probabilidad de que la k-ésima observación haya sido generada por la i-ésima población, con $k=1,\ldots,n$ para $i=1,\ldots,M$. Esta probabilidad está dada por:

$$\tau_{ki}=\frac{\alpha_i\phi(\mathbf{x};\mu_i,\mathbf{\Sigma}_i)}{\sum_{i=1}^{M}\alpha_i\phi(\mathbf{x};\mu_i,\mathbf{\Sigma}_i)} \tag{3.19}$$

Luego los estimadores de máxima verosimilitud de los parámetros en una mezcla gaussiana d-dimensional están dados por:

$$\hat{\alpha}_i=\frac{1}{n}\sum_{k=1}^{n}\tau_{ki} \tag{3.20}$$

$$\hat{\mu}_i=\frac{1}{\sum_{k=1}^{n}\tau_{ki}}\sum_{k=1}^{n}\tau_{ki}(\mathbf{x}_k) \tag{3.21}$$

$$\hat{\Sigma}_i=\frac{1}{\sum_{k=1}^{n}\tau_{ki}}\sum_{k=1}^{n}\tau_{ki}(x_k-\hat{\mu}_i)(x_k-\hat{\mu}_i)^T \tag{3.22}$$

Para resolver las ecuaciones (3.20), (3.21) y (3.22) y así obtener los estimadores, se necesitan las probabilidades τ_{ki} dadas en (3.19), y éstas a su vez necesitan los parámetros del modelo. Entonces, tenemos el problema de estimar parámetros los cuales nece-

sitamos conocer previamente para estimarlos. Una solución a ese problema se presenta a continuación en el *Algoritmo EM.*

Generalidades del Algoritmo (EM)

El algoritmo (EM) fue descrito por Dempster, Laird y Rubin [1]. Es un método de optimización iterativo, compuesto de dos pasos alternados que involucran el cálculo de una esperanza y una maximización. Este método es usado para estimar parámetros desconocidos en la función de máxima verosimilitud de un conjunto de datos muestrales (estimación que puede ser un problema intratable analíticamente) y se ha convertido en una herramienta ampliamente utilizada por investigadores en las diferentes áreas, ya que permite estimar datos faltantes o ausentes en diversos problemas multivariados donde los algoritmos basados en los métodos iterativos de Newton pueden resultar más complicados.

El algoritmo (EM) simplifica la estimación de los parámetros en un modelo estadístico al caracterizar el conjunto de datos Y, en *datos observados* X y *datos ausentes* Z. Luego si asumimos una muestra aleatoria particionada $\mathrm{Y} = (\mathrm{X}, \mathrm{Z})$, para la cual se establece una función de distribución condicionada a los parámetros θ, tenemos la siguiente relación bayesiana:

$$f(\mathrm{X}, \mathrm{Z}|\theta) = f(\mathrm{Z}|\mathrm{X}, \theta) f(\mathrm{X}|\theta),$$

entonces:

$$f(\mathrm{X}|\theta) = \frac{f(\mathrm{X},\mathrm{Z}|\theta)}{f(\mathrm{Z}|\mathrm{X},\theta)}.$$

Por lo tanto, podemos decir que la función de verosimilitud para los datos observados X será:

$$l(\theta|\mathrm{X}) = \frac{l(\theta|\mathrm{X},\mathrm{Z})}{l(\theta,\mathrm{Z}|\mathrm{X})} \tag{3.23}$$

donde $l(\theta|\mathrm{X},\mathrm{Z})$ es la verosimilitud para toda la muestra y $l(\theta,\mathrm{Z}|\mathrm{X})$ es la verosimilitud de los datos faltantes dados los datos observados. Operando con el logaritmo natural a ambos lados en la ecuación (3.23) se obtiene:

$$\ln\left[l(\theta|\mathrm{X})\right] = \ln\left[l(\theta|\mathrm{X},\mathrm{Z})\right] - \ln\left[l(\theta,\mathrm{Z}|\mathrm{X})\right], \tag{3.24}$$

Si asumimos que los datos faltantes fueron introducidos para simplificar la estimación de los parámetros desconocidos, nuestro interés radica en maximizar la función dada por $\ln\left[l(\theta|\mathrm{X})\right]$ en (3.24). Por lo tanto necesitamos que en la diferencia dada en (3.23), la función con la muestra completa $\ln\left[l(\theta|\mathrm{X},\mathrm{Z})\right]$ sea máxima. En la práctica, la maximización de $\ln\left[l(\theta|\mathrm{X},\mathrm{Z})\right]$ es mas fácil de realizar que la maximización de los datos observados $\ln\left[l(\theta|\mathrm{X})\right]$. Es ésta la razón por la cual el algoritmo (EM) usa la función $\ln\left[l(\theta|\mathrm{X},\mathrm{Z})\right]$ como función de verosimilitud en la búsqueda del estimador máximo verosímil de θ.

Así la estructura funcional del algoritmo (EM) es:

- Partir de un estimador inicial $\widehat{\theta}^{[0]}$ y estipular un margen de error o tolerancia *tol* para los valores de los parámetros estimados.

- Iniciar un contador con $t = 0$.
- Hacer $t = t + 1$.
- **Paso E:**

  Usar la estimación actual de θ, es decir $\widehat{\theta}^{[t-1]}$, para calcular la esperanza de $\ln[l(\theta|\mathrm{X}, \mathrm{Z})]$ con respecto a la distribución de los valores ausentes Z, dados los parámetros $\widehat{\theta}^{[t-1]}$ y los datos observados X. Esto nos dará una nueva verosimilitud que denominaremos $l^*(\theta|\mathrm{X})$, es decir:

$$\begin{aligned} l^*(\theta|\mathrm{X}) &= E_{Z|\widehat{\theta}^{[t-1]}}\Big\{\ln[l(\theta|\mathrm{X}, \mathrm{Z})]\Big\} \\ &= E_{Z|\widehat{\theta}^{[t-1]}}\Big\{\sum_{k=1}^{n}\ln[f(x_k;\theta)]\Big\} \\ &= \sum_{k=1}^{n} E_{Z|\widehat{\theta}^{[t-1]}}\Big\{\ln[f(x_k;\theta)]\Big\} \end{aligned} \tag{3.25}$$

- **Paso M:**

  Maximizar $l^*(\theta|\mathrm{X})$ con respecto al vector de variables θ. Llamaremos a este nuevo vector de parámetros $\widehat{\theta}^{[t]}$. Es decir:

$$\widehat{\theta}^{[t]} = \max_{\theta}\Big[l^*(\theta|\mathrm{X})\Big]. \tag{3.26}$$

- Se verifica que $\|\widehat{\theta}^{[t]} - \widehat{\theta}^{[t-1]}\| < tol$:

  Si la diferencia es suficientemente pequeña, nuestro estimador máximo verosímil de los parámetros es $\widehat{\theta}^{[t]}$. Si la diferencia no es suficientemente pequeña, se retorna al paso $(t = t+1)$, incrementando así este contador. Luego, seguimos con el **paso E** y repetimos el proceso hasta lograr la convergencia.

Dempster, Laird y Rubin [1] demuestran que el algoritmo converge y Peña [27] en el apéndice 11.1 demuestra que los valores estimados para mezclas gaussianas con este algoritmo son realmente estimadores máximo verosímiles para la muestra.

El algoritmo (EM) en la estimación de parámetros de mezclas gaussianas

El algoritmo (EM) es una herramienta eficaz para estimar los parámetros $\alpha_i, \mu_i, \Sigma_i$ de una mezcla gaussiana d-dimensional, con $i = 1, \ldots, M$. Para esto introducimos un conjunto de *variables no observadas* $\mathrm{Z} = (\mathrm{z_1}, \ldots, \mathrm{z_n})$, donde cada $\mathrm{z_k} = (z_{k1}, \ldots, z_{kM})$ con $k = 1, \ldots, n$. Estos vectores tienen como propósito indicar de cuál componente de la mezcla proviene cada observación. Luego, cada individuo k tiene M posibilidades:

$$\text{Pertenecer al grupo } 1 \Longrightarrow \mathrm{z_k} = (1, 0, 0, \ldots, 0)_{1\times M}$$

$$\text{Pertenecer al grupo } 2 \Longrightarrow \mathrm{z_k} = (0, 1, 0, \ldots, 0)_{1\times M}$$

$$\vdots$$

$$\text{Pertenecer al grupo } M \Longrightarrow \mathrm{z_k} = (0, 0, 0, \ldots, 1)_{1\times M}$$

Por lo tanto los datos muestrales quedan definidos como: $\mathrm{Y} = (\mathrm{X}, \mathrm{Z})$ donde X representa los datos observados y Z la procedencia de estos (datos no observados).

Según la definición de distribución condicionada, tenemos el valor para el k-ésimo individuo en una mezcla gaussiana dado por:

$$f(y_k) = f(x_k, z_k) = f(x_k|z_k)f(z_k), \tag{3.27}$$

Ahora realizamos un proceso similar al presentado en la sección *generalidades del algoritmo (EM)*. Peña [27] describe la función de densidad de x_i condicionada a z_i como:

$$f(x_k|z_k) = \prod_{i=1}^{M} [\phi(x_k; \mu_i, \mathbf{\Sigma}_i)]^{z_{ki}}, \tag{3.28}$$

En el vector z_k sólo una componente z_{ki} es distinta de cero y esta componente definirá cuál es la función de densidad de procedencia de las observaciones. Análogamente, la función de probabilidades de la variable z_k será:

$$f(z_k) = \prod_{i=1}^{M} [\alpha_i]^{z_{ki}}. \tag{3.29}$$

Entonces, por (3.28) y (3.29), se tiene que la función de densidad conjunta viene dada por

$$f(x_k, z_k) = \prod_{i=1}^{M} [\alpha_i \phi(x_k; \mu_i, \mathbf{\Sigma}_i)]^{z_{ki}}, \tag{3.30}$$

Luego al aplicar el logaritmo natural se obtiene el siguiente resultado, descrito en forma detallada por Moyano [21].

$$\begin{aligned} \ln[l(\alpha, \mu, \Sigma|X, Z)] &= \sum_{k=1}^{n} \ln\Big\{ \prod_{i=1}^{M} [\alpha_i \phi(x_k; \mu_i, \mathbf{\Sigma}_i)]^{z_{ki}} \Big\} \\ &= \sum_{k=1}^{n} \sum_{i=1}^{M} z_{ki} \ln[\alpha_i] + \sum_{k=1}^{n} \sum_{i=1}^{M} z_{ki} \ln[\phi(x_k; \mu_i, \mathbf{\Sigma}_i)]. \end{aligned} \tag{3.31}$$

La función hallada en (3.31) es precisamente la implementada por el algoritmo

(EM) para realizar sus operaciones internas. Entonces el algoritmo (EM) para mezclas está dado por:

- **Partir de unos estimadores iniciales $\widehat{\alpha}_i^{[0]}, \widehat{\mu}_i^{[0]}, \widehat{\mathbf{\Sigma}}_i^{[0]}$ y determinar una tolerancia (tol).**

  Estos estimadores iniciales son establecidos usando algún método de clasificación multivariada, o por información *a priori* de la muestra. La tolerancia (tol) se establece lo suficientemente pequeña para garantizar la precisión de la estimación.

- **Iniciar un contador con $t = 0$.**

- **Hacer $t = t + 1$.**

- **Paso E:** Hallamos la verosimilitud

$$l^*(\alpha, \mu, \mathbf{\Sigma}|\mathrm{X}) = E_{Z|\widehat{\theta}^{[t-1]}}\Big\{\ln[l(\alpha, \mu, \mathbf{\Sigma}|X, Z)]\Big\} = \sum_{k=1}^{n} E_{Z|\widehat{\theta}^{[t-1]}}\Big\{\ln[f(x_k, z_k; \alpha, \mu, \mathbf{\Sigma})]\Big\}. \tag{3.32}$$

Desarrollando operaciones adecuadas, se obtiene que:

$$l^*(\alpha, \mu, \mathbf{\Sigma}|\mathrm{Y}) = \sum_{k=1}^{n}\sum_{i=1}^{M} \widehat{\tau}_{ki}^{[t]} \ln[\alpha_i] + \sum_{k=1}^{n}\sum_{i=1}^{M} \widehat{\tau}_{ki}^{[t]} \ln\big[\phi(x_k; \mu_i, \mathbf{\Sigma}_i)\big], \tag{3.33}$$

donde $\widehat{\tau}_{ki}^{[t]}$ es la probabilidad a posteriori de que la k-ésima observación haya sido generada por la i-ésima población en la iteración o tiempo t. Por la ecuación (3.19), $\widehat{\tau}_{ki}^{[t]}$ está dada por:

$$\boxed{\hat{\tau}_{ki}^{[t]} = \frac{\hat{\alpha}_i^{[t-1]}\phi(x_k; \mu_i, \mathbf{\Sigma}_i)}{\sum_{i=1}^{M} \hat{\alpha}_i^{[t-1]}\phi(x_k; \mu_i, \mathbf{\Sigma}_i)}.} \tag{3.34}$$

- **Paso M:**

$$(\hat{\alpha}, \hat{\mu}, \hat{\mathbf{\Sigma}})^{[t]} = \max_{(\alpha,\mu,\mathbf{\Sigma})} \Big\{ l^*(\theta|\mathrm{Y}) \Big\}$$
$$= \max_{(\alpha,\mu,\mathbf{\Sigma})} \left\{ \sum_{k=1}^{n} \sum_{i=1}^{M} \widehat{\tau}_{ki}^{[t]} \ln[\alpha_i] + \sum_{k=1}^{n} \sum_{i=1}^{M} \widehat{\tau}_{ki}^{[t]} \ln\big[\phi(x_k; \mu_i, \mathbf{\Sigma}_i)\big] \right\} \quad (3.35)$$

Realizando las derivadas adecuadas respecto a $\alpha_i, \mu_i, \mathbf{\Sigma}_i$ para $i = 1, \ldots, M$ (ver Moyano [21]), y según las ecuaciones (3.20), (3.21) y (3.22), se tienen los siguientes estimadores de máxima verosimilitud evaluados en el tiempo t, para los parámetros de una mezcla gaussiana d-dimensional:

$$\boxed{\hat{\alpha}_i^{[t]} = \frac{1}{n} \sum_{k=1}^{n} \hat{\tau}_{ki}^{[t]}} \quad (3.36)$$

$$\boxed{\hat{\mu}_i^{[t]} = \frac{1}{\sum_{k=1}^{n} \hat{\tau}_{ki}^{[t]}} \sum_{k=1}^{n} \hat{\tau}_{ki}^{[t]} * (x_k)} \quad (3.37)$$

$$\boxed{\hat{\mathbf{\Sigma}}_i^{[t]} = \frac{1}{\sum_{k=1}^{n} \hat{\tau}_{ki}^{[t]}} \sum_{k=1}^{n} \hat{\tau}_{ki}^{[t]} * (x_k - \hat{\mu}_i)(x_k - \hat{\mu}_i)^T} \quad (3.38)$$

En resumen, en el **paso M** hallamos:

$$(\hat{\alpha}, \hat{\mu}, \hat{\mathbf{\Sigma}})^{[t]} = \big(\hat{\alpha}_1^{[t]}, \ldots, \hat{\alpha}_M^{[t]} \ ; \ \hat{\mu}_1^{[t]}, \ldots, \hat{\mu}_M^{[t]} \ ; \ \hat{\mathbf{\Sigma}}_1^{[t]}, \ldots, \hat{\mathbf{\Sigma}}_M^{[t]}\big),$$

- **Se evalúa** $\|(\hat{\alpha}, \hat{\mu}, \hat{\mathbf{\Sigma}})^{[t]} - (\hat{\alpha}, \hat{\mu}, \hat{\mathbf{\Sigma}})^{[t-1]}\| < tol$.

Si la comparación es cierta, nuestro estimador máximo verosímil de los parámetros es $(\hat{\alpha}, \hat{\mu}, \hat{\mathbf{\Sigma}})^{[t]}$. Si la diferencia no es suficientemente pequeña, se retorna al

paso ($\mathbf{t} = \mathbf{t} + \mathbf{1}$) y se incrementa este contador, seguimos con el **paso E** y repetimos el proceso hasta lograr la convergencia.

Martínez y Martínez [40] enumeran los siguientes inconvenientes para el buen funcionamiento del algoritmo (EM) en mezclas gaussianas:

- Puede converger a un óptimo local.
- Puede no converger.
- Requiere unos valores iniciales de los parámetros en las densidades componentes $\phi(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i)$.
- Necesita un estimado del número de componentes M.

Fraley y Raftery [4] muestran cómo resolver estos problemas con un enfoque que modela los datos basados en conglomerados (clusters) y presentan un paquete en **R** llamado **MCLUST** el cual permite analizar estos grupos para diversos modelos, combinando la agrupación jerárquica (HC), el algoritmo (EM) para modelos de mezcla de Gaussianas y el criterio de información bayesiana (BIC) para la selección del mejor modelo. Además, debido a que la matriz de covarianzas $\mathbf{\Sigma}_i$ proporciona las características geométricas más importantes de las componentes $\phi(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i)$, **MCLUST** establece algunas parametrizaciones de la matriz de covarianza $\mathbf{\Sigma}_i$ basándose en agrupamientos jerárquicos (HC) y el algoritmo (EM) en los datos.

3.3. Medidas de Similaridad entre dos Mezclas de Gaussianas Multivariadas.

Dadas dos mezclas Gaussianas, $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$, el objetivo aquí es cuantificar la cercanía (similitud) de $\phi_1(\mathbf{x})$ a $\phi_2(\mathbf{x})$ o viceversa. En teoría de la probabilidad se estudian, entre otras, *la divergencia Kullback - Leibler (KL)* y *la distancia Hellinger* para medir la distancia entre dos distribuciones probabilísticas. Para el caso de mezclas gaussianas no existe forma analítica explícita de calcular estas medidas de similaridad. La solución a este problema surge con Julier y Uhlmann [34], quienes describen *the unscented transform UT*, que es un método para calcular las estadísticas de una variable aleatoria que sufre una transformación no lineal. Goldberger, Gordon y Greenspan [8] demostraron que UT se puede utilizar para obtener una buena aproximación de la divergencia-KL entre dos mezclas de distribuciones gaussianas d-dimensionales. Kristan, Leonardis y Skocaj [16] proponen una aproximación a la distancia Hellinger multivariada sobre las mezclas de gaussianas usando UT. Estos conceptos se describen a continuación.

3.3.1. The *Unscented* Transform UT

The unscented transform UT [34] es un método para calcular el vector de medias y la matriz de covarianzas de una variable aleatoria que se somete a una transformación no lineal. Considérese una variable aleatoria $\mathbf{x}$ (d-dimensional) y la función no lineal, $\mathbf{y} = g(\mathbf{x})$, con $\mathbf{y} \in \mathbb{R}^{d'}$ y $d' \leq d$. Asumamos que $\mathbf{x}$ tiene vector de medias μ_x y matriz de covarianza $\mathbf{\Sigma}_x$. Para el cálculo de las estadísticas de $\mathbf{y}$, se forma un vector $\mathcal{X}$ de $(2d+1)$ filas correspondientes a los sigmas vectores o sigma puntos $\mathcal{X}_i$ y un vector con

las ponderaciones correspondientes $\mathcal{W}_i$, de la siguiente manera:

$$\mathcal{X}_i = \begin{cases} \mu_x & i = 0 \\ \mathcal{X}_0 + \left[\left(\sqrt{d+\kappa}\right)\left(\sqrt{\mathbf{\Sigma}_x}\right)_i\right] & i = 1, \ldots, d \\ \mathcal{X}_0 - \left[\left(\sqrt{d+\kappa}\right)\left(\sqrt{\mathbf{\Sigma}_x}\right)_{i-d}\right] & i = (d+1), \ldots, 2d \end{cases}$$

$$\mathcal{W}_i = \begin{cases} \frac{\kappa}{d+\kappa} & i = 0 \\ \frac{1}{2(d+\kappa)} & i = 1, \ldots, d, \ldots, 2d \end{cases}$$

donde $\kappa \in \mathbb{R}$ es un parámetro de escala que designa la dirección de escalado de los sigma puntos $\mathcal{X}_i$ y $\left(\sqrt{\mathbf{\Sigma}_x}\right)_i$ es la i-ésima columna de la raíz cuadrada de la matriz $\mathbf{\Sigma}_x$. Por la ecuación (2.4), $\mathbf{\Sigma}_x = \mathbf{UDU}^T$ es la descomposición de valores singulares (espectral) con vectores propios $\mathbf{u} = \{\mathbf{u}_1, \ldots, \mathbf{u}_d\}$ y matriz diagonal con valores propios $D = diag\{\lambda_1, \ldots \lambda_d\}$, entonces $\left(\sqrt{\mathbf{\Sigma}_x}\right)_i = \sqrt{\lambda_i}\mathbf{u}_i$.

Luego, las estimaciones del vector de medias y la matriz de covarianza de $\mathbf{y}$ pueden ser aproximadas utilizando, respectivamente, el vector de medias muestral ponderado y la matriz de covarianzas ponderada en función de los sigma puntos:

$$\hat{\mu}_y = E[g(\mathbf{x})] = \bar{\mathbf{y}} = \sum_{i=0}^{2d} g(\mathcal{X}_i)\mathcal{W}_i \tag{3.39}$$

$$\hat{\mathbf{\Sigma}}_y = \sum_{i=0}^{2d} \{g(\mathcal{X}_i) - \bar{\mathbf{y}}\}\{g(\mathcal{X}_i) - \bar{\mathbf{y}}\}^T \mathcal{W}_i \tag{3.40}$$

Si $\mathbf{x}$ se distribuye en forma Gaussiana d-dimensional, entonces $\kappa = \text{máx}(0, 3 - d)$.

3.3.2. Divergencia Kullback – Leibler KL

Definición 3.3. *Sea (X, μ) un espacio con una medida finita, no negativa y no singular. Para dos distribuciones $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$ de variable aleatoria continua, la divergencia de KL se define como:*

$$D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \int_X f_1(x) \ln \left| \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \right| d\mu \tag{3.41}$$

Cuando la medida μ es *la medida de Lebesgue* sobre el eje real, resulta:

$$D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \int_{-\infty}^{\infty} f_1(\mathbf{x}) \ln |f_1(\mathbf{x})| \, d\mathbf{x} - \int_{-\infty}^{\infty} f_1(x) \ln |f_2(\mathbf{x})| \, d\mathbf{x} \tag{3.42}$$

La divergencia KL tiene las siguientes propiedades.

1. $D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) \geq 0$.
2. $D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) = 0 \Leftrightarrow f_1(\mathbf{x}) = f_2(\mathbf{x})$.
3. $D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) = E_{f_1}\left[\ln \left| \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \right|\right]$
4. $D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x}))$ no es simétrica para $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$, luego no es una "distancia" en el sentido matemático de la palabra.

3.3.3. Divergencia – KL entre dos Gaussianas Multivaridas.

Proposición 3.1. *Dadas dos distribuciones gaussianas d-dimensionales $\phi(\mathbf{x}; \mu_1, \mathbf{\Sigma}_1)$ y $\phi(\mathbf{x}; \mu_2, \mathbf{\Sigma}_2)$, la divergencia KL se expresa por:*

$$D_{KL}(\phi_{\mathbf{\Sigma}_1}(\mathbf{x};\mu_1),\phi_{\mathbf{\Sigma}_2}(\mathbf{x};\mu_2)) = \frac{1}{2}\left[\ln\frac{|\mathbf{\Sigma}_2|}{|\mathbf{\Sigma}_1|} + traza\left(\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1\right) + (\mu_1-\mu_2)^T\mathbf{\Sigma}_2^{-1}(\mu_1-\mu_2) - d\right] \quad (3.43)$$

Demostración. Ver Davis y Dhillon [12]. ■

3.3.4. Aproximación de la divergencia KL usando UT

Goldberger, Gordon y Greenspan [9] utilizan UT para aproximar de la divergencia KL entre dos mezclas de distribuciones gaussianas d-dimensionales. El proceso se describe a continuación. Sean $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$ dos mezclas de distribuciones gaussianas d-dimensionales, con componentes gaussianas $f_{1,i}$ y $f_{2,j}$ tal que:

$$f_1(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i f_{1,i} \qquad y \qquad f_2(\mathbf{x}) = \sum_{j=1}^{m}\beta_j f_{2,j} \quad (3.44)$$

Puesto que

$$D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \int f_1 \ln|f_1| d\mathbf{x} - \int f_1 \ln|f_2| d\mathbf{x}, \quad (3.45)$$

se muestra cómo hacer la aproximación para $\int f_1 \ln|f_2|$ usando UT. La linealidad de la construcción de $f_1(\mathbf{x})$ para sus componentes produce:

$$\int f_1 \ln|f_2| d\mathbf{x} = \sum_{i=1}^{n}\alpha_i \int f_{1,i}\ln|f_2| = \sum_{i=1}^{n}\alpha_i \underbrace{E_{f_{1,i}}(\ln|f_2|)}_{\mu} \quad (3.46)$$

Ahora, como $E_{f_{1,i}}(\mathbf{x}) = \mu_i$ y $E_{f_{1,i}}(\ln f_2(\mathbf{x}))$ es la media de la variable aleatoria $\ln f_2(\mathbf{x})$, que no es una función lineal de $\mathbf{x}$, entonces esta media puede ser aproximada usando UT así:

$$\int f_1 \ln(f_2) d\mathbf{x} \approx \sum_{i=1}^{n} \alpha_i \sum_{j=0}^{2d} \ln\left[f_2(\mathcal{X}_{ij})\right] \mathcal{W}_{ij}$$

donde $\{\mathcal{X}_{ij}, \mathcal{W}_{ij}\}_{j=0,\ldots,2d}$ es el conjunto ponderado de sigma puntos correspondiente a la i-ésima componente de f_1, el cual se define como:

$$\mathcal{X}_i = \begin{cases} \mu_x & i = 0 \\ \mathcal{X}_0 + \left[\left(\sqrt{d}\right)\left(\sqrt{\mathbf{\Sigma}_x}\right)_i\right] & i = 1, \ldots, d \\ \mathcal{X}_0 - \left[\left(\sqrt{d}\right)\left(\sqrt{\mathbf{\Sigma}_x}\right)_{i-d}\right] & i = (d+1), \ldots, 2d \end{cases}$$

$$\mathcal{W}_i = \begin{cases} 0 & i = 0 \\ \frac{1}{2d} & i = 1, \ldots, d, \ldots, 2d \end{cases}$$

Estas ecuaciones son similares a las vistas en la sección 3.3.1, pero en este caso Goldberger, Gordon y Greenspan [8] asumen $\kappa = 0$. Luego se tiene la siguiente aproximación:

$$\int f_1 \ln(f_2) d\mathbf{x} \approx \frac{1}{2d} \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{2d} \ln\left[f_2(\mathcal{X}_{ij})\right] \tag{3.47}$$

De forma similar realizamos la aproximación para $\int f_1 \ln(f_1)$ usando UT, obteniendo así:

$$\int f_1 \ln(f_1) \approx \frac{1}{2d} \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{2d} \ln\left[f_1(\mathcal{X}_{ij})\right] \tag{3.48}$$

Reemplazamos los resultados dados en (3.47) y (3.48) en (3.45). Luego escribimos la aproximación a la divergencia KL usando UT como:

$$D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) \approx \frac{1}{2d} \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{2d} \left\{\ln\left[f_1(\mathcal{X}_{ij})\right] - \ln\left[f_2(\mathcal{X}_{ij})\right]\right\} \tag{3.49}$$

El **algoritmo KL-UT** resume estas ideas.

Algoritmo 3.1 Algoritmo KL-UT

Entrada: $f_1(x)$ y $f_2(x)$.

1: Hallar $\{\mathcal{X}_{ij}\}_{j=1,\ldots,2d}$ para cada componente de f_1 con $i = 1, \ldots, n$, donde:

$$\mathcal{X}_{ij} = \mu_i + \left[\left(\sqrt{d}\right)\left(\sqrt{\mathbf{\Sigma}_x}\right)_i\right] \quad j = 1, \ldots, d$$
$$\mathcal{X}_{ij} = \mu_i - \left[\left(\sqrt{d}\right)\left(\sqrt{\mathbf{\Sigma}_x}\right)_i\right] \quad j = (d+1), \ldots, 2d$$

2: Calcular:

$$D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \frac{1}{2d} \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{2d} \left|\ln\left[f_1(\mathcal{X}_{ij})\right] - \ln\left[f_2(\mathcal{X}_{ij})\right]\right|$$

Salida: La aproximación de la divergencia KL usando UT:

$$D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x}))$$

3.3.5. Distancia Hellinger

Definición 3.4. *La **Integral Hellinger** es una integral del tipo Riemann para un conjunto de funciones $\{f_\alpha\}$. Sea (X, μ) es un espacio con una medida finita, no negativa*

y no singular. Si $f(E)$, $E \subset X$, es una función totalmente aditiva con $f(E) = 0$ para $\mu(E) = 0$, y si $\delta = \{E_n\}_{n=1}^{N}$ es una partición de X, entonces

$$S_\delta = \sum_{n=1}^{N} \frac{f^2(E_n)}{\mu(E_n)},$$

y la integral de Hellinger de $f(E)$ con respecto a X se define como:

$$\int_X \frac{f^2(dE)}{d\mu} = \sup_\delta S_\delta$$

siempre y cuando este supremo sea finito.

Ahora, cuando $\phi : X \to R$ es una función aditiva de tal manera que $f(E)$ es la integral de Lebesgue $\int_E \phi \, d\mu$, entonces la integral Hellinger se puede expresar en términos de la integral de Lebesgue:

$$\int_X \frac{f^2(dE)}{d\mu} = \int_X \phi^2 d\mu.$$

Definición 3.5. *Sean $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$ dos distribuciones de probabilidad. El cuadrado de* ***la distancia Hellinger*** *se puede expresar como una norma de cálculo integral así:*

$$D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \frac{1}{2} \int \left(\sqrt{f_1(\mathbf{x})} - \sqrt{f_2(\mathbf{x})} \right)^2 dx. \tag{3.50}$$

La distancia Hellinger se utiliza para cuantificar la similaridad entre dos distribuciones de probabilidad y tiene las siguientes propiedades:

1. $0 \leq D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) \leq 1$.

2. $D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) = 1 \Leftrightarrow f_1(\mathbf{x}) \wedge f_2(\mathbf{x})$ son mutuamente singulares;

3. $D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) = 0 \Leftrightarrow f_1(\mathbf{x}) = f_2(\mathbf{x})$.

La divergencia - KL y la distancia Hellinger satisfacen la relación:

$$D_{KL}(f_1(\mathbf{x}), f_2(\mathbf{x})) \geq \frac{2}{\ln 2} D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) \tag{3.51}$$

3.3.6. Distancia Hellinger Ponderada UT

Kristan, Leonardis y Skocaj [16] proponen la *Distancia Hellinger Ponderada* la cual describimos a continuación. Sean $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$ dos mezclas de distribuciones gaussianas d-dimensionales, de componentes gaussianas $\phi(x; \mu_i, \mathbf{\Sigma}_i)$ y $\phi(x; \mu_j, \mathbf{\Sigma}_j)$ así:

$$f_1(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \phi(x; \mu_i, \mathbf{\Sigma}_i) \qquad y \qquad f_2(\mathbf{x}) = \sum_{j=1}^{m} \beta_j \phi(x; \mu_j, \mathbf{\Sigma}_j), \tag{3.52}$$

luego definimos una *distribución de importancia* dada por

$$f_0(\mathbf{x}) = f_0(f_1(\mathbf{x}) + f_2(\mathbf{x})) = \sum_{k=1}^{n+m} w_k \phi(x; \mu_k, \mathbf{\Sigma}_k). \tag{3.53}$$

donde $k \in \{1, \ldots, n, (n+1), \ldots, (n+m)\}$, $\sum_{k=1}^{n+m} w_k = 1$ y w_k son los pesos ponderados de los pesos α_i y β_j.

La distancia de Hellinger definida en (3.50) puede ser reescrita como:

$$D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \frac{1}{2}\int \frac{f_0(\mathbf{x})\left(\sqrt{f_1(\mathbf{x})} - \sqrt{f_2(\mathbf{x})}\right)^2}{f_0(\mathbf{x})} d\mathbf{x} = \frac{1}{2}\int g(\mathbf{x}) f_0(\mathbf{x}) d\mathbf{x} \tag{3.54}$$

donde

$$g(\mathbf{x}) = \frac{\left(\sqrt{f_1(\mathbf{x})} - \sqrt{f_2(\mathbf{x})}\right)^2}{f_0(\mathbf{x})} = \left(\frac{\sqrt{f_1(\mathbf{x})} - \sqrt{f_2(\mathbf{x})}}{\sqrt{f_0(\mathbf{x})}}\right)^2 = \left(\frac{\sqrt{f_1(\mathbf{x})}}{\sqrt{f_0(\mathbf{x})}} - \frac{\sqrt{f_2(\mathbf{x})}}{\sqrt{f_0(\mathbf{x})}}\right)^2 .$$

Luego sustituyendo (3.53) en(3.54)se tiene que:

$$D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \frac{1}{2}\sum_{k=1}^{n+m} w_k \underbrace{\int g(\mathbf{x})\phi(\mathbf{x}; \mu_k, \mathbf{\Sigma}_k) dx}_{E_{\phi_k}[g(\mathbf{x})]} . \tag{3.55}$$

La integral dada en (3.55) es la esperanza de una transformación no-lineal sobre una variable aleatoria $\mathbf{x}$ de tipo gaussiana d-dimensional, lo cual admite *the unscented transform*. Luego por el resultado dado en (3.39) se tiene que

$$D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) \approx \frac{1}{2}\sum_{k=1}^{n+m} w_k \sum_{j=0}^{2d} g(\mathcal{X}_{kj})\mathcal{W}_{kj} \tag{3.56}$$

donde $\{\mathcal{X}_{kj}, \mathcal{W}_{kj}\}_{j=1,\ldots,2d}$ es el conjunto ponderado de sigma puntos correspondiente a la i-ésima gaussiana $\phi(\mathbf{x}; \mu_k, \mathbf{\Sigma}_k)$, el cual se define como:

$$\mathcal{X}_{ij} = \begin{cases} \mu_k & j = 0 \\ \mathcal{X}_{k0} + \left[\left(\sqrt{d+\kappa} \right) \left(\sqrt{\mathbf{\Sigma}_k} \right)_j \right] & j = 1, \ldots, d \\ \mathcal{X}_{k0} - \left[\left(\sqrt{d+\kappa} \right) \left(\sqrt{\mathbf{\Sigma}_k} \right)_{j-d} \right] & j = (d+1), \ldots, 2d \end{cases}$$

$$\mathcal{W}_j = \begin{cases} \frac{\kappa}{d+\kappa} & j = 0 \\ \frac{1}{2(d+\kappa)} & j = 1, \ldots, d, \ldots, 2d \end{cases}$$

Aquí, $\kappa = \text{máx}(0, 3-d)$ y $\left(\sqrt{\mathbf{\Sigma}_k}\right)_j$ es la j-ésima columna de la raíz cuadrada de la matriz $\mathbf{\Sigma}_k$.

El **Algoritmo HD-UT** resume estas ideas

Algoritmo 3.2 Algoritmo HD-UT

Entrada: $f_1(x)$ y $f_2(x)$.

1: Sea $\kappa = \text{máx}(0, 3-d)$

2: Hallar $\{\mathcal{X}_{kj}, \mathcal{W}_{kj}\}_{j=1,\ldots,2d}$ para cada componente de $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$ donde:

$$\begin{aligned} &\mathcal{X}_{k0} = \mu_k && \mathcal{W}_{k0} = \tfrac{\kappa}{d+\kappa} && j = 0 \\ &\mathcal{X}_{kj} = \mathcal{X}_{k0} + \left[\left(\sqrt{d+\kappa} \right) \left(\sqrt{\mathbf{\Sigma}_k} \right)_j \right] && \mathcal{W}_{kj} = \tfrac{1}{2(d+\kappa)} && j = 1, \ldots, d, (d+1), \ldots, 2d. \\ &\mathcal{X}_{kj} = \mathcal{X}_{k0} - \left[\left(\sqrt{d+\kappa} \right) \left(\sqrt{\mathbf{\Sigma}_k} \right)_j \right] \end{aligned}$$

Aquí se tiene: $k = 1, \ldots, n, (n+1), \ldots (n+m)$.

3: Calcular:

$$D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x})) = \frac{1}{2} \sum_{k=1}^{n+m} w_k \sum_{j=0}^{2d} g(\mathcal{X}_{kj}) \mathcal{W}_{kj}$$

Salida: Distancia Ponderada de Hellinger:

$$D^2_{Hellinger}(f_1(\mathbf{x}), f_2(\mathbf{x}))$$

3.4. Agrupación Jerárquica de Componentes en Mezclas de Gaussianas.

Goldberger y Roweis [10] proponen un método para agrupar componentes usando la divergencia -KL entre dos Gaussianas d-dimensionales dado en la expresión (3.43). La idea consiste en crear una matriz de distancias con la medida simétrica

$$dist_{\mathrm{KL}} = \frac{1}{2}\left\{D_{\mathrm{KL}}\left[\phi(\mathbf{x};\mu_a,\mathbf{\Sigma}_a),\phi(\mathbf{x};\mu_b,\mathbf{\Sigma}_b)\right] + D_{\mathrm{KL}}\left[\phi(\mathbf{x};\mu_b,\mathbf{\Sigma}_b),\phi(\mathbf{x};\mu_a,\mathbf{\Sigma}_a)\right]\right\}, \quad (3.57)$$

donde $\phi(\mathbf{x};\mu_k,\mathbf{\Sigma}_k)$ es una densidad gaussiana d-dimensional con media μ_k y matriz de covarianzas $\mathbf{\Sigma}_k$, para $k = a, b$.
Luego las componentes gaussianas con menor distancia $dist_{\mathrm{KL}}$ colapsan en una única gaussiana, para la cual se definen los parámetros:

$$\begin{aligned} &\alpha_{ab} = \alpha_a + \alpha_b, \quad \mu_{ab} = \alpha_{ab}^{-1}\left(\alpha_a\mu_a + \alpha_b\mu_b\right), \\ &\mathbf{\Sigma}_{ab} = \alpha_{ab}^{-1}\sum_{k=a,b}\alpha_k\left[\mathbf{\Sigma}_k + \left(\mu_k - \mu_{ab}\right)\left(\mu_k - \mu_{ab}\right)^T\right] \end{aligned} \quad (3.58)$$

El proceso se repite de forma iterativa hasta llegar a tener k componentes, con $1 \leq k \leq M$ y M la cantidad total de componentes de la mezcla. En la Figura 3.6 se muestra el proceso de agrupación para $k = 2$ en una mezcla con $M = 6$ componentes.

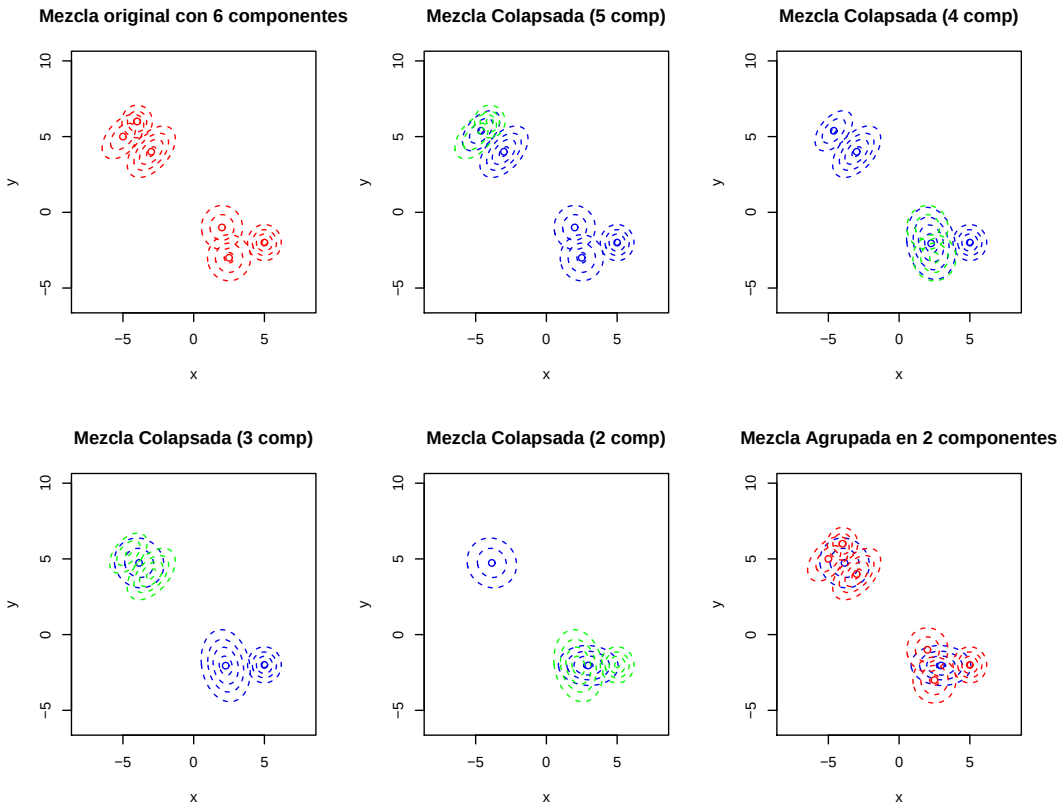

Figura 3.6: Proceso de agrupación jerárquico para $M = 6$ componentes en $k = 2$ componentes usando la distancia $dist_{\mathrm{KL}}$.

CAPÍTULO 4

MEZCLAS ADAPTATIVAS EN FLUJO DE DATOS IDÉNTICAMENTE DISTRIBUIDOS.

4.1. Mezclas Adaptativas (AM)

Otro camino para resolver los problemas del funcionamiento del algoritmo (EM) en mezclas gaussianas es el propuesto por Priebe [28] denominado **Mezclas Adaptativas (AM)**, el cual es un *método híbrido* de los estimadores núcleo y las mezclas gaussianas y en donde no se establece ninguna estructura para las matrices de covarianzas $\mathbf{\Sigma}$.

Las Mezclas Adaptativas (AM) están diseñadas para manejar flujos de datos idénticamente distribuidos [1], actualizando el modelo estadístico existente con la llegada secuencial de nuevos datos puntuales y la ejecución del algoritmo (EM). Si no se satisface un *umbral de creación*, el método adiciona estos nuevos datos como nuevos componentes. El método de Priebe [28] es considerado un método recursivo robusto [2] para la estimación de densidades de probabilidad, donde el estimador para las $[t+1]$ observaciones (es decir para los elementos del conjunto $\{x^{[1]}, x^{[2]}, \ldots, x^{[t]}, x^{[t+1]}\}$ [3], donde $x^{[t]} \in \mathbb{R}^d$), es precisamente una función de probabilidad de la $[t+1]$-ésima observación, denotada por $x^{[t+1]}$, y el estimador basado en las t-observaciones previas $\{x^{[1]}, \ldots, x^{[t]}\}$. Esto es,

$$\hat{f}^{[t+1]}(x^{[1]}, x^{[2]}, \ldots, x^{[t]}, x^{[t+1]}) = \mathcal{F}\left(\hat{f}^{[t]}, x^{[t+1]}\right).$$

Este procedimiento evita la necesidad de almacenar todas las observaciones entrantes, lo que permite la manipulación de grandes rangos de datos. Es así como el proceso de Mezclas Adaptativas (AM) produce una secuencia de estimadores $\{\hat{f}^{[t]}\}$, los cuales combinan la consistencia producida por los estimadores de densidad núcleo para dicha

[1] En esta tesis se consideran observaciones $x^{[t]} \in \mathbb{R}^d$ idénticamente distribuidas (id) para una variable aleatoria $\mathbf{x}$, donde la naturaleza de cualquier correlación entre observaciones es la misma a lo largo del tiempo.

[2] La estadística robusta es una aproximación alternativa a los métodos estadísticos clásicos. La estadística robusta intenta proporcionar métodos que emulan a los métodos clásicos, pero que no son afectados indebidamente por valores atípicos u otras pequeñas discrepancias respecto de las asunciones del modelo (ver [11]).

[3] Aquí consideramos que n-individuos llegan en forma secuencial a la muestra en un tiempo t determinado. Luego se hace la siguiente convención para la notación
$\{x_1^{[1]}, x_2^{[2]}, \ldots, x_n^{[t]}\} \equiv \{x^{[1]}, x^{[2]}, \ldots, x^{[t]}\}$

famila de funciones y la baja complejidad computacional asociada con los métodos de mezcla finita. Luego las Mezclas Adaptativas (AM) son definidas como un proceso de aproximación estocástica, donde los parámetros son actualizados recursivamente usando la siguiente ecuación

$$\hat{\theta}^{[t+1]} = \left[1 - P\left(x^{[t+1]}|\hat{\theta}^{[t]}\right)\right] * U\left(x^{[t+1]}|\hat{\theta}^{[t]}\right) + P\left(x^{[t+1]}|\hat{\theta}^{[t]}\right) * C\left(x^{[t+1]}|\hat{\theta}^{[t]}\right). \quad (4.1)$$

Aquí $P(\cdot)$ representa una **regla de decisión** que toma valores de 0 o 1, $U(\cdot)$ representa una **regla de actualización** de los parámetros como una estimación de máxima verosimilitud recursiva, y $C(\cdot)$ representa una **regla de creación** de un nuevo componente similar al modelo de aproximación de estimación núcleo. La regla de actualización parámetrica se basa en una versión recursiva del algoritmo (EM) desarrollado por Titterington [38]. [4]

4.1.1. Regla de Decisión.

La adición de una nueva componente a una mezcla gaussiana está sujeta a las distancias que existen entre el nuevo dato puntual $x^{[t+1]}$ y cada componente $\phi_(\mathbf{x} : \mu_i, \mathbf{\Sigma}_i)$ de la ecuación (3.15). Aquí un método sencillo que se usa para esta medición es *la distancia Mahalanobis*, ya definida en la ecuación (2.7), cuyo valor (al cuadrado) entre

[4] Titterington propuso usar una versión de la aproximación estocástica iterativa del gradiente ascendente a la superficie log-verosimilitud de los parámetros, dado por:

$$\hat{\theta}^{[t+1]} = \hat{\theta}^{[t]} + \beta^{[t]} \{\nabla_\theta \ln[l(\theta|\mathrm{X})]\}$$

donde $\beta^{[t]}$ es una sucesión que converge a cero.

una nueva observación $x^{[t+1]}$ y el centro de cada componente $\phi(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i)$ está dado por:

$$D_M^2\left(x^{[t+1]}, \hat{\mu}_i^{[t]}; \hat{\mathbf{\Sigma}}_i^{[t]}\right) = \left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right)^T \left(\hat{\mathbf{\Sigma}}_i^{[t]}\right)^{-1} \left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right), \tag{4.2}$$

Luego si la mínima de estas distancias excede un umbral (llamado *umbral de creación* T_c) entonces se crea una nueva componente. Por lo tanto

$$P\left(x^{[t+1]}|\hat{\theta}^{[t]}\right) = \begin{cases} 1, & \text{si mín}_i \left\{D_M^2\left(x^{[t+1]}, \hat{\mu}_i^{[t]}; \hat{\mathbf{\Sigma}}_i^{[t]}\right)\right\} > T_c \\ 0, & \text{en otro caso.} \end{cases} \tag{4.3}$$

4.1.2. Regla de Actualización

Las ecuaciones recursivas para la actualización de los parámetros de una mezcla gaussiana tienen la siguiente forma:

$$\hat{\tau}_i^{[t+1]} = \frac{\hat{\alpha}_i^{[t]}\phi(x^{[t+1]}; \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]})}{\sum_{i=1}^{M^{[t]}} \hat{\alpha}_i^{[t]}\phi(x^{[t+1]}; \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]})}, \qquad \text{donde} \quad i = 1, 2, \ldots, M^{[t]}. \tag{4.4}$$

Aquí $\hat{\tau}_i^{[t+1]}$ representa *el estimado de la probabilidad a posteriori* de que un nuevo dato $x^{[t+1]}$ pertenezca a la i-ésima componente y el superíndice $[t]$ se usa para indicar que el parámetro está basado en las t-observaciones anteriores. El denominador, $\sum_{i=1}^{M^{[t]}} \hat{\alpha}_i^{[t]}\phi(x^{[t+1]}; \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]})$, es la estimación de mezcla gaussiana d-dimensional con vector de parámetros $\hat{\theta}_i^{[t]} = \left(\hat{\alpha}_i^{[t]}, \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]}\right)$.

Llamamos

$$\beta_i^{[t]} = \frac{1}{t} \tag{4.5}$$

$$\hat{\alpha}_i^{[t+1]} = \hat{\alpha}_i^{[t]} + \beta_i^{[t]}\left(\hat{\tau}_i^{[t+1]} - \hat{\alpha}_i^{[t]}\right) \tag{4.6}$$

$$\hat{\mu}_i^{[t+1]} = \hat{\mu}_i^{[t]} + \beta_i^{[t]} \left\{ \frac{\hat{\tau}_i^{[t+1]}}{\hat{\alpha}_i^{[t]}} \left(x^{[t+1]} - \hat{\mu}_i^{[t]} \right) \right\} \tag{4.7}$$

$$\hat{\mathbf{\Sigma}}_i^{[t+1]} = \hat{\mathbf{\Sigma}}_i^{[t]} + \beta_i^{[t]} \left\{ \frac{\hat{\tau}_i^{[t+1]}}{\hat{\alpha}_i^{[t]}} \left[\left(x^{[t+1]} - \hat{\mu}_i^{[t]} \right) \left(x^{[t+1]} - \hat{\mu}_i^{[t]} \right)^T - \hat{\mathbf{\Sigma}}_i^{[t]} \right] \right\} \tag{4.8}$$

4.1.3. Regla de Creación

Otro aspecto a destacar es que $M^{[t]}$, el número de componentes de la mezcla en el tiempo t, varía durante la ejecución del algoritmo. Este inicia con una sola componente y se van adicionando otras, si se observa que los datos no están adecuadamente reflejados en el modelo existente. Entonces se agregará una nueva componente si en la ecuación (4.3) la distancia entre el nuevo dato puntual $x^{[t+1]}$ y el centro de cada componente $\phi(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i)$ es mayor que el valor umbral T_c. Luego la nueva componente es centrada en la nueva observación y los coeficientes de mezcla existentes son actualizados. Si llamamos $M^{[t+1]} \equiv M^{[t]} + 1 \equiv M^*$, las ecuaciones de creación son:

$$\hat{\alpha}_{M^*}^{[t+1]} = \frac{1}{t+1} \tag{4.9}$$

Para garantizar que la suma de los coeficientes de la mezcla sea uno, cuando la nueva componente se agrega, el valor de $\hat{\alpha}_i^{[t+1]}$ debe ser reescalado:

$$\hat{\alpha}^{[t+1]} = \left(\hat{\alpha}_i^{[t]} \left[1 - \hat{\alpha}_{M^*}^{[t+1]} \right], \hat{\alpha}_{M^*}^{[t+1]} \right) \quad donde \quad i = 1, \ldots, M^{[t]}. \tag{4.10}$$

$$\hat{\mu}_{M^*}^{[t+1]} = x^{[t+1]} \tag{4.11}$$

$$\hat{\mathbf{\Sigma}}_{M^*}^{[t+1]} = \sum_{i=1}^{t} \left(\hat{\tau}_i^{[t+1]} * \hat{\mathbf{\Sigma}}_i \right). \tag{4.12}$$

Aquí $\hat{\mathbf{\Sigma}}_{M^*}^{[t+1]}$ es el promedio ponderado de las matrices de covarianzas hasta el tiempo t. En la práctica cualquier otro estimador de la matriz de covarianza inicial puede ser utilizado para la nueva componente.

4.1.4. Algoritmo (AMDE)

Martínez y Martínez [40] muestran el algoritmo (AM) para la estimación de densidades con mezclas adaptativas (AMDE) tal y como se presenta en el algoritmo 4.1, que se muestra debajo.

En la práctica el método de Mezclas Adaptivas (AM) se utiliza para obtener valores iniciales de los parámetros así como un referente en el número de términos necesarios para modelar la densidad. Se podría utilizar este proceso como punto de partida en un proceso de clasificación de datos y luego aplicar el algoritmo (EM) para refinar las estimaciones.

Algoritmo 4.1 Algoritmo (AMDE)

1: Iniciar el proceso de Mezclas Adaptivas (AM) usando el primer dato puntual $x^{[1]}$:

$$\hat{\alpha}_1^{[1]} = 1, \quad \hat{\mu}_1^{[1]} = x^{[1]}, \quad \hat{\mathbf{\Sigma}}_1^{[1]} = \mathbf{I}_{d\times d}.$$

2: Para un nuevo dato puntual $x^{[t+1]}$, calcular la distancia Mahalanobis al cuadrado dada en la ecuación (4.2)

$$D_M^2\left(x^{[t+1]}, \hat{\mu}_i^{[t]}; \hat{\mathbf{\Sigma}}_i^{[t]}\right) = \left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right)^T \left(\hat{\mathbf{\Sigma}}_i^{[t]}\right)^{-1} \left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right).$$

3: **si**

$$\min_i \left\{ D_M^2\left(x^{[t+1]}, \hat{\mu}_i^{[t]}; \hat{\mathbf{\Sigma}}_i^{[t]}\right)\right\} > T_c$$

entonces

4: Hallar *el estimado de la probabilidad a posteriori* de que un nuevo dato $x^{[t+1]}$ pertenezca al i-ésima componente.

$$\hat{\tau}_i^{[t+1]} = \frac{\hat{\alpha}_i^{[t]}\phi(x^{[t+1]}; \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]})}{\sum_{i=1}^{M^{[t]}} \hat{\alpha}_i^{[t]}\phi(x^{[t+1]}; \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]})}, \qquad \textit{donde} \quad i = 1, 2, \ldots, M^{[t]}.$$

5: Crear una nueva componente usando las ecuaciones (4.9), (4.10), (4.11) y (4.12)

$$\hat{\alpha}_{M^*}^{[t+1]} = \frac{1}{t+1}, \quad \hat{\alpha}^{[t+1]} = \left(\hat{\alpha}_i^{[t]}\left[1 - \hat{\alpha}_{M^*}^{[t+1]}\right], \hat{\alpha}_{M^*}^{[t+1]}\right), \quad \hat{\mu}_{M^*}^{[t+1]} = x^{[t+1]}, \quad \hat{\mathbf{\Sigma}}_{M^*}^{[t+1]} = \sum_{i=1}^{t}\left(\hat{\tau}_i^{[t+1]} * \hat{\mathbf{\Sigma}}_i\right)$$

6: **si no**

7: Actualizar las componentes existentes usando $\beta_i^{[t]} = \frac{1}{t}$ y las ecuaciones (4.5), (4.6), (4.7) y (4.8)

$$\hat{\alpha}_i^{[t+1]} = \hat{\alpha}_i^{[t]} + \beta_i^{[t]}\left(\hat{\tau}_i^{[t+1]} - \hat{\alpha}_i^{[t]}\right), \quad \hat{\mu}_i^{[t+1]} = \hat{\mu}_i^{[t]} + \beta_i^{[t]}\left\{\frac{\hat{\tau}_i^{[t+1]}}{\hat{\alpha}_i^{[t]}}\left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right)\right\}.$$

$$\hat{\mathbf{\Sigma}}_i^{[t+1]} = \hat{\mathbf{\Sigma}}_i^{[t]} + \beta_i^{[t]}\left\{\frac{\hat{\tau}_i^{[t+1]}}{\hat{\alpha}_i^{[t]}}\left[\left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right)\left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right)^T - \hat{\mathbf{\Sigma}}_i^{[t]}\right]\right\}$$

8: **fin si**

9: Repita los pasos 2 al 7 para cada uno de los datos muestrales

4.1.5. Problemas en el proceso de Mezclas Adaptativas (AM)

Martínez y Martínez [40] muestran los inconvenientes que tiene usar el algoritmo (AM) para la estimación de densidades. Estos se listan a continuación

1. La complejidad del modelo o el número de componentes a veces es mayor de la necesaria. Tiende a producir modelos demasiado complejos (demasiadas componentes).

2. Diferentes modelos (número de componentes y parámetros estimados en las componentes) pueden en esencia generar la misma estimación de la función o curva para $\hat{f}(\mathbf{x})$.

3. El modelo resultante o la densidad de probabilidad estimada dependen del orden en que los datos se presentan en el algoritmo.

4. Las matrices de covarianzas no tienen restricciones (problema de cálculo computacional).

5. Aplicabilidad limitada a espacios de alta dimensión.

4.2. Modelado y Requerimientos en Flujos de Datos

Heinz y Seeger [5] definen un flujo de datos como una sucesión $x^{[1]}, x^{[2]}, x^{[3]}, \ldots$, con $x^{[t]} \in \mathbb{R}^d$ para $t \in \mathbb{N}$. En este trabajo, excepto que se indique lo contrario, se asumirá que la observación $x^{[t]}$ en el instante $[t]$ es una variable aleatoria continua, idénticamente distribuida (id). Aquí se estudiarán flujos de datos con llegada continua

y potencialmente ilimitados (cuasi–infinitos), así como grandes conjuntos de datos fijos, que por volumen son considerados cuasi–infinitos y los cuales se analizarán en forma de cuasi-flujo con su arribo secuencial. El supuesto de distribución idéntica caracteriza a estos datos como el resultado de un proceso estocástico estacionario.
Con el fin de mantener un ritmo continuo adecuado en los flujos de datos, cualquier técnica de análisis y modelado debe cumplir con los siguientes requisitos de procesamiento [25]:

1. Cada elemento se procesa una sola vez.
2. El tiempo de procesamiento por elemento es constante.
3. La cantidad de memoria es constante.
4. Un modelo válido está disponible en cualquier momento durante el proceso, es decir no requiere que el flujo completo sea construido.
5. Los modelos deben adaptarse a los cambios en flujos de datos, así como a los cambios en las propiedades estadísticas de las variables que el modelo está tratando de estimar.
6. Los modelos siempre deben ser comparables a los mejores modelos homólogos *off-line*, los cuales siempre tienen recursos ilimitados a la mano y acceso arbitrario a todos los elementos.

Ahora, definida la clase de datos a trabajar en esta tesis y los requerimientos necesarios para obtener aproximaciones a modelos adecuados de su distribución de probabilidad, se presentan un método que permiten estimar la función de densidad en un flujo de datos idénticamente distribuidos.

4.3. Estimación de Densidades con Mezclas Adaptativas en línea (oAMDE)

Szewczyk [35] presenta la adaptación de las Mezclas Adaptativas (AM) a flujos de datos; modifica la regla de actualización usando la primera versión del *algoritmo Esperanza-Maximización en línea (oEM)* propuesto por Sato e Ishii [18], en lugar de la versión recursiva que propuso Titterington [38]. Al algoritmo de Szewczyk [35] es lo que denominamos *Primera Versión* de las *Estimación de Densidades con Mezclas Adaptativas en línea* ($oAMDE[V1]$). A continuación se describe primero el algoritmo (oEM).

4.3.1. Esperanza-Maximización en línea (oEM)

Además de los inconvenientes mencionados por Martinez y Martinez [40], el algoritmo EM clásico presenta serios limitantes de recursividad, almacenamiento y convergencia para flujos de datos.

Cappé, en la sección 2.3.3 de [13], muestra las limitaciones del algoritmo EM clásico para flujo de datos. Allí, se dá un ejemplo donde se puede apreciar cómo el tamaño en el conjunto de datos influye tanto en el número de iteraciones como en la trayectoria de convergencia en la estimación de los parámetros, lo que Cappé [13] denomina *un problema de recursividad*. Dicha trayectoria sólo depende de los parámetros de inicialización del algoritmo. En este ejemplo también se fija el número de iteraciones y se observa cómo las estimaciones generadas por el algoritmo EM clásico para un conjunto muy grande de observaciones (del orden de $20,000$), no mejora de forma significativa

respecto a un conjunto de observaciones no tan grande (del orden de $2,000$). Por último, se muestra cómo para el conjunto con menor número de datos la convergencia de las estimaciones se estabiliza más rápidamente (3 a 4 iteraciones), que para el conjunto que posee una cantidad más elevada de observaciones (20 iteraciones). Cabe recordar que los cálculos realizados en el **paso E** deben realizarse para todas las observaciones, por lo tanto en este punto la complejidad es directamente proporcional al número de observaciones.

La solución a estos problemas ha sido tratada en los ultimos años con la modificación del algoritmo EM, almacenando un único conjunto de estimadores suficientes (Paso E) en cada instante $[t]$ y actualizándolos después del procesamiento de cada observación. A esto se le conoce recientemente con el nombre de *EM en línea (oEM)*.

Neal y Hinton [29] propusieron la primera aproximacion a un algoritmo *EM en línea*, la cual denominaron *incremental EM (iEM)*. Aquí el algoritmo empieza con el cálculo o la actualización de los estimadores muestrales suficientes en el tiempo $[t]$ (medias y varianzas para nuestro caso) para el conjunto de datos disponibles. Luego en el **paso E** se halla la función de verosimilitud para el conjunto de datos actualizado y se calculan los valores esperados poblacionales. Finalmente, en el **paso M** se actualizan los antiguos estimados poblacionales restándole a éstos los estimadores suficientes calculados en el tiempo $[t]$ y a la vez sumándole los valores esperados de los parámetros poblacionales calculados en el **paso E**.

Sato e Ishii [18] desarrollaron una variante del *(iEM)* que llamaron *stepwise EM (sEM)*. Este algoritmo es reconocido por muchos como el primer EM en línea (oEM). Aquí se interpola entre valores esperados poblacionales y los estimadores muestrales suficientes, basados en un tamaño de paso $\beta^{[t]}$ ($[t]$ es considerado como el número de actualizacio-

nes hechas en el **paso E** hasta el momento)[5]. Además, Sato e Ishii [18] proporcionan un análisis detallado de la convergencia del algoritmo para el caso de familias exponenciales de parametrización natural y para el caso de mezclas gaussianas. Cappé y Moulines [24] generalizan la propuesta de Sato e Ishii y la denominan *EM en línea (oEM)*. Este método de aproximación a los estimadores de los parámetros en flujos de datos es de naturaleza estocástica y se presenta en el algoritmo 4.2.

El algoritmo *incremental EM (iEM)* de Neal y Hinton [29] es equivalente al *EM en línea (oEM)* de Cappé y Moulines [24] , si se utiliza $\beta^{[t]} = \frac{1}{t}$. Sato e Ishii [18] utilizan un tamaño de paso $\beta^{[t]} = \frac{1}{t+1}$. Para grandes conjuntos de datos fijos (cuasi-flujos), el algoritmo EM clásico es un procedimiento impráctico y menos recomendable que el *EM en línea (oEM)* como se muestran en los experimentos de Liang y Klein [26].

Algoritmo 4.2 EM en línea

1: Dados $l^*(\theta|\mathrm{X}^{[0]})$, $\widehat{\theta}^{[0]}$ y una sucesión de pasos $\left\{\beta^{[t]}\right\}_{t\geq 1}$

2: **Para** $t \geq 1$ **hacer**

3: **Paso E:** (Estocástico)

$$l^*(\theta|\mathrm{X}^{[t]}) = \left[1-\beta^{[t]}\right] l^*(\hat{\theta}^{[t-1]}|\mathrm{X}^{[t-1]}) + \beta^{[t]} E_{Z|\widehat{\theta}^{[t-1]}}\Big\{\ln[f(X^{[t]};\hat{\theta}^{[t-1]})]\Big\}$$

4: **Paso M:**

$$\widehat{\theta}^{[t]} = \max_{\theta}\Big[l^*(\theta|\mathrm{X}^{[t]})\Big].$$

5: **fin Para**

En [13], Cappé aclara que algoritmo EM en línea no debe ser interpretado como

[5]Se asume que la sucesión de tamaños de paso satisface que: $\sum_t \beta^{[t]} = \infty, \sum_t (\beta^{[t]})^2 < \infty$.

una aproximación estocástica de las estadísticas suficientes sino de los parámetros y además demuestra que este algoritmo es asintóticamente equivalente a la aproximación estocástica iterativa del gradiente ascendente propuesta por Titterington [38]. Recomienda considerar sucesiones con tamaño de pasos $\beta^{[t]} = \frac{1}{t^{\alpha}}$ con $0.6 \leq \alpha \leq 0.9$. El ajuste más fuerte se obtiene al tomar α cerca de 0.6, junto con **el promedio de Polyak-Ruppert**, el cual consiste en sustituir los valores de los parámetros estimados en el algoritmo en línea, $\widehat{\theta}^{[t]}$, por el promedio dado por:

$$\tilde{\theta}^{[t]} = \frac{1}{t - t_0} \sum_{t=t_0+1}^{t} \widehat{\theta}^{[t]}, \tag{4.13}$$

donde t_0 es un índice positivo a partir del cual se inicia el algoritmo. La aplicación de este promedio es eficiente, pero requiere elegir un valor de t_0 suficientemente grande para evitar introducir sesgo debido a la falta de convergencia. Cappé [13] sugiere tomar para el caso de grandes conjuntos de datos fijos, un valor de t_0 igual a la mitad de la longitud de estos registros. Demuestra, además que valores de α del orden de 0.6 tiene un buen rendiminiento, un poco más robusto, en su convergencia.

Desde el punto de vista computacional, la principal diferencia entre el algoritmo *EM en línea (oEM)* y el *algoritmo clásico EM*, es que el *en línea* realiza la actualización después de la llegada de cada observación, mientras que el *algoritmo clásico EM* sólo aplica la actualización en el **paso M**, después de haber procesado los datos completos. Si se trabajan con grandes conjuntos de datos fijos (cuasi-flujos), Cappé [13] recomienda lo siguiente:

- Si el número de observaciones disponibles t es pequeño (menos de 1000 observaciones), el *algoritmo clásico EM* puede ser mucho más rápido que el algoritmo *EM en línea*, sobre todo si se quiere obtener una aproximación numérica con

menor error que la suministrada con el estimador de máxima verosimilitud. Esto puede ser innecesario ya que el propio estimador de máxima verosimilitud es solo una aproximación de valor real del parámetro con un error de orden de $\frac{1}{\sqrt{t}}$ en modelos estadísticos regulares.

- Cuando t está en constante aumento, el algoritmo EM en línea es preferible para un valor de t suficientemente grande. En este caso, el estimado del algoritmo *EM en línea* es asintóticamente equivalente a los estimadores de máxima verosimilitud.

En el caso de grandes conjuntos de datos fijos (cuasi-flujos), el algoritmo *EM en línea* debe ser utilizado en repetidas ocasiones mediante el reordenamiento de los datos con el fin de converger al estimador de máxima verosimilitud.

4.3.2. Versiones oAMDE de Cappé.

En este trabajo de tesis tomamos el algoritmo propuesto por Szewczyk [35] y cambiamos el tamaño de paso original $\beta^{[t]} = \frac{1}{t+1}$, basado en el algoritmo de Sato e Ishii [18] y el cual denominamos anteriormente ($oAMDE[V1]$), por el propuesto por Cappé [13], $\beta^{[t]} = \frac{1}{t^{\alpha}}$ con $0.6 \leq \alpha \leq 0.9$, el cual denominaremos ($oAMDE[v2]$). Si tomamos un valor de t_0 igual un porcentaje menor o igual a la mitad de la longitud del total de datos y usamos el promedio de Polyak-Ruppert para calcular la estimación de cada parámetro, obtenemos otra versión para la *Estimación de Densidades con Mezclas Adaptativas en línea* ($oAMDE[v3]$). A continuación describimos el algoritmo de Cappé.

Consideremos el estimador de densidad dado por un modelo de mezclas gaussianas

d-dimensionales, después de llegar t-observaciones, como:

$$\hat{f}^{[t]}(\mathbf{x}, \theta^{[t]}) = \sum_{i=1}^{M^{[t]}} \alpha_i^{[t]} \phi(\mathbf{x}; \mu_i^{[t]}, \mathbf{\Sigma}_i^{[t]}) \tag{4.14}$$

Cuando una nueva observación, $x^{[t+1]}$, arriba (y por el momento suponemos que una nuevo componente no se agregará), entonces los parámetros de la mezcla se actualizan mediante actualizaciones dadas por el algoritmo *EM en línea* de Cappé [13].

Algoritmo 4.3 Algoritmo (oAMDE) de Cappé

1: Inicie con proceso de Mezclas Adaptivas (AM) tradicional para las primeras t_0 observaciones:

2: Para un nuevo dato puntual $x^{[t+1]}$, con $t+1 = t_0+1$, calcule la distancia Mahalanobis al cuadrado dada por la ecuación:

$$D_M^2\left(x^{[t+1]}, \hat{\mu}_i^{[t]}; \hat{\mathbf{\Sigma}}_i^{[t]}\right) = \left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right)^T \left(\hat{\mathbf{\Sigma}}_i^{[t]}\right)^{-1} \left(x^{[t+1]} - \hat{\mu}_i^{[t]}\right).$$

3: **si**

$$\min_i \left\{ D_M^2\left(x^{[t+1]}, \hat{\mu}_i^{[t]}; \hat{\mathbf{\Sigma}}_i^{[t]}\right) \right\} > T_c$$

entonces

4: Cree una nueva componente al hacer: $M^{[t+1]} = M^{[t]} + 1$

5: Use la **Regla de Actualización en Mezclas Adaptativas en línea**

6: **si no**

7: Hacer $M^{[t+1]} = M^{[t]}$

8: Use la **Regla de Actualización en Mezclas Adaptativas en línea**

9: **fin si**

Algoritmo 4.4 Regla de Actualizacion en Mezclas Adaptativas en línea

1: Calcule para cada $t > t_0$:

2:
$$\beta_i^{[t]} = \frac{1}{t^\alpha}$$

3:
$$\hat{\tau}_i^{[t+1]} = \frac{\hat{\alpha}_i^{[t]} \phi(x^{[t+1]}; \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]})}{\sum_{i=1}^{M^{[t]}} \hat{\alpha}_i^{[t]} \phi(x^{[t+1]}; \hat{\mu}_i^{[t]}, \hat{\mathbf{\Sigma}}_i^{[t]})}.$$

4:
$$\hat{\alpha}_i^{[t+1]} = \hat{\alpha}_i^{[t]} + \beta_i^{[t]} \left(\hat{\tau}_i^{[t+1]} - \hat{\alpha}_i^{[t]} \right), \qquad \tilde{\alpha}_i^{[t+1]} = \frac{1}{(t+1) - t_0} \sum_{k=t_0+1}^{t+1} \widehat{\alpha}^{[k]}$$

5:
$$\hat{\mu}_i^{[t+1]} = \hat{\mu}_i^{[t]} + \beta_i^{[t]} \left\{ \frac{\hat{\tau}_i^{[t+1]}}{\hat{\alpha}_i^{[t]}} \left(x^{[t+1]} - \hat{\mu}_i^{[t]} \right) \right\}, \qquad \tilde{\mu}_i^{[t+1]} = \frac{1}{(t+1) - t_0} \sum_{k=t_0+1}^{t+1} \widehat{\mu}^{[k]}.$$

6:
$$\hat{\mathbf{\Sigma}}_i^{[t+1]} = \hat{\mathbf{\Sigma}}_i^{[t]} + \beta_i^{[t]} \left\{ \frac{\hat{\tau}_i^{[t+1]}}{\hat{\alpha}_i^{[t]}} \left[\left(x^{[t+1]} - \hat{\mu}_i^{[t]} \right) \left(x^{[t+1]} - \hat{\mu}_i^{[t]} \right)^T - \hat{\mathbf{\Sigma}}_i^{[t]} \right] \right\},$$

7:
$$\tilde{\mathbf{\Sigma}}_i^{[t+1]} = \frac{1}{(t+1) - t_0} \sum_{k=t_0+1}^{t+1} \widehat{\mathbf{\Sigma}}^{[k]}.$$

4.3.3. Ajuste al oAMDE usando Agrupación de Componentes con Grafos.

Aunque el algoritmo oEM propuesto por Cappé en [13] mejora de manera sustancial el oAMDE propuesto por Szewczyk [35], Cappé advierte de la importancia del ordenamiento en los grandes conjuntos de datos fijos, influyendo esto en la convergencia de

los estimadores y en la complejidad del modelo. Cappé suguiere usar el algoritmo en *repetidas* ocasiones mediante reordenamiento de los datos para aliviar este problema. En esta tesis se propone un algoritmo basado en agrupamientos de componentes en el tiempo t. Estos agrupamientos generan *submezclas* [6] que deben ser colapsadas en componentes únicas, las cuales representarán *de la mejor manera* los subconjuntos de componentes. A continuación se describe el proceso de compresión.

Compresión de Mezclas Gaussianas Multivariadas.

Kristan et al. [16] proponen un algoritmo de compresión de mezclas gaussianas para controlar el exceso de componentes en flujo de datos en línea. El objetivo del algoritmo de compresión es aproximar una mezcla $f^{[t]}(\mathbf{x})$ de $M^{[t]}$-componentes,

$$f^{[t]}(\mathbf{x}) = \sum_{i=1}^{M^{[t]}} \alpha_i^{[t]} \phi(x; \mu_i^{[t]}, \mathbf{\Sigma}_i^{[t]}), \tag{4.15}$$

a una distribución equivalente $\tilde{f}^{[t]}(\mathbf{x})$ con $m^{[t]}$-componentes,

$$\tilde{f}^{[t]}(\mathbf{x}) = \sum_{j=1}^{m^{[t]}} \tilde{\alpha}_j^{[t]} \phi(x; \mu_j^{[t]}, \mathbf{\Sigma}_j^{[t]}), \tag{4.16}$$

donde $m^{[t]} < M^{[t]}$.

Kristan et al. [16] recurren a un enfoque basado en agrupaciones jerárquicas, el cual se describe en esta tesis en la sección 3.4. La idea principal es identificar *grupos de componentes* en la mezcla gaussiana $f^{[t]}(\mathbf{x})$, de tal manera que cada grupo puede ser lo suficientemente bien aproximado por una única componente en la estimación $\tilde{f}^{[t]}(\mathbf{x})$. Sea $\Xi(m^{[t]}) = \{\pi_j^{[t]}\}_{j=1:m^{[t]}}$ una colección de conjuntos con índices disjuntos, la cual

[6] Llamaremos submezclas a un subconjunto de componentes de la mezcla orignal junto con sus pesos. Es claro que en una submezcla la suma de los pesos es menor o igual que 1.

agrupa a $\tilde{f}^{[t]}(\mathbf{x})$ en $m^{[t]}$-submezclas. Las submezclas correspondientes a los índices $i \in \pi_j^{[t]}$ se definen como:

$$f_j^{[t]}(\mathbf{x}; \pi_j^{[t]}) = \sum_{i \in \pi_j} \alpha_i^{[t]} \phi(x; \mu_i^{[t]}, \mathbf{\Sigma}_i^{[t]}), \tag{4.17}$$

Cada submezcla es comprimida a la j-ésima componente $\tilde{\alpha}_j^{[t]} \phi(\mathbf{x}; \tilde{\mu}_j^{[t]}, \tilde{\Sigma}_j^{[t]})$ de $\tilde{f}^{[t]}(\mathbf{x})$. Los parámetros de la componente j-ésima se definen haciendo coincidir los dos primeros momentos (vector de medias y matriz de covarianzas) de la submezcla:

$$\begin{aligned} \tilde{\alpha}_j^{[t]} &= \sum_{i \in \pi_j} \alpha_i^{[t]}, \quad \tilde{\mu}_j^{[t]} = \frac{1}{\tilde{\alpha}_j^{[t]}} \sum_{i \in \pi_j} \alpha_i^{[t]} \mu_i^{[t]}, \\ \tilde{\mathbf{\Sigma}}_j^{[t]} &= \frac{1}{\tilde{\alpha}_j^{[t]}} \left\{ \sum_{i \in \pi_j} \alpha_i^{[t]} \left(\mathbf{\Sigma}_i^{[t]} + \left(\mu_i^{[t]} - \tilde{\mu}_j^{[t]} \right) \left(\mu_i^{[t]} - \tilde{\mu}_j^{[t]} \right)^T \right) \right\} \end{aligned} \tag{4.18}$$

Puesto que existen $\binom{M}{m}$ formas de agrupar componentes en el tiempo $[t]$, se debe escoger solo aquella opción que genere submezclas con menor *error de agrupación*. En esta tesis proponemos formar dichos grupos de componentes (con sus respectivos pesos), tomando *la distancia de Mahalanobis en la orientación del elipsoide de la componente con mayor peso.*

Definición 4.1. *Sean α_a, $\phi(\mathbf{x}; \mu_a, \mathbf{\Sigma}_a)$, α_b y $\phi(\boldsymbol{x}; \mu_b, \mathbf{\Sigma}_b)$ pesos y componentes de una mezcla gaussiana $f(\boldsymbol{x})$. Definimos* ***la distancia de Mahalanobis en la orientación del elipsoide de la componente con mayor peso****, asi:*

$$Si \quad \alpha_a \leq \alpha_b \quad entonces \quad D_{M,b}^2 = (\mu_a - \mu_b)^T \mathbf{\Sigma}_b^{-1} (\mu_a - \mu_b)$$

$$Si \quad \alpha_a > \alpha_b \quad entonces \quad D_{M,a}^2 = (\mu_b - \mu_a)^T \mathbf{\Sigma}_a^{-1} (\mu_b - \mu_a)$$

La *orientación* en la cual se mida la distancia entre las medias μ_a y μ_b, determinará si alguna de ellas pertenece o no a los espacios limitados por las elipsoides de

confianza respectivos, los cuales se definieron en la ecuación (2.9) . En la Figura 4.1 se muestra, para el caso $d = 2$, la importancia de tomar la orientación adecuada en la medición.

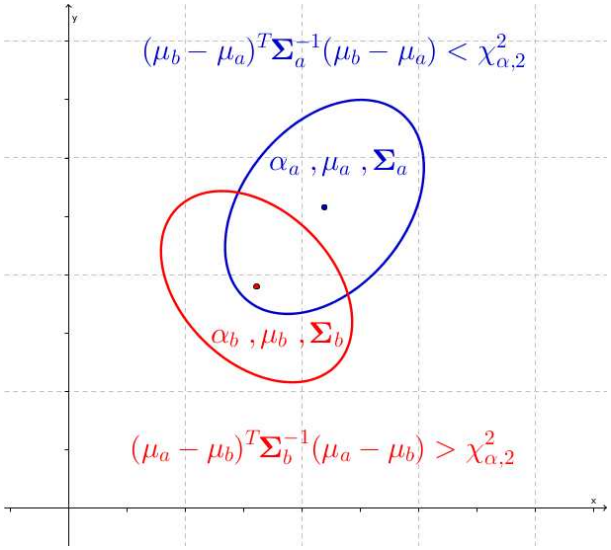

Figura 4.1: Distancia de Mahalanobis orientada para dos componentes de mezclas gaussianas bivariadas.

Ahora podemos construir la matriz de adyacencia $M_{Ady} = (m_{ij})_{M^{[t]} \times M^{[t]}}$, según la definición 2.4, de la siguiente manera:

$$m_{ij} = \begin{cases} 1, & \text{si } 0 < D_M^2 \left(\hat{\mu}_i^{[t]}, \hat{\mu}_j^{[t]}; \hat{\mathbf{\Sigma}}^{[t]} \right) < \chi^2_{\alpha,d} \\ 0, & \text{en otro caso,} \end{cases} \tag{4.19}$$

para $i, j = 1, 2, \ldots, M^{[t]}$ y $\alpha = 1 - tol$.

La matriz M_{Ady} determina un grafo que permite agrupar las componentes en las submezclas buscadas anteriormente.

CAPÍTULO 5

EJEMPLOS SIMULADOS

En este capítulo se realizaron algunos ejemplos simulados para mostrar el funcionamiento de los métodos propuestos. Los programas fueron diseñados y ejecutados en lenguaje de programación R, versión 2.15.0 y se utilizaron las librerías *ks*, *mvtnorm*, *fBasics*, *monomvn*, *accuracy*, *matrixcalc*, *mclust*, *car* y *igraph*.

Los datos aquí trabajados se generan de forma artificial con la instrucción *rmvnorm.mixt* de la libreria *ks*. Para cada uno de los experimentos se eligió un **umbral de creación** $\hat{T}_c$ de la siguiente manera:

1. Se tomaron 6 muestras aleatorias de la totalidad de los datos, con la instrucción *sample* de R, de tamaño $n = 0.1 * N$, donde N es la cantidad total de datos generados.

2. Para cada muestra, se pre-clasificaron los datos muestreados usando la instrucción *Mclust* de la libreria *mclust*, obteniendo así una *mezcla gaussiana multivariada piloto* o *mezcla piloto* para el estudio.

3. En cada mezcla piloto, dada por cada muestra, se determinaron los elipsoides de confianza [1] del $(1 - tol) * 100\,\%$, para cada componente de la mezcla piloto. Se tomó una tolerancia $tol = 10^{-4}$.

4. Para cada elipsoide de confianza, que representa la forma y orientación de cada componente, se determina las longitudes de los semiejes.

5. Las longitudes de los semiejes de cada componente son promediadas y luego se promedian estos valores hallados en todas las componentes de la mezcla piloto para la muestra realizada. El valor aquí encontrado determina *el radio* de la esfera que representará las componentes de manera única. Este radio al cuadrado será el estimador del umbral de creación Tc.

6. Finalmente tomamos TODOS los umbrales estimados $\hat{Tc}$, de todas las muestras, y los promediamos para encontrar el estimador final, que será un valor fijo durante todo el proceso de estimación de la densidad multivariada.

[1] Ver sección 2.3.1.

5.1. Ejemplos para $d = 2$.

5.1.1. Primer Ejemplo.

Se generan $50,000$ datos para la mezcla gaussiana bivariada con parámetros:

$$\alpha_1 = 0.42 \qquad \alpha_2 = 0.28 \qquad \alpha_3 = 0.30$$

$$\mu_1 = (-3, 0) \qquad \mu_2 = (3, 3) \qquad \mu_3 = (0, -3)$$

$$\mathbf{\Sigma}_1 = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix} \quad \mathbf{\Sigma}_2 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix} \quad \mathbf{\Sigma}_3 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

En la Figura 5.1 se muestran los datos y la elipse de confinaza del $99.99\,\%$ para los parámetros dados.

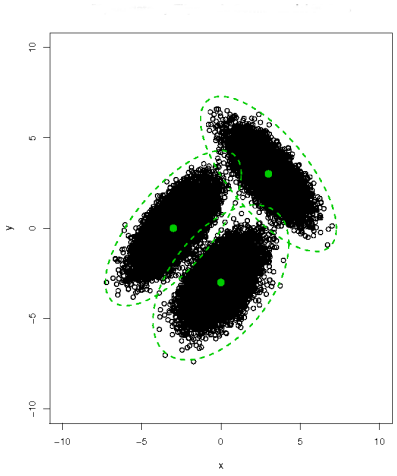

Figura 5.1: Datos y elipses de confinza del $99.99\,\%$ para el primer ejemplo con $d = 2$

Estimación del Umbral de Creación.

El paso a seguir es estimar el umbral de creación $\hat{Tc}$. Para ello se tomaron 6 muestras alatorias de tamaño $n = 5000$ y se obtuvo el valor $\hat{Tc} = 18.47151$. En la Figura 5.2 se observan las 6 muestras tomadas, las elipses de confiazas y las esferas para cada muestra del primer ejemplo. Los radios al cuadrado de cada esfera son los estimadores del umbral de creación $\hat{Tc}$.

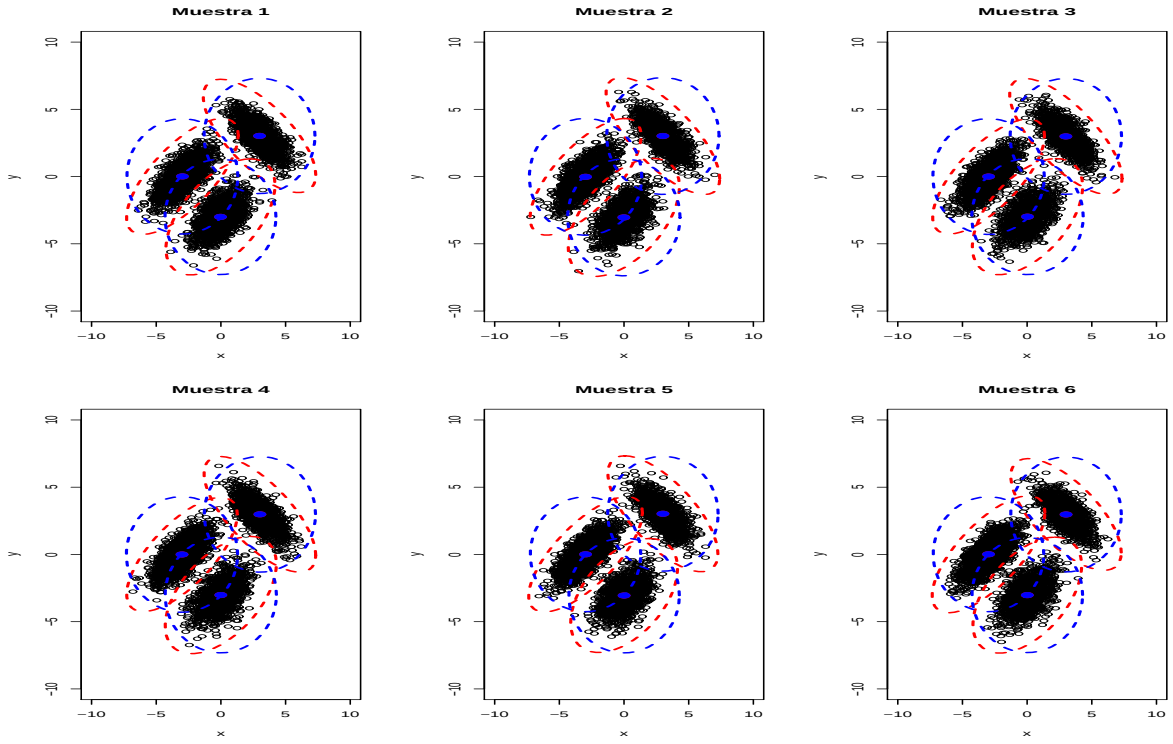

Figura 5.2: Datos, elipses de confianzas y esferas para cada muestra del primer ejemplo

En la Figura 5.3 se observan los valores de los diferentes estimadores de Tc para cada muestra. Aunque son muy similares estas cantidades, se toma el promedio de este

valor como el estimador final $\hat{Tc}$.

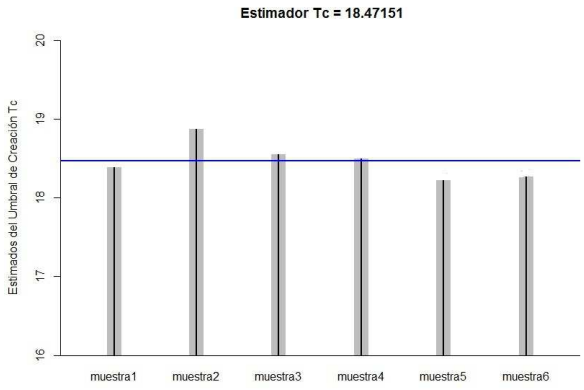

Figura 5.3: Estimadores del umbral de creación $\hat{Tc}$ para cada muestra en el primer ejemplo.

Proceso AMDE.

El proceso de estimación de la densidad bivariada empieza con el algoritmo 4.1 (AMDE), para las primeras $12,500$ observaciones. El arribo de cada observación $x^{[t]}$ se hace de forma secuencial y de uno en uno. Se realizan varias corridas al programa y se encuentran situaciones como la presentada en la Figura 5.4. En la parte (a) de la figura se observa una estimación para los primeros $12,500$ datos con una mezcla gaussiana bivariada de 7 componentes. Si eliminamos de la mezcla aquellas componentes cuyo peso sea menor que la tolerancia (tol) establecida anteriormente, se obtiene el modelo de la parte (b) de la figura, el cual tiene 6 componentes.

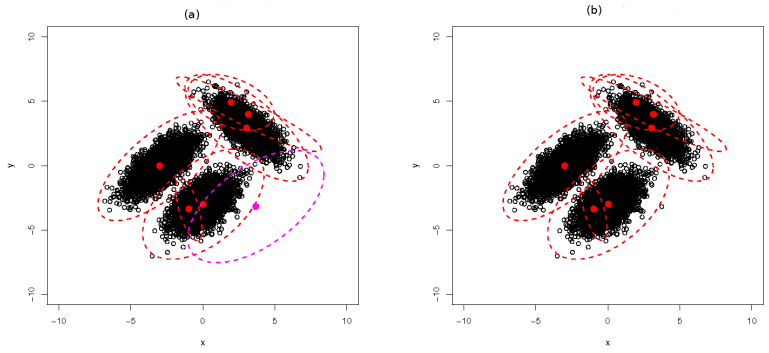

Figura 5.4: Estimación AMDE para $t = 12,500$ en el primer ejemplo. En (a) todas las componentes estimadas en el proceso. En (b) se eliminó la componente con peso poco significativo.

Las mezclas adaptativas tienden a producir modelos *más complejos* de lo necesario. El nivel de complejidad está marcado por el estimador del umbral Tc escogido. En nuestro caso las componentes en *exceso* tienden a estar cerca de las *componentes objetivo*, tal y como se muestran en la Figura 5.4. Luego se hace natural pensar en *agrupar componentes*. Acá se ordenan las componentes halladas en el proceso AMDE por su peso, en forma decreciente, tal como se muestra en la Figura 5.5.

Para cada par de componentes encontradas en el proceso AMDE, se calcula la *distancia de Mahalanobis orientada* obteniendo así la matriz

	1	2	3	4	5	6
1	0.00000	61.355227	39.847396	38.385741	49.763140	28.879525
2	61.35523	0.000000	35.133312	49.284309	1.088063	69.440158
3	39.84740	35.133312	0.000000	3.105704	205.572201	4.623284
4	38.38574	49.284309	3.105704	0.000000	1510.604587	2.297595
5	49.76314	1.088063	205.572201	1510.604587	0.000000	1067.289535
6	28.87952	69.440158	4.623284	2.297595	1067.289535	0.000000

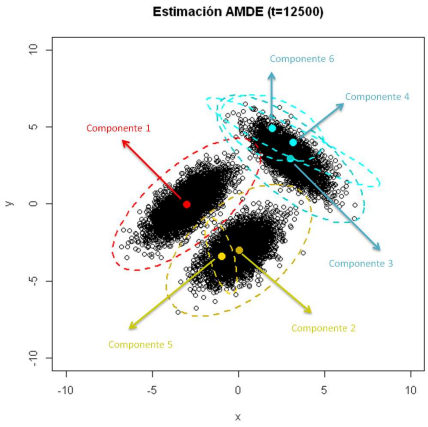

Figura 5.5: Etiquetado de las componentes de la mezcla en el proceso AMDE en el primer ejemplo.

Ahora se construye la matriz de adyacencia definida en la sección 4.3.3, donde $\chi^2_{0.9999,2} = 18.42068$, y el respectivo grafo que nos indicará los grupos a formar. Estos se pueden ver en la Figura 5.6

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	1	0
3	0	0	0	1	0	1
4	0	0	1	0	0	1
5	0	1	0	0	0	0
6	0	0	1	1	0	0

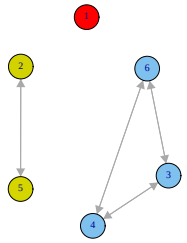

Figura 5.6: Matriz de Adyacencia y grafo para la estimación AMDE en el primer ejemplo.

Luego, las componentes agrupadas en submezclas se comprimen según lo visto en

la sección 4.3.3, obteniendo una estimación de 3 componentes tal y como se muestra en la Figura 5.7.

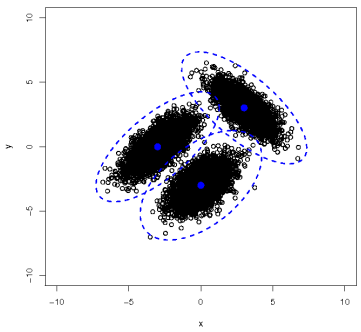

Figura 5.7: Ajuste de Componentes en AMDE para el primer ejemplo.

Proceso oAMDE

A continuación se presenta el proceso de estimación en línea oAMDE , donde luego de haber ajustado la densidad estimada en el proceso AMDE, se toma el modelo ajustado y se continúa la actulización con la llegada de cada nuevo dato. Aquí se presenta el proceso oAMDE para el 50 %, 75 % y 100 % de los datos. Al finalizar cada una de estas etapas, de ser necesario, se ajustan las mezclas estimadas con el proceso visto en la sección 4.3.3.

oAMDE para $t = 25,000$.

En esta parte de la tesis se implementaron los algoritmos $oAMDE_{V2}$ y $oAMDE_{V3}$. Pasados los primeros 25,000 datos, es decir el 50 % del total generado, se eliminaron las componentes que no tenían un peso significativo en el mezcla estimada. No se hizo

necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla estimada de componentes pesos son significativos. Los resultados se aprecian en la Figura 5.8.

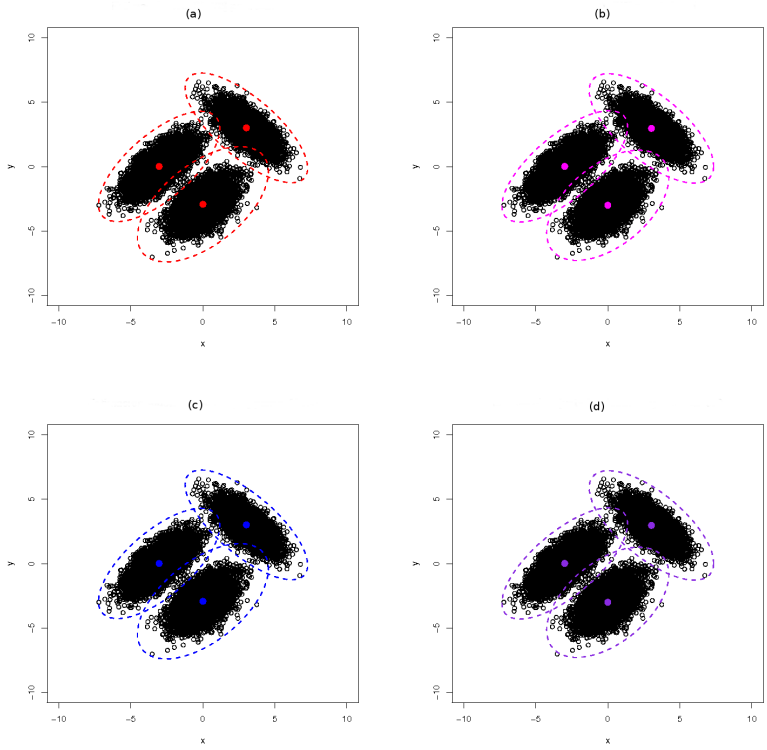

Figura 5.8: Estimación oAMDE para $t = 25,000$ en el primer ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura(b) muestra el proceso con el algoritmo $oAMDE[V3]$.Las Figuras (c) y (d) muestran los procesos ajustados para las Figuras (a) y (b).

oAMDE para $t = 37,500$.

Ahora se muestra el proceso transcurrido con el 75 % de los datos. Se eliminan las componentes que no tienen un peso significativo en el mezcla estimada y obtiene el oAMDE en las versiones de Cappé. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma mezcla estimada de componentes cuyos pesos son significativos. Los resultados se aprecian en la Figura 5.9.

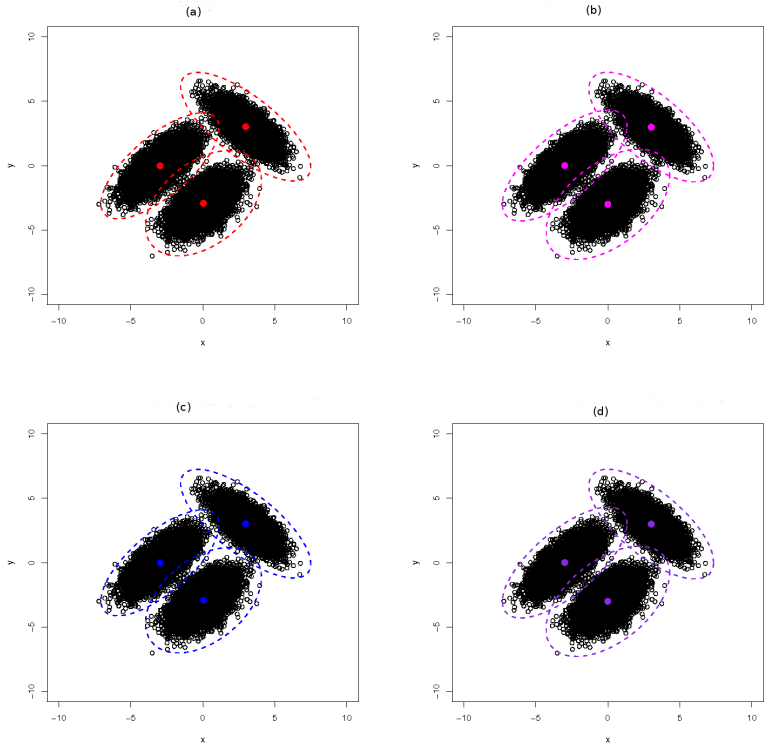

Figura 5.9: Estimación oAMDE para $t = 37,500$ en el primer ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) muestra el proceso con el algoritmo $oAMDE[V3]$. Las Figuras (c) y (d) muestran estos mismos procesos ajustados.

oAMDE para $t = 50,000$.

Se finaliza el ejemplo para el 100 % de los datos. Igual que antes, se eliminan las componentes que no tengan un peso significativo en el mezcla estimada y obtiene el oAMDE en las versiones $V2$ y $V3$. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla estimada de componentes cuyos pesos son significativos. Los resultados se aprecian en la Figura 5.10.

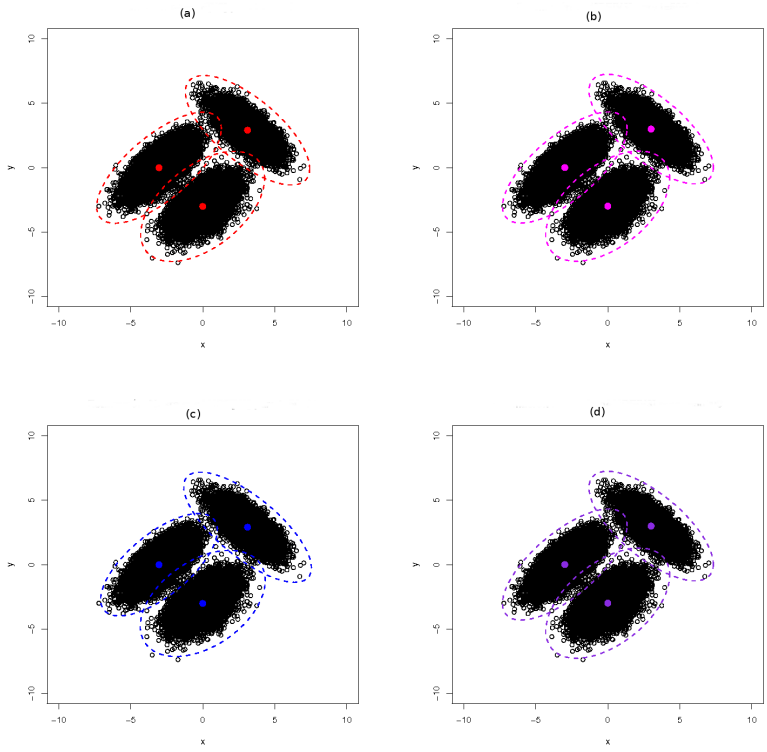

Figura 5.10: Estimación oAMDE para $t = 50,000$ en el primer ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) muestra el proceso con el algoritmo $oAMDE[V3]$. Las Figuras (c) y (d) muestran estos mismos procesos ajustados.

Errores de Estimación para el Primer Ejemplo.

A continuación se presentan los errores de estimación del modelo de mezclas gaussianas. Se midío la similaridad entre el modelo estimado y el modelo original que generó los datos. Puesto que la medida divergencia KL-UT definida en la ecuación 3.49 no es simétrica, se promedian las medidas resultantes en las dos direcciones. Los resultados se muestran la tabla 5.1.

Cant. Datos	**Estimación**	**Cant. Comp**	**Div.KL-UT**	**Dist.H-UT**
$12,500$	AMDE	6	4.092423×10^{-2}	4.351481×10^{-4}
	AMDE-Ajustada	3	2.798531×10^{-2}	2.236741×10^{-3}
$25,000$	oAMDE[V2]-Ajustada	3	9.864958×10^{-2}	2.236741×10^{-3}
	oAMDE[V3]-Ajustada	3	1.891993×10^{-2}	1.307277×10^{-4}
$37,500$	oAMDE[V2]-Ajustada	3	8.791329×10^{-2}	1.909656×10^{-3}
	oAMDE[V3]-Ajustada	3	1.138928×10^{-2}	3.374412×10^{-5}
$50,000$	oAMDE[V2]-Ajustada	3	8.549594×10^{-2}	1.808629×10^{-3}
	oAMDE[V3]-Ajustada	3	1.436978×10^{-2}	4.489006×10^{-5}

Tabla 5.1: Errores de Estimación para el primer ejemplo.

En la tabla se puede apreciar que el ajuste hecho al finalizar el proceso AMDE, disminuye la cantidad de componentes y el valor del error. También que durante todo el ejemplo las estimaciones oAMDE[V3]-Ajustadas producen errores mas pequeños que las estimaciones realizadas con oAMDE[V2]-Ajustadas. El proceso de estimación oAMDE[V2]-Ajustada muestra que al aumentar la cantidad de datos que arriban al modelo, el valor calculado del error disminuye levemente.

5.1.2. Segundo Ejemplo.

Se generan $100,000$ datos para la mezcla gaussiana bivariada con parámetros:

$$\alpha_1 = 0.12 \qquad \alpha_2 = 0.16 \qquad \alpha_3 = 0.20$$

$$\mu_1 = (-15, 18) \qquad \mu_2 = (0, 10) \qquad \mu_3 = (-10, 1)$$

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix} \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \quad \boldsymbol{\Sigma}_3 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

$$\alpha_4 = 0.24 \qquad \alpha_5 = 0.15 \qquad \alpha_6 = 0.13$$

$$\mu_4 = (3, -3) \qquad \mu_5 = (2, -18) \qquad \mu_6 = (18, 2)$$

$$\boldsymbol{\Sigma}_4 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix} \quad \boldsymbol{\Sigma}_5 = \begin{pmatrix} 0.7 & 0 \\ 0 & 0.7 \end{pmatrix} \quad \boldsymbol{\Sigma}_6 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}$$

En la Figura 5.11 se muestran los datos y la elipse de confinaza del $99.99\,\%$ para los parámetros dados.

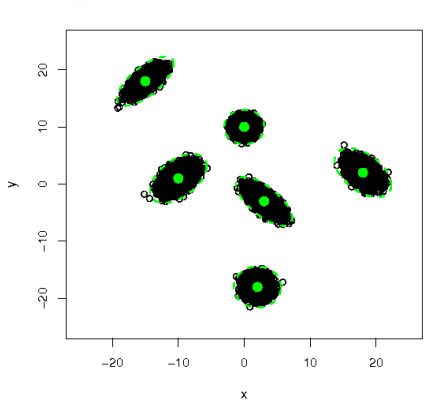

Figura 5.11: Datos y elipses de confinza del $99.99\,\%$ para el segundo ejemplo con $d = 2$

Estimación del Umbral de Creación.

De la misma forma que en el primer ejemplo, se tomaron 6 muestras aleatorias de tamaño $n = 10000$ y se obtuvo el valor $\hat{Tc} = 15.6036$. En la Figura 5.12 se observan las 6 muestras tomadas, las elipses de confianzas y las esferas promedio para cada muestra del segundo ejemplo. Los radios al cuadrado de cada esfera son los estimadores del umbral de creación $\hat{Tc}$.

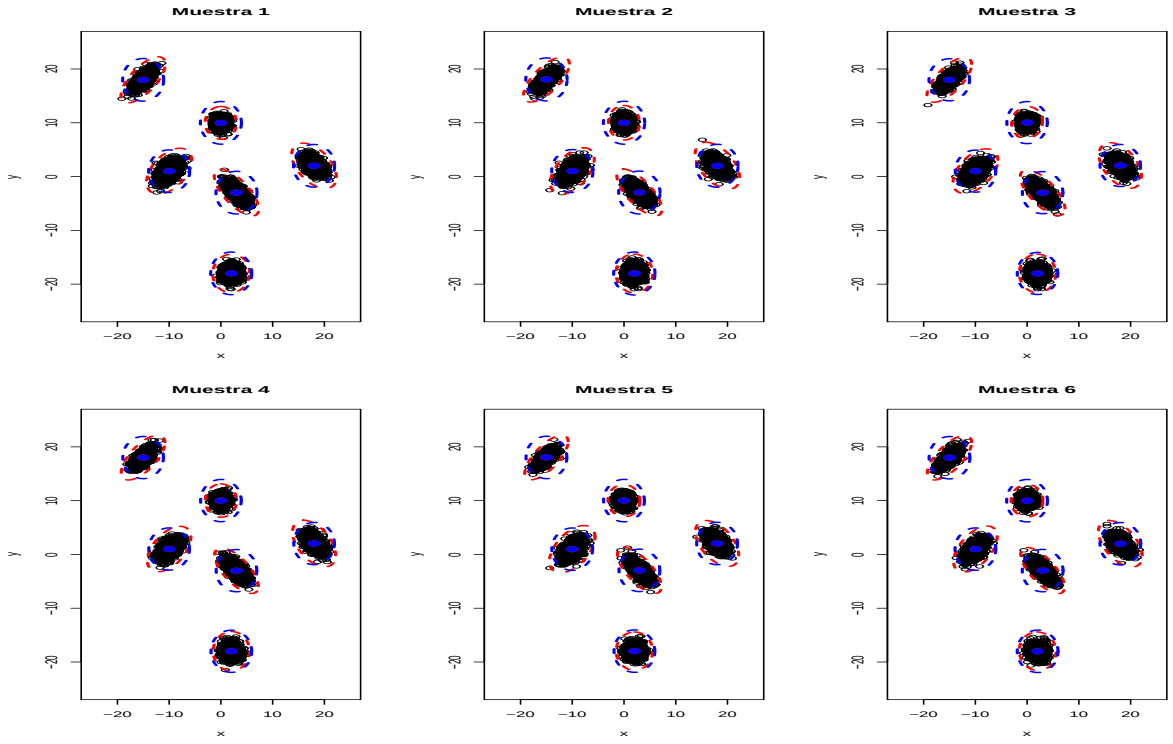

Figura 5.12: Datos, elipses de confianzas y esferas para cada muestra del segundo ejemplo

En la Figura 5.13 se observan los valores de los diferentes estimadores de Tc para cada muestra. Aunque son muy similares estas cantidades, se toma el promedio de este valor como el estimador final $\hat{Tc}$.

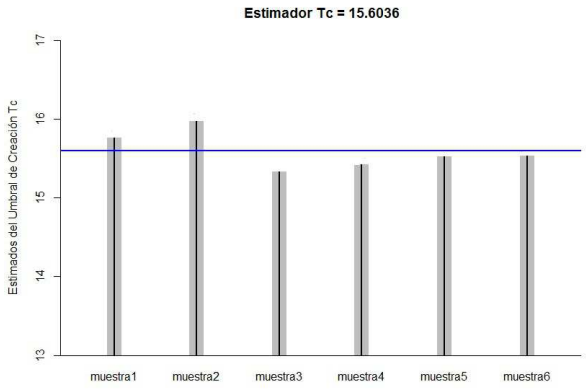

Figura 5.13: Estimadores del umbral de creación $\hat{Tc}$ para cada muestra en el primer ejemplo.

Proceso AMDE.

El arribo de cada observación $x^{[t]}$ se hace de forma secuencial uno a uno. Aquí se presenta el modelo de mezcla estimado después de $25,000$ datos. Los resultados se muestran en la Figura 5.4. En la parte izquierda de la Figura se observa la estimación para los $25,000$ datos con una mezcla gaussiana bivariada de 16 componentes. Si eliminamos de la mezcla aquellas componentes cuyo peso sea menor que la tolerancia (tol) establecida anteriormente, se obtiene el modelo de la parte derecha de la Figura, el cual tiene 13 componentes.

Igual que en el ejemplo anterior, las componentes en *exceso* tienden a estar cerca

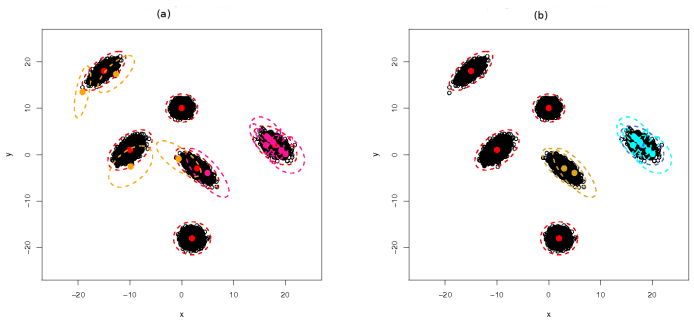

Figura 5.14: Estimación AMDE para $t = 25,000$ en el segundo ejemplo. La Figura (a) muestra todas las componentes estimadas en el proceso AMDE. La figura (b) muestra las componentes significativas para el modelo estimado.

de las *componentes objetivo*, tal y como se muestran en la Figura 5.14. Luego se hace natural pensar en *agrupar componentes*. Acá se ordenan las componentes halladas en el proceso AMDE por su peso, en forma decreciente, tal como se muestra en la Figura 5.15.

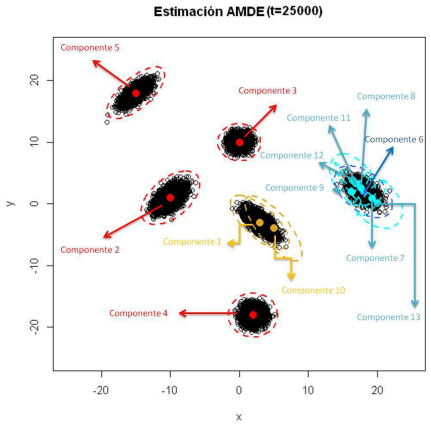

Figura 5.15: Etiquetado de las componentes de la mezcla en el proceso AMDE en el segundo ejemplo.

Para cada par de componentes encontradas en el proceso AMDE, se calcula la *distancia de Mahalanobis orientada* obteniendo asi la matriz

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.00	225.83	237.30	480.75	456.40	710.86	736.23	725.85	589.76	4.48	714.86	714.59	753.47
2	225.83	0.00	119.49	1011.00	545.01	1049.18	1189.35	995.83	918.83	448.43	871.31	922.23	1291.60
3	237.30	119.49	0.00	1580.26	556.45	747.68	872.54	700.81	632.07	436.37	590.94	635.67	963.90
4	480.75	1011.00	1580.26	0.00	2313.73	949.52	945.77	973.54	876.20	300.65	997.42	982.57	946.19
5	456.40	545.01	556.45	2313.73	0.00	4110.55	4502.86	3940.21	3804.96	2934.43	3564.56	3723.49	4782.59
6	710.86	1049.18	747.68	949.52	4110.55	0.00	1.89	0.51	4.10	391.88	4.31	2.14	5.37
7	736.23	1189.35	872.54	945.77	4502.86	1.89	0.00	89.35	156.75	1546.35	262.00	177.22	20.39
8	725.85	995.83	700.81	973.54	3940.21	0.51	89.35	0.00	71.48	5791.74	203.42	67.91	870.05
9	589.76	918.83	632.07	876.20	3804.96	4.10	156.75	71.48	0.00	48045.08	509.92	534.30	1006.88
10	4.48	448.43	436.37	300.65	2934.43	391.88	1546.35	5791.74	48045.08	0.00	699.07	705.31	783.12
11	714.86	871.31	590.94	997.42	3564.56	4.31	262.00	203.42	509.92	699.07	0.00	0.50	26.50
12	714.59	922.23	635.67	982.57	3723.49	2.14	177.22	67.91	534.30	705.31	0.50	0.00	20.43
13	753.47	1291.60	963.90	946.19	4782.59	5.37	20.39	870.05	1006.88	783.12	26.50	20.43	0.00

Ahora se construye la matriz de adyacencia definida en la sección 4.3.3, donde $\chi^2_{0.9999,2} = 18.42068$, y el respectivo grafo nos indicará los grupos a formar. Estas se pueden ver en la Figura 5.25

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	1	1	0	1	1	1
7	0	0	0	0	0	1	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0	0	0	1	0
12	0	0	0	0	0	1	0	0	0	0	1	0	0
13	0	0	0	0	0	1	0	0	0	0	0	0	0

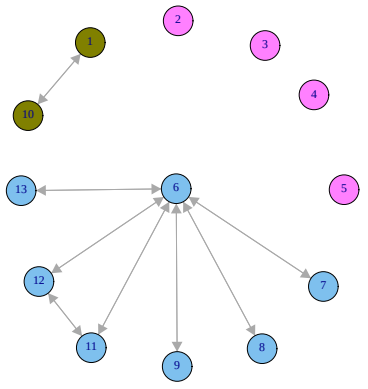

Figura 5.16: Matriz de adyacencia y grafo para la estimación AMDE en el segundo ejemplo.

Luego, las componentes agrupadas en submezclas se comprimen según lo visto en la sección 4.3.3, obteniendo una estimación de 6 componentes tal y como se muestra

en la Figura 5.17.

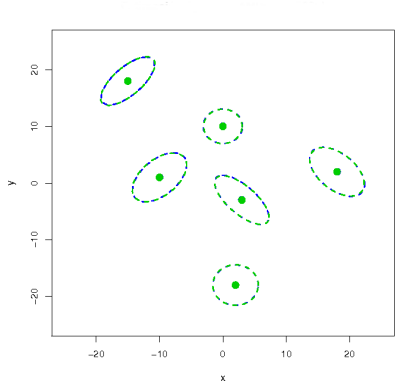

Figura 5.17: Ajuste de Componentes en AMDE para el segundo ejemplo.

Proceso oAMDE

Aquí se presenta el proceso oAMDE para el 50 %, 75 % y 100 % de los datos. Al finalizar cada una de estas etapas, de ser necesario, se ajustan las mezclas estimadas con el proceso visto en la sección. 4.3.3

oAMDE para $t = 50,000$.

En esta parte de la tesis se implementaron los algoritmos $oAMDE[V2]$ y $oAMDE[V3]$. Pasado los primeros $25,000$ datos, es decir el 50 % del total generado, se eliminaron las componentes que no tenían un peso significativo en el mezcla estimada. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla esimada sin las componentes con pesos significativos. Los resultados se aprecian en la Figura 5.18.

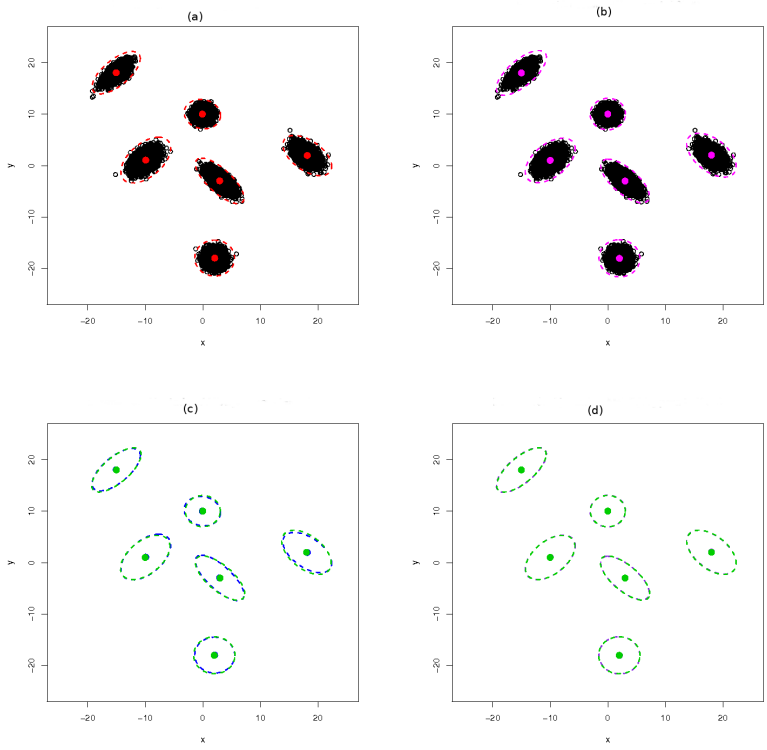

Figura 5.18: Estimación oAMDE para $t = 50,000$ en el segundo ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) muestra el proceso con el algoritmo $oAMDE[V3]$. Las Figuras (c) y (d) muestran estos mismos procesos ajustados.

oAMDE para $t = 75,000$.

Ahora se muestra el proceso transcurrido con el 75 % de los datos. Se eliminan las componentes que no tienen un peso significativo en la mezcla estimada y obtiene el oAMDE en las versiones de Cappé. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla esimada sin las componentes con pesos significativos. Los resultados se aprecian en la Figura 5.19.

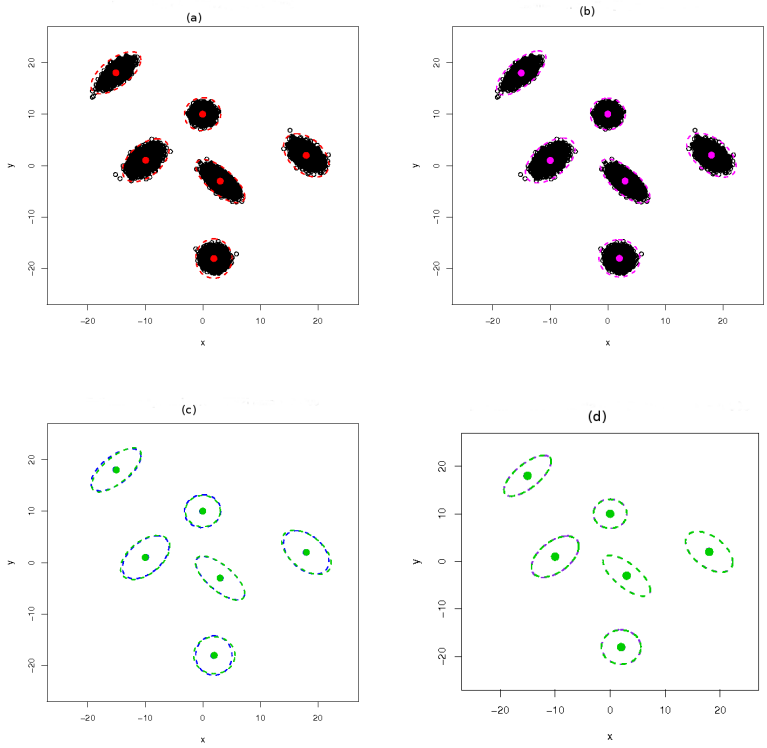

Figura 5.19: Estimación oAMDE para $t = 75,000$ en el segundo ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) muestra el proceso con el algoritmo $oAMDE[V3]$. Las Figuras (c) y (d) muestran estos mismos procesos ajustados.

oAMDE para $t = 100,000$.

Se finaliza el ejemplo para el 100 % de los datos. Igual que antes se eliminan las componentes que no tengan un peso significativo en el mezcla estimada y obtiene el oAMDE en las versiones $V2$ y $V3$. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla estimada de componentes cuyos pesos son significativos. Los resultados se aprecian en la Figura

5.20.

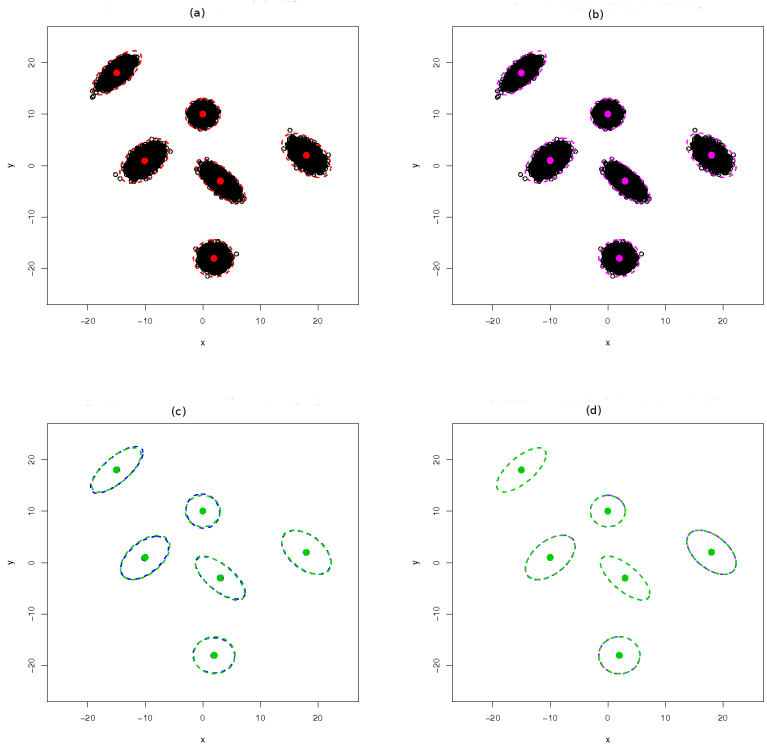

Figura 5.20: Estimación oAMDE para $t = 100,000$ en el segundo ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) muestra el proceso con el algoritmo $oAMDE[V3]$. Las Figuras (c) y (d) muestran estos mismos procesos ajustados.

Errores de Estimación para el Segundo Ejemplo.

A continuación se presentan los errores de estimación del modelo de mezclas gaussianas. Se midío la similaridad entre el modelo estimado y el modelo original que generó los datos. Puesto que la medida divergencia KL-UT definida en la ecuación 3.49 no es simétrica, se promedia las medidas resultantes entre el modelo estimado y

el modelo original. Los resultados se muestran la tabla 5.2.

Cant. Datos	**Estimación**	**Cant. Comp**	**Div.KL-UT**	**Dist.H-UT**
$25,000$	AMDE	17	4.129972×10^{-2}	3.584033×10^{-4}
	AMDE-Ajustada	6	1.117705×10^{-2}	1.615287×10^{-3}
$50,000$	oAMDE[V2]-Ajustada	6	1.117705×10^{-1}	3.012714×10^{-3}
	oAMDE[V3]-Ajustada	6	1.889491×10^{-2}	1.228301×10^{-4}
$75,000$	oAMDE[V2]-Ajustada	6	9.829458×10^{-2}	3.161018×10^{-3}
	oAMDE[V3]-Ajustada	6	2.097185×10^{-2}	9.565701×10^{-5}
$100,000$	oAMDE[V2]-Ajustada	6	9.808798×10^{-2}	2.292152×10^{-3}
	oAMDE[V3]-Ajustada	6	1.408252×10^{-2}	4.364152×10^{-5}

Tabla 5.2: Errores de Estimación para el segundo ejemplo.

En la tabla se puede apreciar que el ajuste hecho al finalizar el proceso AMDE, disminuye la cantidad de componentes y el valor del error. También que durante todo el ejemplo las estimaciones oAMDE[V3]-Ajustadas producen errores más pequeños que las estimaciones realizadas con oAMDE[V2]-Ajustadas. El proceso de estimación oAMDE[V2]-Ajustada muestra que al aumentar la cantidad de datos que arriban al modelo, el valor calculado del error disminuye levemente.

Con estos dos ejemplos simulados en dimensión $d = 2$, se pudo observar que aunque los resultados obtenidos con el proceso oAMDE[V3] son ligemente mejores, en cuanto a que producen en general errores menores que los obtenidos en el proceso oAMDE[V2], este último es más deseable en un flujo de datos ya que no depende de estimaciones anteriores, como sí lo hace el proceso oAMDE[V3]. Además, bajo la propuesta hecha

acá con los ajustes de la cantidad de componentes realizada con grafos, es más fácil de adecuar ésta a el proceso oAMDE[V2] que al proceso oAMDE[V3] . Los ejemplos a continuación trabajan sólo para el proceso oAMDE[V2] con sus respectivos ajustes.

5.2. Ejemplos para $d = 3$.

5.2.1. Tercer Ejemplo.

Se generan $50,000$ datos para la mezcla gaussiana con $d = 3$ con parámetros:

$$\alpha_1 = 0.42 \qquad \alpha_2 = 0.28 \qquad \alpha_3 = 0.30$$

$$\mu_1 = (3,3,3) \qquad \mu_2 = (3,0,-3) \qquad \mu_3 = (-3,0,0)$$

$$\mathbf{\Sigma}_1 = \begin{pmatrix} 1 & 0.7 & 0.5 \\ 0.7 & 1 & 0.7 \\ 0.5 & 0.7 & 1 \end{pmatrix} \quad \mathbf{\Sigma}_2 = \begin{pmatrix} 1 & -0.7 & 0.5 \\ -0.7 & 1 & -0.7 \\ 0.5 & -0.7 & 1 \end{pmatrix} \quad \mathbf{\Sigma}_3 = \begin{pmatrix} 1 & -0.5 & -0.7 \\ -0.5 & 1 & 0.7 \\ -0.7 & 0.7 & 1 \end{pmatrix}$$

En la Figura 5.21 se muestran los datos y los elipsoides de confianza del $99.99\,\%$ para los parámetros dados.

Estimación del Umbral de Creación.

En la Figura 5.23 se observan las 6 muestras de tamaño $n = 5000$, los elipsoides de confianza y las esferas. Los radios al cuadrado de cada esfera son los estimadores del umbral de creación $\hat{Tc}$, el cual tuvo el valor de 18.32522.

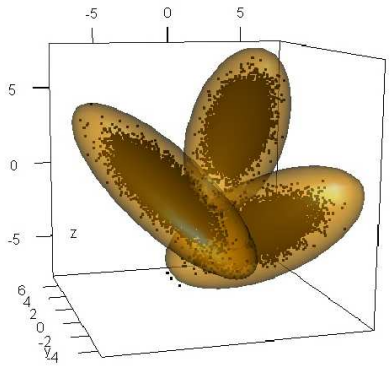

Figura 5.21: Datos y elipsoides de confinza del 99.99 % para el tercer ejemplo con $d = 3$

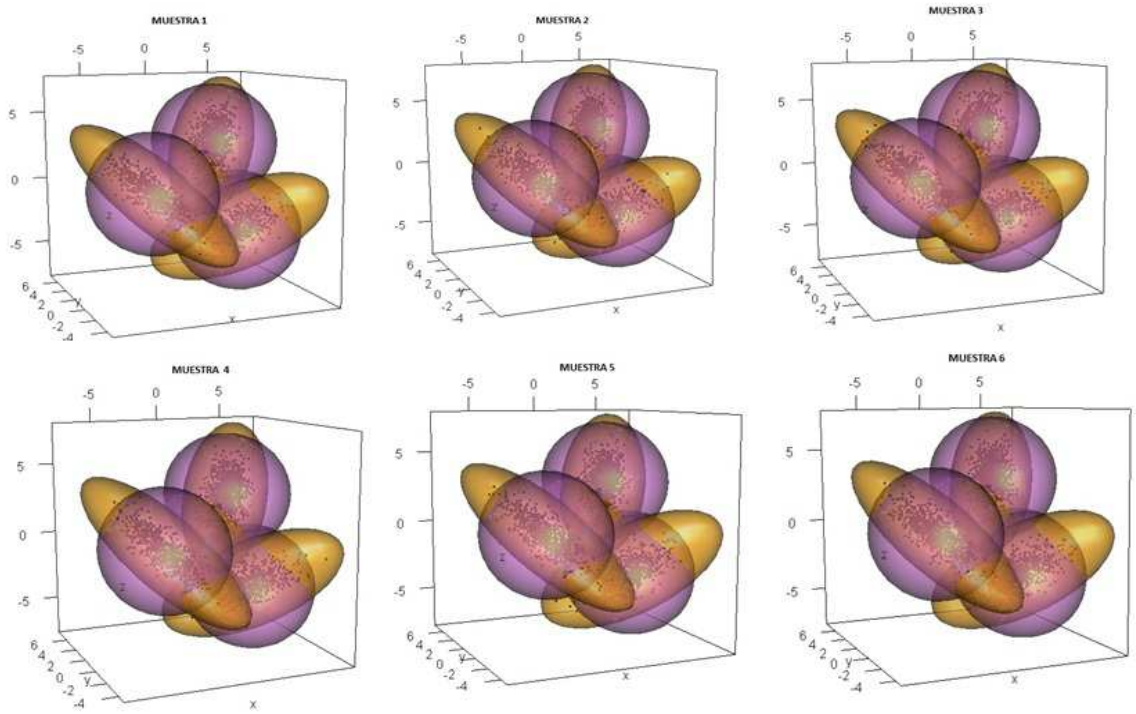

Figura 5.22: Datos, elipsoides de confianzas y esferas promedio para cada muestra del tercer ejemplo

En la Figura 5.23 se observan los valores de los diferentes estimadores de Tc para cada muestra. Aunque son muy similares estas cantidades, se toma el promedio de este valor como el estimador final $\hat{Tc}$.

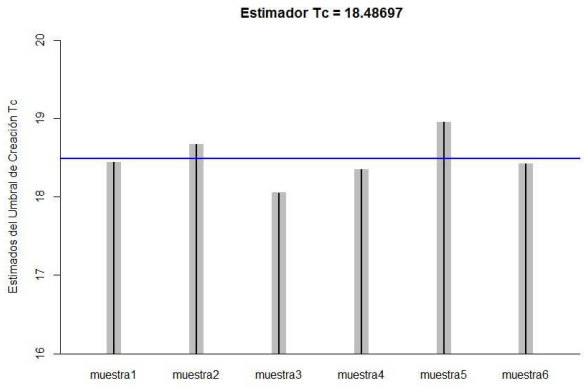

Figura 5.23: Estimadores del umbral de creación $\hat{Tc}$ para cada muestra en el tercer ejemplo.

Proceso AMDE.

Aquí se presenta el modelo de mezcla estimado después de $10,000$ datos. Los resultados se muestran en la Figura 5.24. Allí se observa la estimación para los primeros $10,000$ datos con una mezcla gaussiana de 11 componentes, donde se eliminaron aquellas componentes cuyo peso son menor que la tolerancia (tol) establecida anteriormente. Igual que en el ejemplo anterior, las componentes en *exceso* tienden a estar cerca de las *componentes objetivo*. Luego se hace natural pensar en *agrupar componentes*. Acá se ordenan las componentes halladas en el proceso AMDE por su peso, en forma decreciente, tal y como se realizó en los ejemplos anteriores.

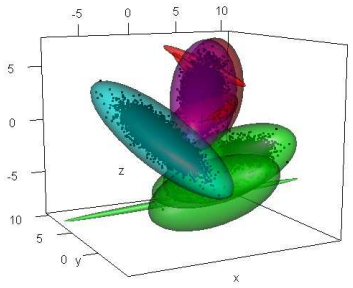

Figura 5.24: Estimación AMDE para $t = 10,000$ en el tercer ejemplo.

Para cada par de componentes encontradas en el proceso AMDE, se calcula la *distancia de Mahalanobis orientada* obteniendo así la matriz

	1	2	3	4	5	6	7	8	9	10	11
1	0.0000000	39.79911	51.098455	82.892061	2.760964	63.861117	0.8788414	4.900576	219.04175	19.17588	79.92967
2	39.7991125	0.00000	47.250696	65.342025	78.212216	50.604682	178.8696629	251.286943	185.70587	265.03299	42.30845
3	51.0984546	47.25070	0.000000	3.767984	70.501823	4.329025	213.5207022	321.087606	25.78074	318.36795	21.01287
4	82.8920605	65.34203	3.767984	0.000000	142.023674	5.961869	341.3672966	495.135509	20.37268	437.56547	23.98943
5	2.7609642	78.21222	70.501823	142.023674	0.000000	271.614376	57.6678915	148.736559	762.02389	52.59023	481.61366
6	63.8611166	50.60468	4.329025	5.961869	271.614376	0.000000	211.9671506	321.714240	25.95179	291.83629	33.16826
7	0.8788414	178.86966	213.520702	341.367297	57.667891	211.967151	0.0000000	267.764722	8215.47512	605.97968	7381.45060
8	4.9005760	251.28694	321.087606	495.135509	148.736559	321.714240	267.7647216	0.000000	10617.53875	223.18576	8993.52119
9	219.0417524	185.70587	25.780737	20.372679	762.023894	25.951791	8215.4751228	10617.538746	0.00000	20075.66677	2333.94609
10	19.1758798	265.03299	318.367950	437.565470	52.590232	291.836294	605.9796781	223.185765	20075.66677	0.00000	70.70449
11	79.9296662	42.30845	21.012867	23.989435	481.613665	33.168262	7381.4505963	8993.521186	2333.94609	70.70449	0.00000

Ahora se construye la matriz de adyacencia definida en la sección 4.3.3, donde $\chi^2_{0.9999,3} = 21.10751$ y el respectivo grafo que nos indicará los grupos a formar. Estas se pueden ver en la Figura 5.25

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	1	0	1	1	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	1	0	0	0	0	1
4	0	0	1	0	0	1	0	0	1	0	0
5	1	0	0	0	0	0	0	0	0	0	0
6	0	0	1	1	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	1	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	0
11	0	0	1	0	0	0	0	0	0	0	0

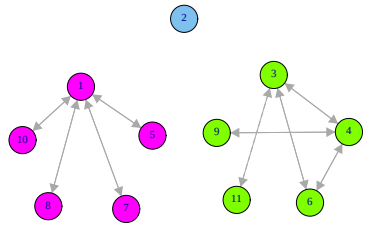

Figura 5.25: Matriz de adyacencia y grafo para la estimación AMDE en el tercer ejemplo.

Luego, las componentes agrupadas en submezclas se comprimen según lo visto en la sección 4.3.3, obteniendo una estimación de 3 componentes tal y como se muestra en la Figura 5.26.

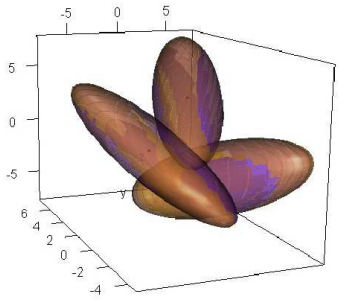

Figura 5.26: Ajuste de componentes en AMDE para el tercer ejemplo.

Proceso oAMDE

Aquí se presenta el proceso oAMDE para el 40 %, 60 %, 80 % y 100 % de los datos. Al finalizar cada una de estas etapas, de ser necesario, se ajustan las mezclas estimadas con el proceso visto en la sección 4.3.3.

oAMDE para $t = 20,000$.

En esta parte de la tesis se implementó el algoritmos $oAMDE[V2]$. Pasados los primeros $20,000$ datos, es decir el 40 % del total generado, se eliminaron las componentes que no tenían un peso significativo en la mezcla estimada. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla estimada de componentes cuyos pesos son significativos. Los resultados se aprecian en la Figura 5.27.

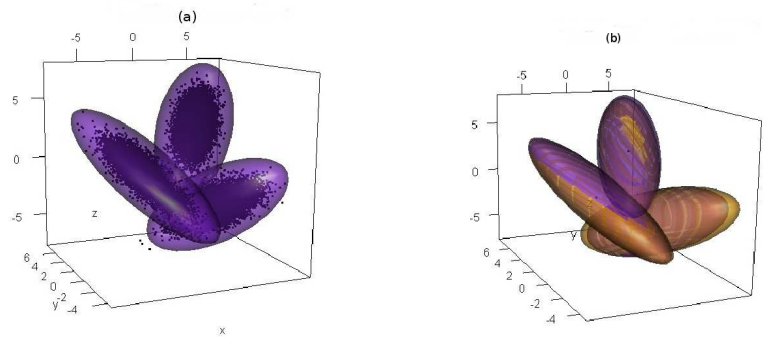

Figura 5.27: Estimación oAMDE para $t = 20,000$ en el tercer ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) el modelo ajustado, que para el caso es el mismo.

oAMDE para $t = 30,000$.

Ahora se muestra el proceso transcurrido con el 60 % de los datos. Se eliminan las componentes que no tienen un peso significativo en la mezcla estimada y obtiene el oAMDE en las versiones de Cappé. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla estimada de componentes cuyos pesos son significativos. Los resultados se aprecian en la Figura 5.28.

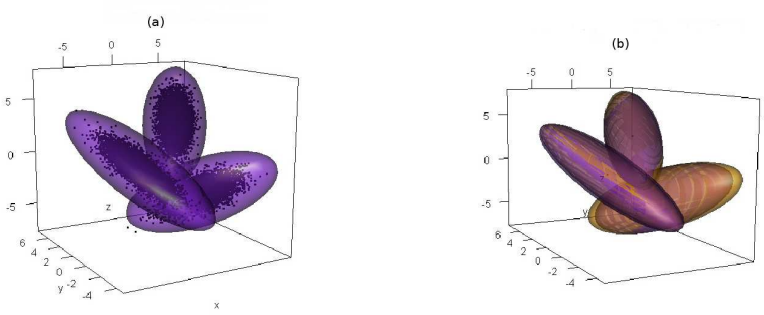

Figura 5.28: Estimación oAMDE para $t = 30,000$ en el tercer ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) el modelo ajustado, que para el caso es el mismo.

oAMDE para $t = 40,000$.

Ahora se analiza el modelo de mezcla estimado para el 80 % de los datos. Igual que antes, se eliminan las componentes que no tengan un peso significativo en el mezcla estimada y obtiene el oAMDE en la versión V_2. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla estimada de componentes cuyos pesos son significativos. Los resultados se aprecian en la Figura

5.29.

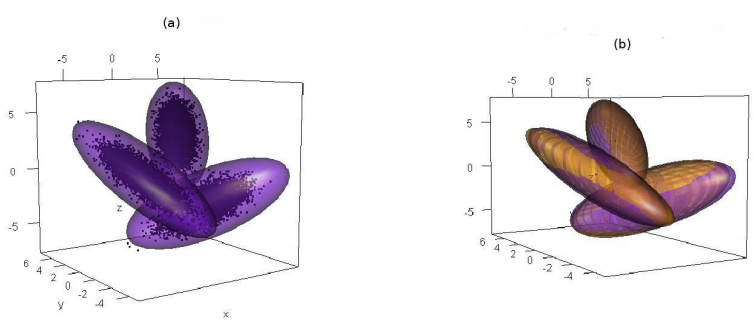

Figura 5.29: Estimación oAMDE para $t = 40,000$ en el tercer ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) el modelo ajustado, que para el caso es el mismo.

oAMDE para $t = 50,000$.

Se finaliza el ejemplo para el 100 % de los datos. Igual que antes se eliminan las componentes que no tengan un peso significativo en el mezcla estimada y obtiene el oAMDE en las versión V_2. No se hizo necesario aplicar el proceso de ajuste con grafos, por lo que la mezcla ajustada es la misma que la mezcla estimada sin las componentes con pesos significativos. Los resultados se aprecian en la Figura 5.30.

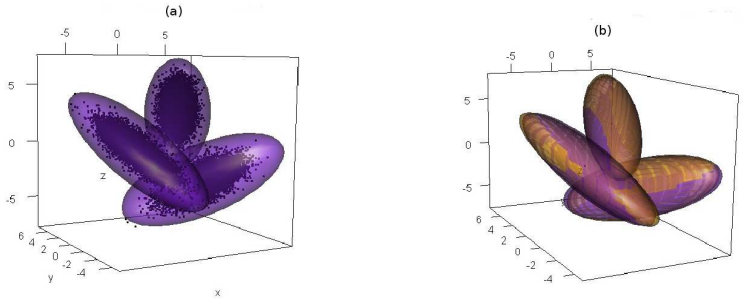

Figura 5.30: Estimación oAMDE para $t = 50,000$ en el tercer ejemplo. La Figura (a) muestra el proceso con el algoritmo $oAMDE[V2]$ y la Figura (b) el modelo ajustado, que para el caso es el mismo.

Errores de Estimación para el Tercer ejemplo.

A continuación se presentan los errores de estimación del modelo de mezclas gaussianas. Se midió la similaridad entre el modelo estimado y el modelo original que generó los datos. Puesto que la medida divergencia KL-UT definida en la ecuación 3.49 no es simétrica, se promedian las medidas resultantes entre el modelo estimado y el modelo original. Los resultados se muestran el tabla 5.3.

Cant. Datos	**Estimación**	**Cant. Comp**	**Div.KL-UT**	**Dist.H-UT**
$10,000$	AMDE	11	5.954973×10^{-2}	9.68047×10^{-4}
	AMDE-Ajustada	3	3.472086×10^{-2}	2.059552×10^{-4}
$20,000$	oAMDE[V2]-Ajustada	3	1.05195×10^{-1}	2.190078×10^{-3}
$30,000$	oAMDE[V2]-Ajustada	3	1.394823×10^{-1}	3.321876×10^{-3}
$40,000$	oAMDE[V2]-Ajustada	3	1.170076×10^{-1}	2.33802×10^{-3}
$50,000$	oAMDE[V2]-Ajustada	3	9.573998×10^{-2}	1.634479×10^{-3}

Tabla 5.3: Errores de Estimación para el Tercer Ejemplo.

En la tabla se puede apreciar que el ajuste hecho al finalizar el proceso AMDE, disminuye la cantidad de componentes y el valor del error. El proceso de estimación oAMDE[V2]-Ajustada muestra que al aumentar la cantidad de datos que arriban al modelo, el valor calculado del error disminuye levemente.

CAPÍTULO 6

CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

- Las mezclas adaptativas apliacadas en flujos de datos presentan problemas en la complejidad de los modelos estimados (crea componentes en exceso). En esta tesis se presentó una forma no empírica de hallar un estimador del umbral de creación Tc, el cual mostró ser bueno para el control en el crecimiento del número componentes de la mezcla estimada.

- La eliminación de componentes no significativas, junto con el ajuste hecho con

agrupaciones de componentes usando grafos para el modelo estimado en algun tiempo t, mostró ser un buen método para reducir la complejidad del modelo estimado.

- El proceso de estimación en línea oAMDE[V3] mostró menor valor en el error medido entre la mezcla objetivo y la mezcla estimada, sin embargo bajo el ajuste propuesto aquí es un procedimiento poco aconsejable en aplicaciones reales, ya que depende de varias estimaciones anteriores , lo que en un proceso a largo plazo contradice el sentido de las mezclas adaptativas.

- El proceso de estimación en línea oAMDE[V2], junto con el ajuste propuesto, presenta un buen rendimiento. Este procedimiento es más recomendable para trabajos con datos reales, ya que reduce la cantidad de procesos a ejecutar.

- Las estimaciones de densidades para flujo de datos, que no usan todos los datos que han llegado en pasos anteriores, son más deseables que los métodos tradicionales, ya que con buenos algoritmos y buenos programas, se mejora la calidad de estimación.

- El lenguaje de programación R es aún muy escaso en recursos para enfrentar análisis de datos de este tipo. El problema de la dimensionalidad es un factor que aún afecta la implementación de algunos de los paquetes aquí usados.

6.2. Trabajos Futuros

- Adecuar las ecuaciones de compresión de componentes expuestas en esta tesis, e implementadas en el ajuste con agrupaciones usando grafos, para ser usadas de

forma eficiente en el proceso oAMDE[V3].

- Mejorar los programas empleados para mejorar los tiempos de procesamiento.
- Usar diversos lenguajes de programación para ejecutar los procesos aqui presentados y comparar la eficiencia y ganancia de éstos en cada entorno.
- Comparar el proceso oAMDE[V2] bajo el ajuste propuesto aquí, con otros algoritmos que usen la misma idea de actualización secuencial del modelo para flujos de datos.
- Implementar estos ajustes a los procesos en datos reales y a densidades con pesos cambiantes.

BIBLIOGRAFÍA

[1] D. B. Rubin A. P. Dempster, N. M. Laird. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Serie B*, 39:1–38, 1977.

[2] E. Acuña. Clasificación usando estimación de densidad por kernel. Notas de Clase, 2004.

[3] Y. Zhu B. Han, D. Comaniciu. Sequential kernel density approximation and its application to real-time visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 30:1186 –1197, 2008.

[4] A.E. Raftery C. Fraley. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[5] B. Seeger C. Heinz. Cluster kernels: Resource-aware kernel density estimators over streaming data. *Knowledge and Data Engineering, IEEE Transactions*, 20:880 –

893, 2008.

[6] T. Duong. Bandwidth selectors for multivariate kernel density estimation. *Bulletin of the Australian Mathematical Society*, 71:351, 2005.

[7] T. Duong. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, 21:1–16, 2007.

[8] H. Greenspan J. Goldberger, S. Gordon. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference*, volume 1, pages 487 –493, 2003.

[9] J. Dreyfuss J. Goldberger, H.K. Greenspan. Simplifying mixture models using the unscented transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 30:1496 –1502, 2008.

[10] S. Roweis J. Goldberger. Hierarchical clustering of a mixture model. In Léon Bottou Saul Lawrence, Yair Weiss, editor, *Advances in Neural Information Processing Systems*, pages 505–512. MIT Press, Cambridge, MA, 2005.

[11] J. Picek J. Jurečková. *Robust Statistical Methods with R.* Chapman & Hall/CRC, Reino Unido, 2006.

[12] I.Dhillon J.V. Davis. Differential entropic clustering of multivariate gaussians. In *Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2006.

[13] D. M. Titterington K. Mengersen, C.P. Robert. *Mixtures : Estimation and applications*. Wiley ; John Wiley [distributor], Hoboken, N.J. : Chichester :Reino Unido, 2011.

[14] J. Klemelä. *Smoothing of Multivariate Data: Density Estimation and Visualization*. Wiley Series in Probability and Statistics. John Wiley & Sons, Canada, 2009.

[15] T. Downs M. Gallagher, M. Frean. *Real-valued Evolutionary Optimization using a Flexible Probability Density Estimator*, volume 1, pages 840–846. Morgan Kaufmann, 1999.

[16] A. Leonardis M. Kristan, D. Skocaj. Multivariate online kernel density estimation with gaussian kernels. *Pattern Recognition*, 44:2630 – 2642, 2011.

[17] D. Skocaj M. Kristan, A. Leonardis. Supplemental online material for the paper: Multivariate online kernel density estimation with gaussian kernels. 2011.

[18] S. Ishii M. Sato. On-line em algorithm for the normalized gaussian network. *Neural Computation*, 12:407–432, 2000.

[19] H. Wang M. Song. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. *Proceedings of SPIE*, 5803:174–183, 2005.

[20] A. Miñarro. Estimación no paramétrica de la función de densidad. Notas de Clase, 1998.

[21] H.J. Moyano. Mezclas finitas de distribuciones normales: Una alternativa para clasificar. Universidad Industrial de Santander, Colombia, 2007.

[22] M.C. Jones M.P. Wand. *Kernel Smoothing*. Monographs on Statistics and Applied Probability. Chapman & Hall, Reino Unido, 1995.

[23] R. Cipolla O. Arandjelovic. Incremental learning of temporally-coherent gaussian mixture models. In *BMVC*, 2005.

[24] E. Moulines O. Cappé. On-line expectation-maximization algorithm for latent data models. *Journal Of The Royal Statistical Society Series B*, 71:593–613, 2009.

[25] G. Hulten P. Domingos. A general framework for mining massive data stream. *Journal of Computational and Graphical Statistics*, 12:2003, 2003.

[26] D. Klein P. Liang. Online EM for unsupervised models. In *North American Association for Computational Linguistics (NAACL)*, 2009.

[27] D. Peña. *Análisis de Datos Multivariantes*. McGraw Hill, España, 2002.

[28] C. E. Priebe. Adaptive Mixtures. *Journal of the American Statistical Association*, 89:796–806, 1994.

[29] G.E. Hinton R. Neal. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[30] D.W. Wichern R.A. Johnson. *Applied Multivariate Statistical Analysis*. Series in Statistics. Prentice-Hall, USA, 2001.

[31] J.S. Almeida S. Vinga. Convolution integrals of normal distribution functions. supplementary material to: Rényi continuous entropy of dna sequences:. *Journal of Theoretical Biology*, 231:377 – 388, 2004.

[32] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability. Chapman and Hall, Reino Unido, 1986.

[33] J.S. Simonoff. *Smoothing Methods in Statistics*. Series in Statistics. Springer, USA, 1996.

[34] J.K. Uhlmann S.J. Julier. New extension of the kalman filter to nonlinear systems. *Proceedings of SPIE*, 3:182–193, 1997.

[35] W. F. Szewczyk. Time-evolving adaptive mixtures. *National Security Agency Tech Rep*, (January 1959):1–21, 2005.

[36] M. Hazelton T. Duong. Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of Nonparametric Statistics*, 15:17–30, 2003.

[37] G. R. Terrell. The maximal smoothing principle in density estimation. *Journal of the American Statistical Association*, 85:470–477, 1990.

[38] D.M. Titterington. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society Series B Methodological*, 46:257–267, 1984.

[39] Nenadic O. W. Zucchini, A. Berzel. Applied smoothing techniques. Institute for Statistics and Econometrics, University of Gottingen, Germany, 2005. Lecture notes.

[40] A.R. Martínez W.L. Martínez. *Computational Statistics Handbook with MATLAB*. Computer Science and Data Analysis Series. Chapman & Hall/CRC, Reino Unido, 2002.