CAPITULO III: Mezcla de Gaussianas para clasificación

3.1 Modelos de Mezcla

En un contexto estadístico podemos definir un modelo de mezcla cómo un modelo probabilístico para representar la presencia de subpoblaciones dentro de una misma población. Podemos también decir que un modelo de mezcla corresponde a una distribución de mezcla que representa la distribución de probabilidad de alguna observación en la población en general. De cualquier forma mientras sean asociados los problemas de distribución de mezclas con las propiedades o características de una población en general con respecto a esas subpoblaciones, los modelos de mezcla son usados para crear inferencias estadísticas, aproximaciones y predicciones acerca de las propiedades de las subpoblaciones a partir de las observaciones o datos adquiridos de la población estudiada sin necesidad de información que identifique a la sub población.

3.1.1 Modelos de Mezcla Gaussiana (GMM)

En el caso común en que las n- distribuciones sean gaussianas, hablamos de un Modelo de Mezcla Gaussiana el cual definiremos con la abreviación (GMM). Ajustando un modelo de mezcla, si el número n- de distribuciones es conocido, puede ser realizado:

- Un algoritmo de media n- o medianas n-. Estos son algoritmos de clustering y regresan solo los centroides y los límites de las diferentes componentes. Aunque, por supuesto, la varianza puede ser calculada empíricamente después del clustering.
- El algoritmo ExpectationMaximization (EM): Encuentra la Máxima Probabilidad (ML – MaximumLikelihood) estimada para los parámetros del modelo.

El objetivo que se busca realizar con este modelo de mezcla Gaussiana (GMM) es encontrar una aproximación o estimación a partir de sus componentes encontrando un acomodamiento de los datos que contienen las componentes un ejemplo se observa en la Figura 19.

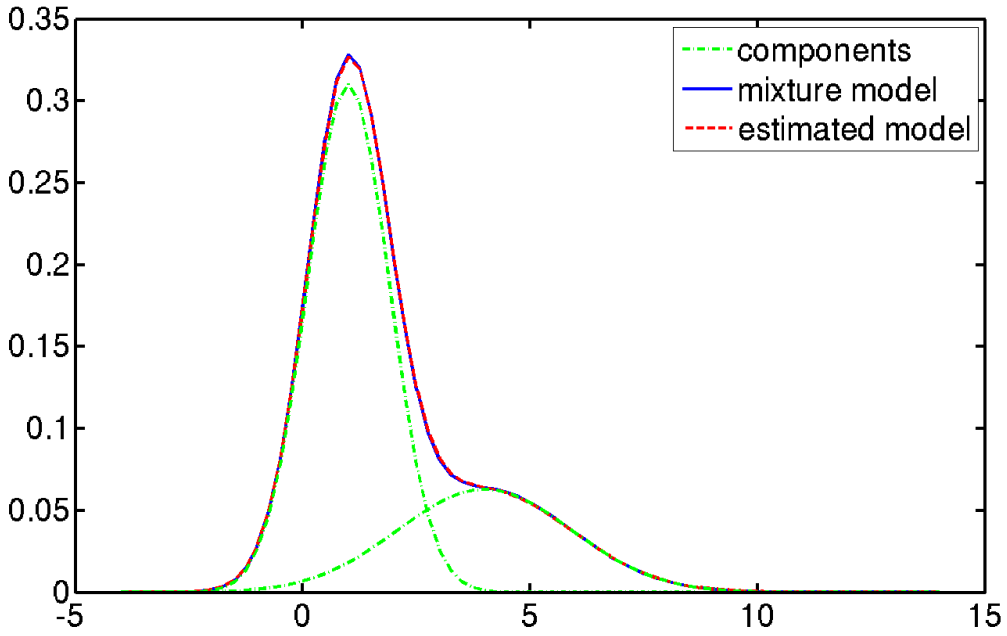

Figura 19: GMM Modelo de Mezcla Gaussiana a partir de dos componentes.

3.2 Árboles de dependencia

Las distribuciones por árbol de dependencia permiten el modelado de primer orden de dependencias entre características. El modelo de árbol de dependencia aplica el modelo de redes bayesianas al asumir dependencias entre características dentro de grupos o familias de datos. En pruebas realizadas utilizando datos simulados, los árboles de dependencia comparados con las redes bayesianas y modelos de dependencia completa resultan más eficaces para encontrar grupos que pertenecen a estructuras de dependencia. En esencia, árboles de dependencia tienen una gran ventaja a comparación de otros modelos de clasificación, la principal es que los árboles no son susceptibles al over-fitting, el cual es un problema bastante frecuente en la estimación de modelos de mezcla de datos dispersos. Los árboles de dependencia nos ofrecen una mejor aproximación de la distribución de datos en relación con el modelo simple de redes bayesianas.

Es necesario comentar también que Charniak (1991) asegura que una de las desventajas de usar teorías de probabilidad es que la especificación completa que requiere una distribución de probabilidad necesita de un número absurdo de datos para cada nodo. Un ejemplo es que para un número n de variables aleatorias se requieren 2^n-1 datos. Supongamos entonces que un modelo que contiene cinco nodos requerirá de 31 datos. Podemos entonces inferir que para modelos con un mayor número de nodos, el número de datos que se manejarían serían muy grandes. Nosotros al trabajar árboles de dependencia de pocos niveles y pocos nodos no requerimos un manejo excesivo de datos, permitiéndonos ocupar éste método sin preocuparnos del número de datos que necesitaremos.

Los árboles de dependencia trabajan bajo un esquema de dirección por jerarquía, donde los nodos del árbol representan características y cada nodo se le asigna una relación con respecto a otro, de esta manera se crea una topología definiendo relaciones entre padres (parent) con mayor jerarquía que su nodo hijo (children). Una topología definida por el árbol dado sería definido con la siguiente ecuación.

$$P(x_i \mid \theta_k)^T = \prod_{j=1}^{p} P(x_{ij} \mid x_{ipa(j)}, \theta_{jk})$$

Donde P(xi|θ) es una distribución condicional Gaussiana y θ, son los parámetros de la distribución condicional. Un ejemplo de distribución de un árbol condicional (Figura 20) nos muestra la relación que existe con los nodos existentes en él. Se observa dependencia de B con A y a su vez, el nodo C y D dependen de B. Así podemos concluir que el nodo A es la raíz del árbol.

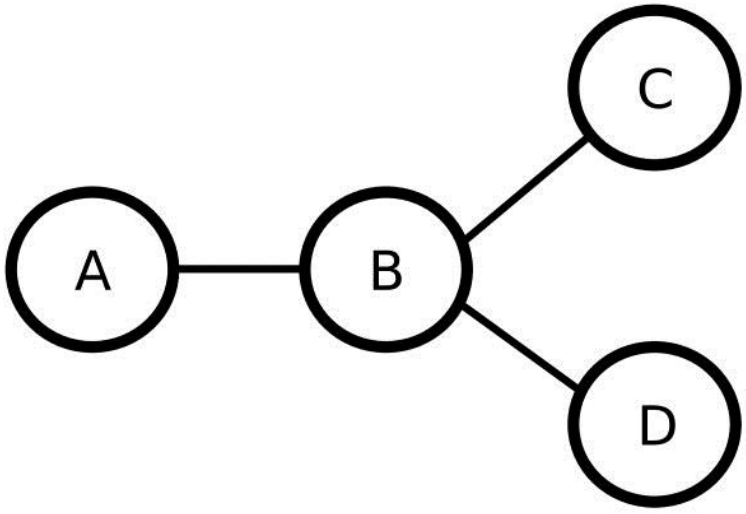

Figura 20: Árbol de dependencia con 4 características con la distribución: $P(x_A, x_B, x_C, x_D) = P(x_A)P(x_B|x_A)P(x_C|x_B)P(x_D|x_B)$. Recuperado de: BMC Bioinformatics "PyMix - The Python mixture package - a tool for clustering of heterogeneous biological data"

Un detalle importante es el mecanismo para obtener la estructura del árbol. Para algunas aplicaciones en particular, la estructura puede ser determinada por un conocimiento previo. En un caso de análisis de expresión de genes de procesos de desarrollo la estructura del árbol se desarrolla por el desarrollo de la célula. Cuando la estructura es desconocida, la estructura con máxima probabilidad (maximum likelihood) puede ser estimada a partir de los datos. Para mi proyecto en concreto hacemos depender los coeficientes de detalle de los de aproximación.

En resumen, un modelo de árbol de dependencia nos brinda una mejor aproximación con distribuciones de nodos. Además la estructura de estimación puede ser muy útil en la vinculación de dependencias importantes entre características dentro de un cluster.

3.3 Algoritmo Expectation Maximization (EM)

El método de Expectation Maximization (EM) nos brinda la oportunidad de adecua o estimar un modelo estadístico en los casos donde los datos estén incompletos o cuando contienen variables desconocidas. En el caso de aproximaciones por modelado de mezcla podemos observar que las variables desconocidas o datos incompletos, nos indican que componentes han generado cada muestra. Para poder trabajar con éste método es necesario estimar los parámetros de las componentes de la mezcla y para eso requerimos datos como lo son la media (mean) y la varianza o desviación estándar. Datos que extraemos directamente de nuestros coeficientes Wavelet.

La librearía de Pymix también nos permite trabajar con éste método donde lo único que debemos desplegar es la siguiente línea de comandos (Figura 21) que se muestra, donde podemos observar que para poder utilizar dicho método es necesario crear un número n de mezclas gaussianas dependientes nombradas como n1, n2, n3, etc. Las cuales adquieren una dependencia anterior a éste método. Los parámetros que se deben meter dentro de la función son los datos como DataSet, nombrados en el ejemplo de abajo como *data*, el número máximo de iteraciones (steps) y finalmente la tolerancia.

```
163
164 #Expectation Maximization
165
166 m = mixture.MixtureModel(3,[0.8,0.1,0.1],[n1,n2,n3])
167 #random.seed(1)
168 data = m.sampleDataSet(100)
169 m.modelInitialization(data)
170 m.EM(data,40,0.01)
171
172
```

Figura 21: Método de Expectation Maximization (EM) en Python

Finalmente el método nos arroja un resultado en término de dos componentes donde se muestra la nueva media y varianza de cada componente. Mostrando el nuevo peso adquirido en cada muestra y los datos necesarios para poder graficar las curvas características para nuestros parámetros que se refieren a los coeficientes Wavelets.

3.4 Clustering

El clustering es un algoritmo de agrupamiento que se efectúa a partir de un conjunto de vectores ordenado por algún criterio o rasgo en común. Los criterios de ordenamiento por lo general se pueden dividir en dos tipos, por variables discretas o similitud. Por lo general los vectores de datos que se encuentran dentro del mismo cluster comparten algún criterio o rasgo o en común. Es una aplicación frecuentemente aplicada en el data mining, se puede lograr una caracterización del grupo a partir de la extracción de características de los integrantes de éste y formando un elemento que represente a la población de ese cluster. Dentro del data mining es considerado como un algoritmo de aprendizaje no supervisado ya que intenta encontrar patrones o similitudes entre variables descriptivas.