Aprendizaje no supervisado

Mtro. José Gustavo Fuentes Cabrera

Facultad de Estudios Superiores

Acatlán

Apuntes de Análisis Multivariado

Licenciatura en Actuaría

Índice

1. Introducción a modelación no supervisada

En esta sección estudiaremos lo correspondiente a modelación no supervisada. En el módulo previo, donde teníamos un escenario de modelación supervisada, nuestro conjunto de entrenamiento era de la forma:

$$\{\left(x^{(i)}, y^{(i)}\right), i = 1, 2, \ldots, n\}$$

En virtud de lo anterior, nuestra tarea de aprendizaje automático (en lo sucesivo aprendizaje) consistía en aprender un mapeo entre el espacio de observaciones (matriz de predictores) y las correspondientes respuestas (variable objetivo). En esta línea de pensamiento, en aprendizaje no supervisado contamos únicamente con el conjunto de la forma:

$$S_n = \{x^{(i)}, i = 1, 2, \ldots, n\}$$

Es natural cuestionar en este momento, ¿Cuál es ahora la tarea de aprendizaje? En este caso, el aprendizaje no supervisado tiene por objeto descubrir la estructura subyacente, informativa y útil en el conjunto S_n, donde el análisis de conglomerados (clustering) es de particular interés, siendo su principal utilidad el obtener grupos de observaciones similares. Formalmente, el problema de clustering puede ser escrito como sigue:

Entrada: Conjunto de entrenamiento $S_n = \{x^{(i)}, i = 1, 2, \ldots, n\}$, donde $x^{(i)} \in \mathbb{R}^d$, $k \in \mathbb{N}$
Salida: Conjunto de clusters $C_1, \ldots, C_k$.

La salida del algoritmo de clustering puede ser presentada en dos formas, a saber:

- Cada grupo(cluster) como un conjunto.
- Los centroides que representan a cada grupo.

Es preciso puntualizar el criterio de similitud que usaremos para determinar la composición de los clusters, para ello debemos ser capaces de comparar por pares vectores en el conjunto de entrenamiento para determinar si son en efecto similares (pertenecen al mismo cluster) o no (pertenecen a un cluster distinto). Dicha comparación puede ser en terminos de similitud tal como la *similitud coseno* o por disimilitud como la distancia euclídea, donde las expresiones respectivas son:

- $\cos\left(x^{(i)}, x^{(j)}\right) = \frac{x^i \cdot x^j}{\|x^i\|\|x^j\|} = \frac{\sum_{l=1}^{d} x_l^{(i)} x_l^{(j)}}{\sqrt{\sum_{l=1}^{d}\left(x_l^{(i)}\right)^2}\sqrt{\sum_{l=1}^{d}\left(x_l^{(j)}\right)^2}}$
- $dist\left(x^{(i)}, x^{(j)}\right) = \|x^{(i)} - x^{(j)}\|^2 = \sum_{l=1}^{d}\left(x_l^{(i)} - x_l^{(j)}\right)^2$

Una vez que hemos seleccionado la métrica de distancia, estamos en posibilidad de especificar la función objetivo para el agrupamiento, es decir, especificamos el costo de elegir algún conjunto particular de clusters (o sus centroides), así, el clustering óptimo se obtiene al minimizar dicho costo. A menudo, se expresa el costo en términos de la *distorsión asociada* a clusters individuales. Para definir tal distorsión, usaremos la suma de las distancias al cuadrado de cada vector en el cluster versus su correspondiente centroide, en consecuencia, para el cluster C con centroide z, la distorsión se definirá como $\sum_{i \in C} \|x^{(i)} - z\|^2$, por tanto, el costo de generar los clusters $C_1, C_2, \ldots, C_k$ será básicamente la suma de los costos de cada cluster individual, por tanto:

$$costo\left(C_1, C_2, \ldots, C_k, z^{(1)}, z^{(2)}, \ldots, z^{(k)}\right) = \sum_{j=1,\ldots,k} \sum_{i \in C_j} \|x^{(i)} - z^{(j)}\|^2$$

De esta manera, nuestra meta será encontrar la clusterización que minimice este costo.
La aproximación expuesta corresponde al llamado clustering de optimización que es el más utilizado en la práctica, fue elegido para ilustrar la idea básica de la metodología de modelación no supervisada. En temas posteriores, revisaremos a detalle las particularidades de cada método.

2. Visualización de hiperespacios en dimensión reducida

En la presente sección, revisaremos lo correspondiente a visualización de clusters en dimensión reducida. Como podemos ver, nuestro conjunto de entrenamiento $S_n = \{x^{(i)}, i = 1, 2, \ldots, n\}$, con $x^{(i)} \in \mathbb{R}^d$ no tiene una representación geométrica inmediata para $d > 3$, que dicho sea de paso, ocurre en la mayoría de las situaciones en la práctica. Es por ello que recurriremos

a diversos artificios matemáticos para representar de la manera más fidedigna posible la información de alta dimensionalidad en una dimensión más baja (usualmente $\mathbb{R}^2$ o $\mathbb{R}^3$).

2.1. Análisis de componentes principales

El primer método a revisar es el llamado análisis de componentes principales (PCA por sus siglas en lengua inglesa). El método tiene por objeto transformar un conjunto de variables en un nuevo conjunto denominado componentes principales. Los nuevos componentes tienen la característica de ser incorrelacionados (ortogonales) y se ordenan de acuerdo a la cantidad de información (varianza) que llevan incorporada. Las componentes principales se expresan como una combinación lineal de las variables originales. Procedamos a su obtención:

Sean $X_1, X_2, \ldots, X_p$ un conjunto de variables de una muestra de tamaño n interrelacionadas entre sí, se busca obtener otro conjunto $Z_1, Z_2, \ldots, Z_k$ con $k \leq p$ tales que sean una combinación lineal del conjunto inicial y que expliquen la mayor parte de su variabilidad.

Obtengamos la primera componente:

$$Z_{1i} = u_{11}X_{1i} + u_{12}X_{2i} + \ldots + u_{1p}X_{pi}$$

Al tomar las n observaciones muestrales, tenemos:

$$\begin{bmatrix} Z_{11} \\ Z_{12} \\ \vdots \\ Z_{1n} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{21} & \ldots & X_{p1} \\ X_{12} & X_{22} & \ldots & X_{p2} \\ \vdots & \vdots & \ddots & \vdots \\ X_{1n} & X_{2n} & \ldots & X_{pn} \end{bmatrix} \begin{bmatrix} u_{11} \\ u_{12} \\ \vdots \\ u_{1p} \end{bmatrix}$$

O en notación matricial:

$$\vec{Z_1} = X\vec{u}_1$$

Al suponer que las X_j están estandarizadas, podemos asumir que:

$$E[\vec{Z_1}] = E[X\vec{u}_1] = E[X]\vec{u}_1 = 0$$

Y la varianza sería:

$$\begin{aligned} V[\vec{Z_1}] &= \frac{1}{n}\sum_{i=1}^{n} Z_{1i}^2 \\ &= \frac{1}{n}\vec{Z_1}^T\vec{Z_1} \\ &= \frac{1}{n}\vec{u_1}^T X^T X \vec{u_1} \\ &= \vec{u_1}^T \left[\frac{1}{n}X^T X\right]\vec{u_1} \\ &= \vec{u_1}^T V \vec{u_1} \end{aligned}$$

Donde V es la matriz de covarianzas.
Para hallar $\vec{Z_1}$ necesitamos maximizar la varianza tal que la suma de los pesos u_{1j} al cuadrado sea igual a la unidad, en consecuencia, tenemos un problema de optimización restricta.

$$\begin{aligned} &max\ V[\vec{Z_1}] = \vec{u_1}^T V \vec{u_1} \\ &s.a.\ \sum_{j=1}^{p} u_{1i}^2 = \vec{u_1}^T\vec{u_1} = 1 \end{aligned}$$

Para resolverlo, recurrimos a los multiplicadores de Lagrange:

$$\begin{aligned} L &= \vec{u_1}^T V \vec{u_1} - \lambda\left(\vec{u_1}^T\vec{u_1} - 1\right) \\ \frac{\partial L}{\partial \vec{u_1}} &= 2V\vec{u_1} - 2\lambda\vec{u_1} = 0 \Rightarrow (V - \lambda I)\,\vec{u_1} = 0 \end{aligned}$$

La ecuación anterior solo tiene solución si $\|V - \lambda I\| = 0$ y en consecuencia, λ es un valor propio de la matriz V. Al premultiplicar por $\vec{u_1}^T$, tenemos:

$$\vec{u_1}^T (V - \lambda I)\,\vec{u_1} = 0 \Rightarrow \vec{u_1}^T V\vec{u_1} - \lambda\vec{u_1}^T I\vec{u_1} = \vec{u_1}^T V\vec{u_1} - \lambda = 0 \Rightarrow \vec{u_1}^T V\vec{u_1} = \lambda = V[\vec{Z_1}]$$

Sabemos que $\lambda_1, \lambda_2, \ldots, \lambda_n$ pueden ordenarse de forma ascendente tal que: $\lambda_1 > \lambda_2 > \ldots > \lambda_n$, de esta manera maximizaremos la varianza explicada tomando el mayor valor propio de V.

2.2. Escalamiento multidimensional

Otra técnica que es de utilidad para disminuir la dimensión de nuestra matriz entrada es conocida como escalamiento multidimensional(MDS por sus siglas en lengua inglesa). Este procedimiento consiste en generar un mapeo entre un espacio de alta dimensionalidad a otro de más baja dimensionalidad conservando la relación entre las distancias. En MDS clásico, las distancias se consideran euclídeas, considérese una matriz de $n \times p$, se genera una matriz de distancias y a partir de ella, se busca un mapeo en $\mathbb{R}^2$ o $\mathbb{R}^3$ tal que produzca con suficiente cercanía dicha matriz. Sea d_{ij} la distancias entre las observaciones $x^{(i)}$ y $x^{(j)}$ del espacio original, definimos δ_{ij} como la distancia correspondiente en dimensión reducida que producirá los vectores $x'^{(i)}$. Buscamos entonces un mecanismo que nos permita minimizar la discrepancia entre las distancias presentadas, naturalmente podemos proponer:

$$\sum_{i<j} \left(d_{ij} - \delta_{ij}\right)^2$$

Sin embargo, aunque matemáticamente correcto, presenta el incoveniente de ser sensible a la escala del espacio. Es por ello que incorporamos un factor normalizante a esta función de costo y adicionalmente convertimos a unidades lineales para tener:

$$\sqrt{\frac{\sum_{i<j} \left(d_{ij} - \delta_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}}$$

Que es conocida como *stress normalizado*. A continuación, mostramos los pasos necesarios para MDS clásico:

1. Generar la matriz de proximidades cuadradas $P^{(2)} = [p^2]$
2. Aplicar doble centrado $B = -\frac{1}{2} J P^{(2)} J$ usando la matriz $J = I - n^{-1}11'$, donde n es el número de objetos.
3. Extraer los m valores propios mayores y sus correspondientes vectores propios $\vec{e_1}, \vec{e_2}, \ldots, \vec{e_m}$.
4. Se genera una configuración m-espacial de los n objetos derivados de la matriz de coordenadas $X = E_m \Lambda_m^{\frac{1}{2}}$,donde E_m es la matriz de

los m vectores propios y Λ_m es la matriz diagonal de los m valores propios de B.

El algoritmo anterior garantiza la minimización del stress cuando las distancias involucradas son euclideas.

2.3. Incrustación de vecinos estocásticos distribuída t

Conocida como t-SNE por sus siglas en inglés, es una técnica no lineal de reducción dimensional, posee un enfoque similar a MDS, sin embargo, el trasfondo del mapeo dimensional es distinto. t-SNE minimiza la divergencia entre dos distribuciones: la distribución que mide las similitudes por pares en el conjunto de entrenamiento y la distribución de las similitudes en el espacio de baja dimensionalidad. Considérese un conjunto de alta dimensionalidad $S_n = \{x^{(i)}, i = 1, 2, \ldots, n\}$ junto con una función de distancia $d\left(x^{(i)}, x^{(j)}\right)$. El objetivo es aprender una incrustación s-dimensional en el que cada objeto será representado por un punto $E = \{y^{(i)}, i = 1, 2, \ldots, n\}$ donde $y^{(i)} \in \mathbb{R}^s$, $s \in \{2, 3\}$ donde t-SNE definirá la probabilidad conjunta p_{ij} que mide la similitud por pares entre los vectores $x^{(i)}$ y $x^{(j)}$ al simetrizar dos probabilidades condicionales como sigue:

$$p_{j|i} = \frac{\exp\left(-d\left(x^{(i)}, x^{(j)}\right)^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-d\left(x^{(i)}, x^{(k)}\right)^2 / 2\sigma_i^2\right)}, \quad p_{i|i} = 0$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

La dispersión σ_i se elige de tal suerte que la perplejidad (medida de que tan bien una distribución de probabilidad predice una muestra, $2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)}$) de la distribución condicional P_i es igual a una perplejidad definida u. En la incrustación s-dimensional E, las similitudes entre los vectores $y^{(i)}$ y $y^{(j)}$ se miden mediante un kernel de colas pesadas, en particular, la similitud incrustada q_{ij} para los vectores $y^{(i)}$ y $y^{(j)}$ se calcula mediante un kernel normalizado t-Student con un grado de libertad:

$$q_{ij} = \frac{\left(1 + \|y^{(i)} - y^{(j)}\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y^{(k)} - y^{(l)}\|^2\right)^{-1}} \quad q_{ii} = 0$$

Las colas pesadas del kernel normalizado t-Student permiten que vectores de entrada con disimilitud $x^{(i)}$ y $x^{(j)}$ sean modelados por sus contrapartes en baja dimensión $y^{(i)}$ y $y^{(j)}$ que están lejos entre sí. La ubicación de los puntos incrustados $y^{(i)}$ se determinan mediante la minimización de la divergencia de Kullback-Leibler entre las distribuciones conjuntas P y Q:

$$C\left(E\right) = KL\left(P||Q\right) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Dada la asimetría de la divergencia de Kullback-Leibler, la función objetivo se enfoca en modelar valores altos de p_{ij}(vectores similares) mediante valores altos de q_{ij} (puntos cercanos en el espacio incrustado). La función objetivo es no convexa en el incrustamiento E y se optimiza típicamente mediante gradiente descendiente.

$$\frac{\partial C}{\partial y^{(i)}} = 4 \sum_{j \neq i} \left(p_{ij} - q_{ij}\right) q_{ij} \sum_{k \neq l} \left(1 + \|y^{(k)} - y^{(l)}\|^2\right)^{-1}$$

3. Clustering Jerárquico

Uno de los métodos más simples es el llamado clustering jerárquico, el mismo se basa en la comparación de similitud por pares de las observaciones en el conjunto de entrenamiento. El clustering jerárquico se divide en dos tipos principalmente:

- Aglomerativo
- Divisivo

El clustering aglomerativo se basa en la premisa de que cada observación del conjunto de entrenamiento pertenece a su propio cluster y mediante el cómputo iterativo de las similitudes entre observaciones se van produciendo grupos. Por el contrario, el clustering jerárquico divisivo inicia con un solo cluster que contiene todo el conjunto de entrenamiento y genera particiones en cada iteración sucesiva.

El método funciona básicamente como sigue:

Sea $S_n = \{x^{(i)}, i = 1, 2, \ldots, n\}$ el conjunto de entrenamiento a clusterizar, entonces:

1. Asignar cada vector $x^{(i)}$ a un cluster separado, formando asi n clusters.
2. Unir los dos vectores más cercanos en un cluster, de esta forma se tienen $n - 1$ clusters.
3. Tomar los dos clusters más cercanos y fusionarlos, se obtendrán entonces $n - 2$ clusters
4. Repetir el paso 3 hasta que se forme un único cluster.

Revisemos ahora los distinos métodos de enlace para medir la similitud entre clusters.

3.1. Método de promedio

La similitud entre clusters se medirá como la distancia promedio entre cada vector dentro del cluster C_K versus cada vector dentro del cluster

C_L, esto es:

$$S_{KL} = \frac{1}{n_K n_L} \sum_{i \in C_K} \sum_{j \in C_L} d\left(x^{(i)}, x^{(j)}\right)$$

3.2. Método de centroide

En primera instancia se realiza el cómputo del centroide de cada cluster y posteriormente se calcula la distancia entre los centroides.

$$S_{KL} = \|\bar{X}_K - \bar{X}_L\|$$

3.3. Método de Ward

El método de Ward se basa en un criterio de optimalidad donde se busca minimizar la varianza dentro del cluster. En cada iteración se buscan todas las particiones posibles al fusionar dos clusters en una generación previa tales que minimicen la varianza de acuerdo a la selección, esto es:

$$S_{KL} = \frac{\|\bar{X}_K - \bar{X}_L\|^2}{\frac{1}{n_K} + \frac{1}{n_L}}$$

4. Clustering de optimización

Anteriormente revisamos la metodología de modelación no supervisada y precisamente la aproximación realizada fue en mayor medida desde el punto de vista del clustering de optimización, donde minimizamos el costo de las agupaciones. Dado un conjunto de puntos:

$$C_j = \{i \in \{1, \ldots, n\} | z^{(j)} \; representa \; a \; x^{(j)}\}$$

entonces:

$$cost\left(z^{(1)}, \ldots, z^{(k)}\right) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} \|x^{(i)} - z^{(j)}\|^2$$

La expresión anterior simplemente asigna cada punto a su más cercano representante. Geométricamente, la partición provocada por los centroides puede ser visualizada como una partición de Voronoi en $\mathbb{R}^d$, donde $\mathbb{R}^d$ se divide en k células convexas. La célula será entonces la región en el espacio donde el correspondiente centroide z es el mejor representante, obsérvese la figura siguiente:

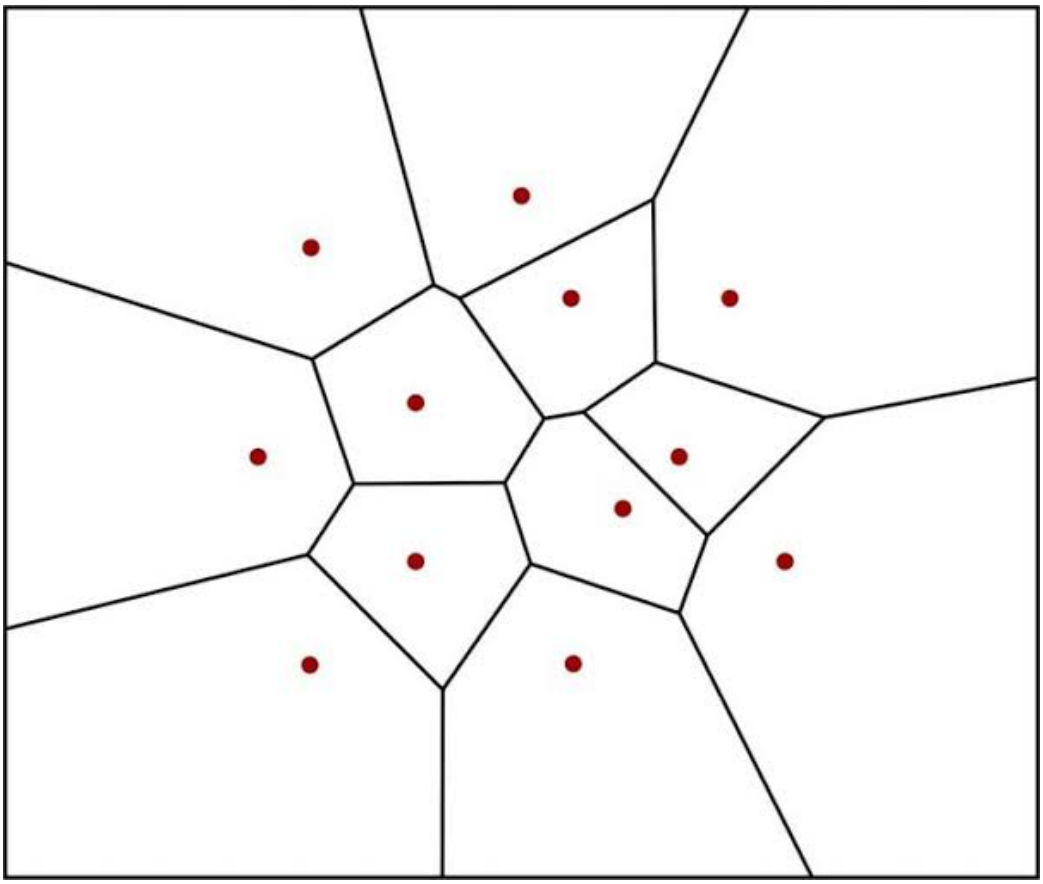

Figura 4.1: Diagrama de Voronoi

Revisaremos ahora el algoritmo más común para esta tarea.

4.1. K-medias

Dado un criterio de optimización, necesitamos un algoritmo que nos permita minimizar el costo de las agrupaciones descrito. Si enumeraramos cada una de las particiones posibles, el cómputo requerido es inaccesible, de esta forma, utilizaremos un algoritmo de aproximación para completar esta tarea. Dicho algoritmo es conocido como *k-medias*, el mismo de manera alternativa encuentra los mejores clusters para los centroides y los mejores centroides para los clusters. Las iteraciones se realizan según lo siguiente:

1. Inicializar los centroides $z^{(1)}, \ldots, z^{(k)}$
2. Repetir hasta que no haya un cambio significativo en la función de costo:
   a) Para cada $j = 1, \ldots, k$: $C_j = \{i | x^{(i)} \; es \; el \; más \; cercano \; a \; z^{(j)}\}$
   b) Para cada $j = 1, \ldots, k$: $z^{(j)} = \frac{1}{|C_j|} \sum_{i \in C_j} x^{(i)}$

El algoritmo garantiza convergencia, sin embargo, no necesariamente a un óptimo global, por tanto, la elección de los centroides iniciales o incluso un orden distinto de los vectores presentado en el entrenamiento puede alterar significativamente los resultados.

5. Clustering de densidad

5.1. Modelos Gaussianos Mixtos

Considérese un conjunto de puntos que se dividen en 2 o más clusters, es posible describir cada cluster a través de una distribución gaussiana para cada uno. Dicho modelo incluirá las diferentes localizaciones y dispersiones de cada gaussiana así como cuántos puntos pertenecen a cada cluster (proporciones de mezcla). Un modelo con esta aproximación es llamado *modelo mixto*. Los modelos mixtos asumen un proceso generativo en dos etapas: en primera instancia, se selecciona a que cluster pertenece el punto a generar y posteriormente, se genera el punto a partir del modelo correspondiente. Un hecho interesante es que es posible ver los modelos mixtos

como extensiones probabilísticas de *k-medias*.
Supóngase la existencia de exactamente k clusters, definimos entonces nuestro modelo mediante:

$$N\left(x; \mu^{(j)}, \sigma_j^2 I\right),\ j = 1, \ldots, k$$

Dado lo anterior, debemos estimar de alguna forma los parámetros $\mu^{(1)}, \ldots, \mu^{(k)}, \sigma_1^2, \ldots, \sigma_1^k$ sin conocimiento a priori de la pertenencia de los puntos a algún cluster particular. Dado que los clusters no serán necesariamente del mismo tamaño, debemos incluir parámetros adicionales $p_1, \ldots, p_k$ (las proporciones de la mezcla) los cuales especificarán la proporción de puntos esperada en cada cluster.
Para generar puntos a partir de la mezcla gaussiana tomaremos una muestra sobre el índice j, la muestra será tomada con base en la distribución multinomial dada por $p_1, \ldots, p_k$, donde $\sum_{j=1}^{k} p_k = 1$. Una vez que se conoce el cluster, obtenemos una muestra x de la correspondiente gaussiana, es decir:

$$j \sim Multinomial\left(p_1, \ldots, p_k\right)$$
$$x \sim P\left(x|\mu^{(j)}, \sigma_j^2\right)$$

Dado que no contamos con datos etiquetados en modelación no supervisada, mediante las mezclas gaussianas evaluaremos la probabilidad de que un vector x pueda generarse a partir de una muestra de nuestro modelo y posteriormente ajustar los parámetros para incrementar dicha probabilidad. Cada x pudo haber sido generado a partir de cualquiera de los clusters pero con diferente probabilidad. De esta forma, para evaluar $P\left(x|\theta\right)$ donde θ se refiere a todos los parámetros del modelo mixto.

$$\theta = \{\mu^{(1)}, \ldots, \mu^{(k)}, \sigma_1^2, \ldots, \sigma_1^k, p_1, \ldots, p_k\}$$

Sumamos entonces sobre todas aquellas alternativas que pudiesen haber generado al vector x, esto es:

$$P\left(x|\theta\right) = \sum_{j=1}^{k} p_j N\left(x; \mu^{(j)}, \sigma_j^2\right)$$

Este será el modelo gaussiano mixto que debemos estimar a partir de los datos $S_n = \{x^{(t)}, t = 1, \ldots, n\}$, cabe señalar que no es una tarea trivial determinar la forma y posición de los clusters. Para ilustrar esto, revisemos en primera instancia simplificando el problema al estimar mediante datos etiquetados para posteriormente generalizar a la obtención a partir únicamente de S_n. Si los datos estuviesen etiquetados cada punto podría ser asignado a un solo cluster y podríamos estimar nuestra mezcla gaussiana como dijimos. Adicionalmente, podríamos evaluar el tamaño de los clusters basados únicamente en el número de puntos. Sea $\delta\left(j|t\right)$ una función indicadora definida por:

$$\delta(t) = \begin{cases} 1, \ si\ x^{(t)}\ es\ asignado\ a\ j \\ 0, \ en\ otro\ caso \end{cases}$$

Al introducir esta notación, la función objetivo de máxima verosimilitud es:

$$\sum_{t=1}^{n}\left[\sum_{j=1}^{k}\delta\left(j|t\right)\log\left(p_j N\left(x^{(t)}; \mu^{(j)}, \sigma_j^2 I\right)\right)\right] = \sum_{j=1}^{k}\left[\sum_{t=1}^{n}\delta\left(j|t\right)\log\left(p_j N\left(x^{(t)}; \mu^{(j)}, \sigma_j^2 I\right)\right)\right]$$

En el miembro izquierdo de la igualdad previa, la suma interna sobre los clusters simplemente selecciona la gaussiana que debería usarse para generar el punto correspondiente, mientras que el miembro derecho las sumas se intercambian con la intención de ilustrar el hecho de que cada gaussiana individual puede ser resuelta de forma separada. Es de destacar también la inclusión de p_j en la generación de cada punto, es decir, la probabilidad a priori de seleccionar el cluster j para el punto $x^{(t)}$. Así, la solución de máxima verosimilitud basada en datos etiquetados está dada por:

$$\hat{n_j} = \sum_{t=1}^{n} \delta\left(j|t\right) \ \left(número\ de\ puntos\ asignados\ al\ cluster\ j\right)$$

$$\hat{p_j} = \frac{\hat{n_j}}{n} \ \left(fracción\ de\ puntos\ en\ el\ cluster\ j\right)$$

$$\hat{\mu}^{(j)} = \frac{1}{\hat{n_j}} \sum_{t=1}^{n} \delta\left(j|t\right) x^{(t)} \ \left(media\ de\ puntos\ en\ el\ cluster\ j\right)$$

$$\sigma_j^2 = \frac{1}{d\hat{n_j}} \sum_{t=1}^{n} \delta\left(j|t\right) \|x^{(t)} - \hat{\mu}^{(j)}\|^2 \ \left(dispersión\ media\ cuadrada\ en\ el\ cluster\ j\right)$$

En el caso complementario, es decir, donde no existen etiquetas, recurriremos al algoritmo Expectation-Maximization (EM) para hacer la estimación de los parámetros. Nuestro objetivo es maximizar la verosimilitud de que nuestro modelo mixto genere los datos, matemáticamente equivalente a maximizar:

$$l\left(S_n;\theta\right) = \sum_{t=1}^{n} \log P\left(x^{(t)}|\theta\right) = \sum_{t=1}^{n} \log\left(\sum_{j=1}^{k} p_j N\left(x^{(t)};\mu^{(j)},\sigma_j^2 I\right)\right)$$

con respecto a los parámetros θ. La expresión anterior conlleva considerar las distintas combinaciones de k gaussianas que expliquen mejor los datos, por tanto, será computacionalmente costoso y difícil. La solución a este problema es un proceso iterativo (el mencionado EM). El algoritmo se basa en el caso etiquetado, tomando como base el mismo modelo para hacer las asignaciones , después, se estima cada modelo individual del cluster basado en los puntos asignados a el, por tanto, en cada iteración las asignaciones pueden cambiar, si notamos, el algoritmo es muy parecido a

k-medias, sin embargo, en este caso no es conveniente asignar a un cluster únicamente ya que debemos considerar la posibilidad de que un punto pueda ser generado por distintos modelos, de esta forma, realizamos asignaciones *suaves* a cada cluster basadas en las probabilidades relativas de generación de cada cluster. Comenzamos por inicializar los parámetros de la mezcla, por ejemplo, una opción es inicializar las medias $\mu^{(1)}, \ldots, \mu^{(k)}$ como en *k-medias* y establecer las varianzas σ_j^2 equivalentes a la varianza total de los datos:

$$\hat{\sigma}^2 = \frac{1}{dn} \sum_{t=1}^{n} \|x^{(t)} - \hat{\mu}\|^2$$

Donde $\hat{\mu}$ es la media de todos los puntos. Para asegurar que las gaussianas puedan considerar todos los puntos y para evitar la atracción excesiva de un cluster particular, consideramos uniformes los coeficientes de la mezcla gaussiana, es decir, $p_j = 1/k, \;\; j = 1, \ldots, k$. Hecho esto, procedemos a definir los pasos del algoritmo EM:

- **Paso-E**: Asignar puntos suavemente a los clusters de acuerdo a la probabilidad posterior:

$$p\left(j|t\right) = \frac{p_j N\left(x; \mu^{(j)}, \sigma_j^2 I\right)}{P\left(x|\theta\right)} = \frac{p_j N\left(x; \mu^{(j)}, \sigma_j^2 I\right)}{\sum_{l=1}^{k} p_l N\left(x; \mu^{(l)}, \sigma_l^2 I\right)}$$

  En la expresión anterior, $\sum_{j=1}^{k} p\left(j|t\right) = 1$, lo cual es análogo en forma probabilística de $\delta\left(j|t\right)$ en el caso etiquetado. Cada punto $x^{(t)}$ será asignado al cluster j con peso $p\left(j|t\right)$, donde a mayor peso, más fuerte la aseveración de que el punto fue generado por el cluster j.

- **Paso-M** Una vez calculado $p\left(j|t\right)$, suponemos que las asignaciones al cluster fueron dadas y las usamos para estimar las gaussianas de forma separada tal y como en el caso etiquetado:

$$\hat{n_j} = \sum_{t=1}^{n} p\left(j|t\right) \;\; (n\acute{u}mero \; efectivo \; de \; puntos \; asignados \; al \; cluster \; j)$$

$$\hat{p_j} = \frac{\hat{n_j}}{n} \;\; (fracci\acute{o}n \; de \; puntos \; en \; el \; cluster \; j)$$

$$\hat{\mu}^{(j)} = \frac{1}{\hat{n_j}} \sum_{t=1}^{n} p\left(j|t\right) x^{(t)} \ \ (media\ ponderada\ de\ puntos\ en\ el\ cluster\ j)$$

$$\sigma_j^2 = \frac{1}{d\hat{n_j}} \sum_{t=1}^{n} p\left(j|t\right) \|x^{(t)} - \hat{\mu}^{(j)}\|^2 \ \ (dispersión\ media\ ponderada\ cuadrada\ en\ el\ cluster\ j)$$

Al concluir este paso, usamos estos parámetros nuevamente en el **Paso-E** e iteramos.

Este algoritmo garantiza el incremento monotónico del logaritmo de la verosimilitud de los datos bajo el modelo mixto, puede estacionarse en un óptimo local, sin embargo, es menos frágil que *k-means* debido a las asignaciones suaves.

6. Perfilamiento de grupos

Una vez que se ha extraído el patrón mediante la técnica matemática seleccionada, es imperativo obtener una descripción cualitativa de cada uno de los clusters obtenidos. Lo anterior se debe a que será dicha descripción la que permita accionar en la práctica lo necesario para implementar los beneficios derivados de la identificación de segmentos. Para lograr este fin, deben encontrarse en primera instancia aquellas variables continuas que son estadísticamente significativas para explicar el cluster (hayan sido utilizadas en la obtención del patrón o no, esto dependerá de cada caso particular). En segundo término, las variables discretas involucradas serán contrastadas en frecuencia relativa versus la proporción total de los segmentos. En el caso de las variables continuas, se recomienda el uso de un contraste ANOVA de tipo no paramétrico (prueba de Kruskal-Wallis) y posterior prueba *post-hoc* de Tukey, mientras que en el caso discreto recurriremos a la prueba χ^2.

7. Reportes de estabilidad

Cuando el perfil de los segmentos está completo y en caso de que intervenga en el caso de estudio la dimensión tiempo, será deseable analizar

si los segmentos tienen proporciones sigificativamente distintas a través del tiempo. Para ello, se contrasta la proporción de segmentos en cada momento del tiempo versus la proporción general de los segmentos en el conjunto de datos. En la siguiente figura se muestra este hecho:

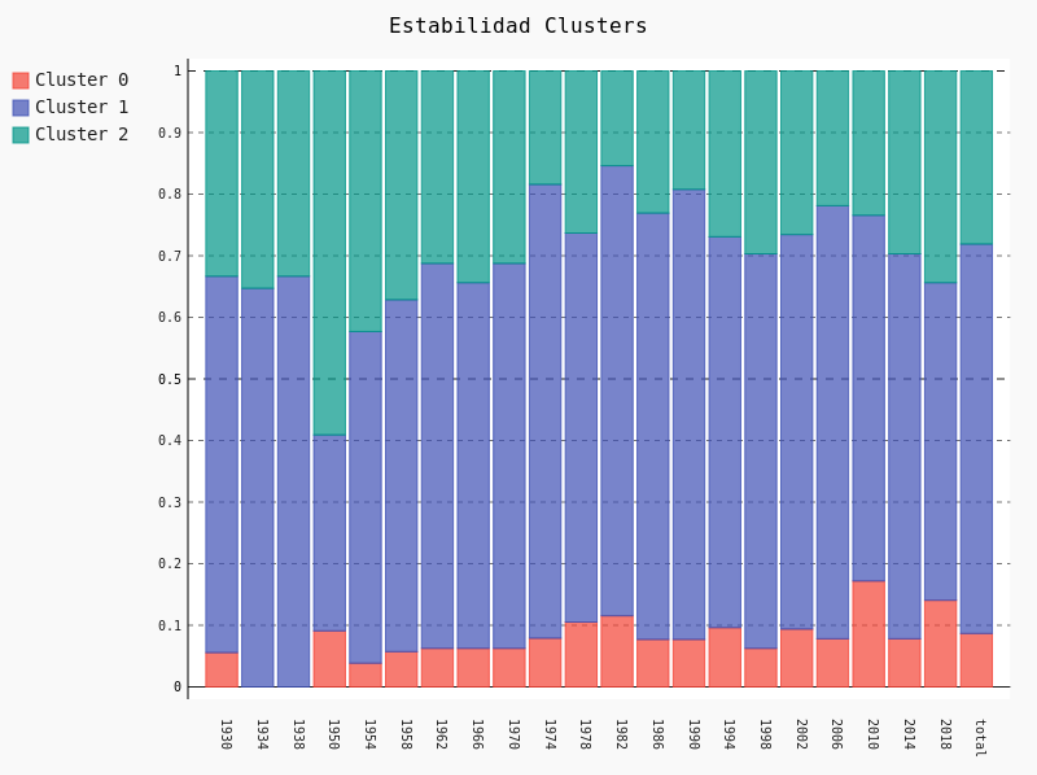

Figura 7.1: Clusters a través del tiempo

A simple vista puede observarse la relativa estabilidad de los segmentos. La validación estadística podrá comprobarse mediante la prueba χ^2 o mediante el cálculo del PSI(Population Stability Index) el cual nos indica que una población diverge de las proporciones esperadas mediante un criterio de entropía, esto es:

$$PSI = \sum \left((Actual - Esperado) * \log \left(\frac{Actual}{Esperado} \right) \right)$$

Donde valores de PSI menores a 0.1 indican un cambio insignificante en la población, valores entre 0.1 y 0.25 corresponden a un cambio menor en la población y valores superiores a 0.25 representan un cambio mayor en la población. En el siguiente gráfico se muestra el cálculo del PSI para la población de la figura 7.1:

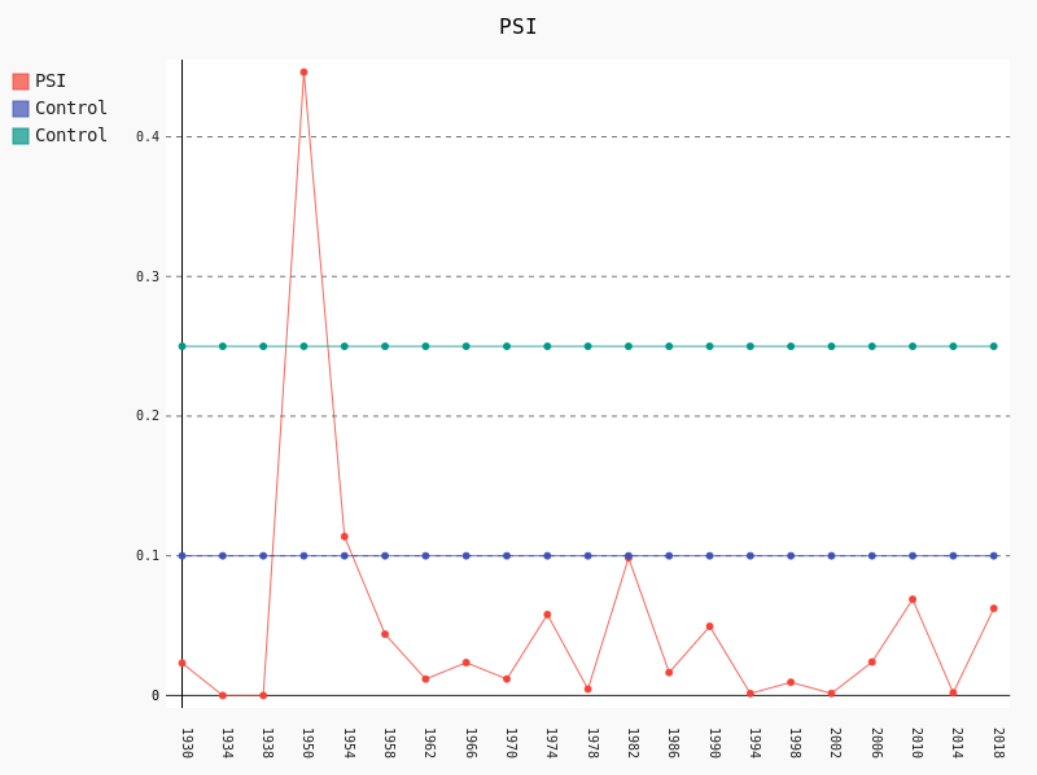

Figura 7.2: Clusters a través del tiempo