UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

Algoritmo EM, su Implementación y Aplicaciones

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Actuario

PRESENTA:

María Cristina Guzmán Solís

TUTOR

Dr. Fernando Baltazar-Larios

Ciudad Universitaria, Cd. Mx. 2019

Agradecimientos

A mis padres, Mayté y Juan Carlos, por su amor y apoyo incondicional y motivación para ser mejor persona cada día, por el ejemplo que me han dado y por ser la razón principal de lograr lo que he logrado hasta ahora.

Al proyecto PAPIME PE102618, que por su apoyo me facilitó realizar esta tesis.

A mi asesor Fernando Baltazar Larios, por su paciencia y que sin su valiosa guía y ayuda, no me habría sido posible realizar esta tesis. También le agradezco en conjunto con el profesor Sergio Iván López Ortega, por haberme invitado a participar en dicho proyecto, por su gran apoyo a lo largo de estos meses y por inspirarme y motivarme a superarme tanto personal como académicamente.

A mis sinodales Ramsés Mena, Verónica Miró y María Fernanda Gil Leyva, por su guía, apoyo, correcciones y comentarios sobre mi tesis.

A mis amigas Alba y Carla, que han vivido conmigo de cerca este proceso y me han apoyado en todo momento.

A la Residencia Universitaria Latinoamericana, quien me recibió y me acogió por tres años de mi carrera, me facilitó enfocarme en mis estudios, por las experiencias que viví ahí, porque me permitió conocer a gente extraordinaria y en donde conocí a mis amigas más cercanas.

Y a la Facultad de Ciencias y la Universidad Nacional Autónoma de México, por darme la excelente educación que recibí y que me ha abierto muchas puertas.

Índice general

Introducción

En este trabajo hablaremos sobre el algoritmo Esperanza-Maximización, mejor conocido como algoritmo EM. Dada la gran variedad de aplicaciones en los cuales el algoritmo EM se puede implementar, así como por el carácter genérico de su procedimiento, muchos prefieren referirse a él como un meta-algoritmo; sin embargo, continuaremos refiriéndonos a él como algoritmo EM. Dicho algoritmo se usa para estimación por máxima verosimilitud en problemas con datos incompletos (*incomplete-data* problems). A pesar de que en el presente trabajo usaremos el algoritmo EM para estimación por máxima verosimilitud, cabe mencionar que también dicho algoritmo puede ser adaptado para obtener el estimador MAP (Maximum a posteriori) en el contexto bayesiano.

A diferencia de sus predecesores como Newton-Rhapson y el método de Scoring para la obtención de estimadores máximo verosímiles, el algoritmo EM tiene la ventaja de ser mucho más sencillo y de lidiar con problemas de datos incompletos, en los cuales los métodos antes mencionados pueden resultar más complicados de implementar. Para leer más acerca de los métodos numéricos de Newton para obtener estimadores máximo verosímiles refiérase a [15].

El propósito de este trabajo es presentar el algoritmo EM como herramienta para resolver el problema de estimación máximo verosímil en problemas de información incompleta. Se muestran algunas implementaciones y aplicaciones del algoritmo EM tanto en su versión determinista como en su versión estocástica. Empezamos en el Capítulo 1 con una breve introducción a la estimación por máxima verosimilitud y presentamos propiedades de los estimadores máximo verosímiles, y finalizamos la sección con una introducción a las familias exponenciales.

En el Capítulo 2 presentamos la formulación del algoritmo EM, en donde introducimos los conceptos que se utlizarán a lo largo del trabajo. Procedemos con una breve descripción de la historia del algoritmo EM, antes de la publicación del artículo de Dempster, Laird y Rubin [12] y después se presentan algunos ejemplos de implementaciones del algoritmo EM. Finalmente, mostramos dos aplicaciones de clasificación, una con datos reales muy conocidos y otra con datos simulados, para clasificación de estados moduladores en un modelo de riesgo modulado [2].

En el Capítulo 3 hablaremos sobre la teoría básica del algoritmo EM, presentamos resultados de convergencia de la secuencia generada por el algoritmo, tanto de la función de log-verosimilitud como de los estimadores máximo verosímiles generados a cada iteración. También presentamos el Teorema de Convergencia Global y continuamos con el capítulo pasando al tema del cálculo de errores estándar y tasa de convergencia. Concluimos el capítulo presentado métodos de aceleración de convergencia con un ejemplo.

Por último, en el Capítulo 4 hablamos sobre las versiones estocásticas del algoritmo EM, que son el algoritmo EM Monte Carlo y el algoritmo EM Estocástico. Iniciamos con una breve introdución a los métodos de Monte Carlo para integración y procedemos con la formulación de los algoritmos EM Monte Carlo y EM Estocástico. Finalmente, mostramos los ejemplos vistos

en el Capítulo 2, pero ahora implementados con las versiones estocásticas. Exponemos algunos resultados importantes de la teoría de dichos algoritmos y concluimos con una aplicación vista en el Capítulo 2, sólo que esta vez hacemos uso del algoritmo EM Estocástico para estimación del proceso de riesgo modulado a tiempo continuo discretamente observado.

Capítulo 1

Estimación por Máxima Verosimilitud

Dado que el objetivo del algoritmo EM es obtener estimadores máximo verosímiles, es importante dar una introducción y mencionar las propiedades importantes de los estimadores obtenidos por medio de la estimación por máxima verosimilitud.

Como se explicará más adelante, el procedimiento que sigue el algoritmo EM se facilita mucho cuando tratamos con familias exponenciales, por lo que también daremos una breve introducción a las familias exponenciales y cómo estimar por máxima verosimilitud los parámetros de distribuciones provenientes de la familia exponencial.

1.1. Máxima Verosimilitud

La verosimilitud es una manera de resumir la evidencia que los datos que tenemos nos dan acerca de los parámetros desconocidos. La verosimilitud es una función del parámetro desconocido $\boldsymbol{\theta}$ en el espacio parametral, $\boldsymbol{\Theta}$, la cual mide, qué tan "plausible" es el parámetro desconocido, dados los datos que se tienen. Es lógico querer entonces obtener aquel valor de $\boldsymbol{\theta}$ en $\boldsymbol{\Theta}$ tal que maximice dicha función. Dicho valor será nuestro estimador máximo verosímil. La función de verosimilitud la definimos del siguiente modo.

Definición 1. Sea $x_1, x_2, ..., x_n$ una muestra aleatoria con distribución $p(x_i|\theta), \quad \theta \in \boldsymbol{\Theta}$. La función de verosimilitud de la muestra $x_1, x_2, ..., x_n$, está dada por:

$$L(\theta|\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|\theta).$$

Nos es conveniente trabajar con el logartimo de dicha función, el cual lo denotaremos como $\ell(\theta|\mathbf{x})$. Es decir, la función $\ell(\theta|\mathbf{x})$ queda expresada como

$$\ell(\boldsymbol{\theta}|\boldsymbol{x}) = \sum_{i=1}^{n} \log(p(x_i|\theta)).$$

El vector de estimadores máximo verosímiles que denotaremos por $\hat{\boldsymbol{\theta}}_n$ será aquel que maximice a $L(\boldsymbol{\theta}|\mathbf{x})$ en $\boldsymbol{\Theta}$. Sin embargo, se preferirá maximizar en vez a $\ell(\boldsymbol{\theta}|\boldsymbol{x})$, pues dada su propiedad de ser monótona creciente resulta equivalente a maximizar a la propia verosimilitud. Optaremos por usar la log-verosimilitud dado que computacionalmente es más eficiente y nos evita problemas numéricos.

1.2. Propiedades de los estimadores máximo verosímiles

Los estimadores máximo verosímiles $\hat{\theta}_n$ cuentan con propiedades deseables como lo son la consistencia, eficiencia y normalidad asintótica e invarianza, las cuales explicaremos a continuación. Antes de enunciar dichas propiedades, enunciaremos dos resultados que usaremos a lo largo de esta sección.

Ley de los Grandes Números

Teorema 1.2.1. *(Ley Débil de los Grandes Números) Sea $x_1, ..., x_n$ una muestra aleatoria proveniente de una variable aleatoria X tal que $\mathbb{E}[X] = \mu < \infty$. Entonces para todo $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}\left(\mid \bar{X} - \mu \mid > \epsilon\right) = 0, \tag{1.1}$$

con $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Por otro lado, tenemos la Ley Fuerte de los Grandes Números, la cual dice que $\bar{X}$ converge casi seguramente a μ, es decir lo siguiente.

Teorema 1.2.2. *(Ley Fuerte de los Grandes Números) Sea $x_1, ..., x_n$ una muestra aleatoria proveniente de una variable aleatoria X tal que $\mathbb{E}[X] = \mu < \infty$. Entonces*

$$\mathbb{P}\left(\lim_{n\to\infty} \mid \bar{X}_n - \mu \mid < \epsilon\right) = 1. \tag{1.2}$$

Otro resultado asintótico importante es el del Teorema de Límite Central, el cual enunciamos a continuación

Teorema 1.2.3. *(Teorema de Límite Central) Sean $x_1, ..., x_n$ una muestra aleatoria proveniente de una variable aleatoria X, tales que $\mathbb{E}[X] < \infty$ y $Var(X) = \sigma^2 < \infty$. Entonces $\sqrt{n}\left(\bar{X}_n - \mathbb{E}[X]\right)$ converge en distribución a la distribución normal con media cero y varianza σ^2.*

Es decir, sea $G_n(x)$ la función de distribución de $\dfrac{\sqrt{n}(\bar{X} - \mathbb{E}[X])}{\sigma}$. Entonces, para toda x en la cual G_n es continua se cumple que

$$\lim_{n\to\infty} G_n(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{\frac{-y^2}{2}}\, dy. \tag{1.3}$$

1.2.1. Consistencia

Una de las propiedades de los estimadores máximo verosímiles es que son consistentes, lo cual nos dice básicamente que dichos estimadores difieren del valor "verdadero" desconocido (al cual denotaremos como θ_0), con probabilidad cero, asintóticamente. Formalmente, lo definimos del siguiente modo.

Definición 2. Decimos $\hat{\theta}_n$ es consistente si

$$\lim_{n\to\infty} \mathbb{P}\left(\mid \hat{\theta}_n - \theta_0 \mid > \epsilon\right) = 0 \tag{1.4}$$

donde θ_0 es el valor verdadero del parámetro de la distribución de la muestra.

Observación 1. Una condición suficiente para (1.4) es

$$\mathbb{E}\left[\left(\hat{\theta}_n - \theta_0\right)^2\right] \to 0,$$

cuando $n \to \infty$.

Se demuestra haciendo uso de la desigualdad de Chebyshev.

Ejemplo 1.2.1. Sea $x_1, ..., x_n$ una muestra aleatoria provenientes de una variable aleatoria X tal que $X \sim N(\mu, \sigma^2)$. Veamos que el estimador máximo verosímil para μ es consistente.

Sabemos que dicho estimador es $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$. Su varianza está dada por

$$\text{Var}\,(\hat{\mu}) = \mathbb{E}\left[(\hat{\mu} - \mu)^2\right] = \frac{\sigma^2}{n},$$

la cual converge a cero conforme $n \to \infty$. Entonces, $\hat{\mu}$ converge en probabilidad a μ y por lo tanto, $\hat{\mu}$ es consistente.

1.2.2. Normalidad Asintótica

Otra propiedad importante con la que cuentan los estimadores máximo verosímiles es que tienen distribución normal asintóticamente, es decir

$$\sqrt{n}\left(\hat{\theta}_n - \theta_0\right) \to N(0, I(\theta_0)^{-1}), \tag{1.5}$$

donde $I(\theta)$ es la información de Fisher, que definimos a continuación.

Definición 3. Sea X una variable aleatoria con función de densidad (o función de probabilidad, en el caso de variables aleatorias discretas) $f(x \,|\, \theta)$. La Información de Fisher de X está dada por

$$I(\theta) = \mathbb{E}\left[\frac{\partial}{\partial\theta}\log f(x \,|\, \theta)\right]^2 \tag{1.6}$$

Otro modo de calcular la información de Fisher es por medio del siguiente lema.

Lema 1.2.1.

$$\mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\log f(x \,|\, \theta)\right] = -I(\theta)$$

Demostración. Por un lado, tenemos que

$$\frac{\partial}{\partial\theta}\log f(x \,|\, \theta) = \frac{\frac{\partial}{\partial\theta} f(x \,|\, \theta)}{f(x \,|\, \theta)},$$

y también que

$$\frac{\partial^2}{\partial\theta^2}\log f(x \,|\, \theta) = \frac{\frac{\partial^2}{\partial\theta^2} f(x \,|\, \theta)}{f(x \,|\, \theta)} - \frac{(\frac{\partial}{\partial\theta} f(x \,|\, \theta))^2}{f^2(x \,|\, \theta)}.$$

Luego, suponiendo válido el intercambio entre derivada e integral, obtenemos

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\log f(x\,|\,\theta)\right] &= \int\left(\frac{\partial^2}{\partial\theta^2}\log f(x\,|\,\theta)\right) f(x\,|\,\theta)dx \\
&= \int\left(\frac{\frac{\partial^2}{\partial\theta^2}f(x\,|\,\theta)}{f(x\,|\,\theta)} - \frac{(\frac{\partial}{\partial\theta}f(x\,|\,\theta))^2}{f^2(x\,|\,\theta)}\right) f(x\,|\,\theta)dx \\
&= \int\frac{\partial^2}{\partial\theta^2}f(x\,|\,\theta) - \mathbb{E}\left[\frac{\partial}{\partial\theta}\log f(x\,|\,\theta)\right]^2 \\
&= 0 - I(\theta) \\
&= -I(\theta)
\end{aligned}$$

□

Ahora sí podemos enunciar el siguiente teorema:

Teorema 1.2.4. *$\hat{\theta}_n$ cumple que*

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \to N(0, I(\theta_0)^{-1})$$

Demostración. Denotemos por $l_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log f(x_i\,|\,\theta)$ a la log-verosimilitud de la muestra, entonces se tiene que

$$l_n^{'}(\hat{\theta}_n) = 0,$$

con $l_n^{'}(\hat{\theta}_n) = \left[\frac{\partial}{\partial\theta}l_n(\theta)\right]_{\theta=\hat{\theta}_n}$.

Haciendo uso del teorema del valor medio, dados θ_0 y θ_1 un valor tal que $\theta_1 \in [\hat{\theta}_n, \theta_0]$obtenemos la siguiente expresión

$$0 = l_n^{'}(\hat{\theta}_n) = l_n^{'}(\theta_0) + l_n^{''}(\theta_1)(\hat{\theta}_n - \theta_0). \tag{1.7}$$

Por tanto, obtenemos la expresión

$$\sqrt{n}(\hat{\theta}_n - \theta_0) = -\frac{\sqrt{n}l_n^{'}(\theta_0)}{l_n^{''}(\theta_1)}. \tag{1.8}$$

Por otro lado, definamos la función por $l(\theta)$ del siguiente modo

$$l(\theta) = \int(\log f(x\,|\,\theta))f(x\,|\,\theta_0)dx. \tag{1.9}$$

Notemos que por Ley de los Grandes Números, ocurre que

$$l_n(\theta) \to l(\theta). \tag{1.10}$$

Como θ_0 maximiza a $l(\theta)$ y suponiendo válido el intercambio de la integral con la derivada, tenemos que $l^{'}(\theta_0) = \mathbb{E}_{\theta_0}\left(\frac{\partial}{\partial\theta}\left[\log f(x\,|\,\theta)\right]_{\theta=\theta_0}\right) = 0$. Por lo tanto, (1.8) se convierte en

$$\sqrt{n}l_n'(\theta_0) = \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}\left[\log f(x_i\,|\,\theta)\right]_{\theta=\theta_0} - 0\right)$$
$$= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}\left[\log f(x_i\,|\,\theta)\right]_{\theta=\theta_0} - \mathbb{E}_{\theta_0}\left(\frac{\partial}{\partial\theta}\left[\log f(x_i\,|\,\theta)\right]_{\theta=\theta_0}\right)\right) \tag{1.11}$$

lo cual converge en distribución a $N(0, \text{Var}\left(\frac{\partial}{\partial\theta}\left[\log f(x_i\,|\,\theta)\right]_{\theta=\theta_0}\right))$ por el Teorema de Límite Central.

Por otro lado, del denominador de (1.8) tenemos por la Ley de los Grandes Números que

$$l_n''(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2}{\partial\theta^2}\log f(x_i\,|\,\theta) \to \mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\log f(x\,|\,\theta)\right]. \tag{1.12}$$

Además como $\theta_1 \in [\hat{\theta}_n, \theta_0]$ y por consitencia tenemos que $\hat{\theta}_n \to \theta_0$ ocurre que

$$l_n''(\theta_1) \to \mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\left[\log f(x\,|\,\theta)\right]_{\theta=\theta=0}\right] = -I(\theta_0) \tag{1.13}$$

por el lema 1.2.1. Combinando 1.11 con 1.13 tenemos finalmente que

$$-\frac{\sqrt{n}l_n'(\theta_0)}{l_n''(\theta_1)} \to N\left(0, \frac{\text{Var}\left(\frac{\partial}{\partial\theta}\left[\log f(x\,|\,\theta)\right]_{\theta=\theta_0}\right)}{(I(\theta_0))^2}\right) \tag{1.14}$$

y notemos que

$$\text{Var}\left(\frac{\partial}{\partial\theta}\left[\log f(x\,|\,\theta)\right]_{\theta=\theta_0}\right) = \mathbb{E}\left[\frac{\partial}{\partial\theta}\left[\log f(x\,|\,\theta)\right]^2_{\theta=\theta_0}\right] - \mathbb{E}\left[\frac{\partial}{\partial\theta}\left[\log f(x\,|\,\theta)\right]_{\theta=\theta_0}\right]^2 = I(\theta_0) - 0.$$

□

1.3. Familias Exponenciales

Decimos que una función de densidad $p_X(\boldsymbol{x};\boldsymbol{\theta})$ con parámetro $\boldsymbol{\theta} = (\theta_1, ..., \theta_m)$ de un vector aleatorio X pertenece a una familia exponencial si puede ser expresada como

$$p_X(\boldsymbol{x};\boldsymbol{\theta}) = a(\boldsymbol{x})\exp\left\{\boldsymbol{b}(\boldsymbol{\theta})^T\boldsymbol{t}(\boldsymbol{x})\right\}, \tag{1.15}$$

donde $\boldsymbol{b}(\boldsymbol{\theta})$ es un vector de dimensión $k \times 1$ con $k \geq m$, que es una función del vector de parámetros $\boldsymbol{\theta}$, $\boldsymbol{t}(\boldsymbol{x})$ un vector $(k \times 1)$ de estadísticos suficientes y $c(\boldsymbol{\theta})$ y $a(\boldsymbol{x})$ son funciones escalares de $\boldsymbol{\theta}$ y de $\boldsymbol{x}$, respectivamente. En el caso en el que $\boldsymbol{b}(\boldsymbol{\theta}) = \boldsymbol{\theta}$, se dice que la familia exponencial está en forma canónica.

Dado que $p_X(\boldsymbol{x};\boldsymbol{\theta})$ es una función de densidad, se tiene lo siguiente:

$$c(\boldsymbol{\theta}) = \log\left(\int_{\mathcal{S}_X} a(\boldsymbol{x}) \exp\left\{\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{x})\right\} dx\right) \tag{1.16}$$

donde $\mathcal{S}_X$ es el soporte de X.

A continuación enunciamos un resultado que cumplen los estadísticos suficientes provenientes de familias exponenciales.

Proposición 1.3.1. *Sea X un vector aleatorio con función de densidad de la forma 1.15. Entonces*

$$\mathbb{E}\left[t_j(X)\right] = \frac{\partial c(\boldsymbol{\theta})}{\partial \theta_j},$$

para $j = (1, ..., m)$.

Demostración. Esto ocurre porque

$$\frac{\partial c(\boldsymbol{\theta})}{\partial \theta_j} = \frac{\frac{\partial}{\partial \theta_j} \int_{\mathcal{S}_X} a(\boldsymbol{x}) e^{\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{x})} dx}{\int_{\mathcal{S}_X} a(\boldsymbol{x}) e^{\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{x})} dx} \tag{1.17}$$

Bajo condiciones de regularidad podemos intercambiar diferenciación con integración, con lo cual

$$\begin{aligned} \frac{\partial c(\boldsymbol{\theta})}{\partial \theta_j} &= \frac{\int a(\boldsymbol{x}) e^{\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{x})} t_j(\boldsymbol{x}) dx}{\int a(\boldsymbol{x}) e^{\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{x})} dx} \\ &= \frac{\int a(\boldsymbol{x}) e^{\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{x}) - c(\boldsymbol{\theta})} t_j(\boldsymbol{x}) dx}{\int a(\boldsymbol{x}) e^{\boldsymbol{b}(\theta)^T \boldsymbol{t}(\boldsymbol{x}) - c(\boldsymbol{\theta})} dx} \\ &= \mathbb{E}\left[t_j(X)\right] \end{aligned} \tag{1.18}$$

□

La estimación por máxima verosimiltud para familias exponenciales se reduce a resolver las ecuaciones

$$t_j(\boldsymbol{x}) = \mathbb{E}\left[t_j(X)\right]. \tag{1.19}$$

La propiedad de invarianza de los estimadores máximo verosímiles es una propiedad suficiente para considerar la forma canónica de la familia exponencial a la hora de obtener dichos estimadores. Es decir, al maximizar a $\ell(\boldsymbol{\theta}; \boldsymbol{x})$

$$\frac{\partial \ell(\boldsymbol{\theta}|\boldsymbol{x})}{\partial \theta_j} = \boldsymbol{t}_j(\boldsymbol{x}) - \frac{\partial c(\boldsymbol{\theta})}{\partial \theta_j}, \tag{1.20}$$

igualando a cero y haciendo uso de la Proposición(1.3.1) nos da la igualdad de (1.19).

Ejemplo 1.3.1. Distribución Normal Univariada

Sea $\boldsymbol{x} = x_1, ..., x_n$ un muestra aleatoria con distribución normal con parámetros μ, σ^2. Entonces

$$\begin{aligned} p(\boldsymbol{x}; \mu, \sigma^2) &= (2\pi\sigma^2)^{-n/2} e^{\frac{-\sum_{i=1^n}(x_i-\mu)^2}{2\sigma^2}} \\ &= (2\pi\sigma^2)^{-n/2} e^{-(\sum_{i=1^n} x_i^2 + 2\sum_{i=1}^n x_i\mu - n\mu^2)\frac{1}{2\sigma^2}} \\ &= a(\boldsymbol{x}) \exp\left\{-\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{x}) - c(\boldsymbol{\theta})\right\} \end{aligned} \tag{1.21}$$

con

$$\begin{aligned} \boldsymbol{b}(\boldsymbol{\theta}) &= \left(\frac{\mu}{\sigma^2}, \frac{1}{2\sigma^2}\right) \\ \boldsymbol{t}(\boldsymbol{x}) &= \left(\sum_{i=1}^n x_i, \sum_{i=1}^n x_i^2\right)^T \\ a(x) &= 1 \\ c(\boldsymbol{\theta}) &= \frac{n^2\mu}{4\sigma^2} \log(2\pi\sigma^2). \end{aligned}$$

Por lo tanto, pertenece a la familia exponencial. Usando 1.19, debemos resolver para μ la ecuación

$$\begin{aligned} \sum_{i=1}^n x_i &= \mathbb{E}\left[\sum_{i=1}^n x_i\right] \\ &= n\mu \end{aligned} \tag{1.22}$$

Por lo tanto,

$$\begin{aligned} \hat{\mu} &= \frac{1}{n}\sum_{i=1}^n x_i \\ &= \bar{X}. \end{aligned} \tag{1.23}$$

Análogamente, para σ^2 tenemos

$$\begin{aligned} \sum_{i=1}^n x_i^2 &= \mathbb{E}\left[\sum_{i=1}^n x_i^2\right] \\ &= n(\sigma^2 + \mu^2), \end{aligned} \tag{1.24}$$

entonces resolviendo para σ^2

$$\begin{aligned} \hat{\sigma^2} &= \frac{1}{n}\sum_{i=1}^n x_i^2 - \bar{X}^2 \\ &= \frac{1}{n}\sum_{i=1}^n \left(x_i - \bar{X}\right)^2. \end{aligned} \tag{1.25}$$

Capítulo 2

Algoritmo Esperanza-Maximización

En este capítulo presentamos la formulación del algoritmo EM, en la cual introduciremos notación que se usará a lo largo del trabajo. Una vez mencionada la formulación del algoritmo EM se procede a mostrar algunas implementaciones, de las cuales se muestran los códigos realizados en el lenguaje `R` en el Apéndice.

Al final del capítulo presentamos ejemplos de aplicaciones del algoritmo EM para clasificación por medio de mezclas de densidades.

2.1. Formulación del algoritmo

El algoritmo Esperanza-Maximización es un procedimiento iterativo por medio del cual se obtienen estimadores máximo verosímiles. El procedimiento consiste en repetir los pasos de Esperanza y Maximización (abreviados como E y M, respectivamente), llamados así por el hecho de calcular una esperanza en el paso E y luego maximizar dicha esperanza en el paso M. Dichos pasos se repiten hasta conseguir la convergencia, obteniendo como resultado aproximaciones a los estimadores máximo verosímiles deseados.

El algoritmo es aplicado en situaciones en las cuales se tienen datos incompletos, por ejemplo, casos donde hay datos faltantes, distribuciones truncadas, observaciones censuradas, agrupadas, etc. Sin embargo, la idea de datos incompletos la extendemos a casos en los cuales la presencia de datos incompletos no es evidente.

Nos referiremos al vector de datos completos bajo la notación de $\boldsymbol{x}$ y denotaremos como X a la variable aleatoria correspondiente a dicho vector. Análogamente nos referiremos con $\boldsymbol{y}$ al vector de datos observados. Además, usaremos $\boldsymbol{z}$ para hacer referencia al vector de datos no observados. Entonces, podemos expresar a los datos completos como $\boldsymbol{x} = (\boldsymbol{y}, \boldsymbol{z})$.

La función de densidad conjunta de los datos completos la expresaremos todo el tiempo como $g_c(\boldsymbol{x}; \boldsymbol{\theta})$, mientras que $g(\boldsymbol{y}; \boldsymbol{\theta})$ se referirá a la función de densidad conjunta de los datos observados, siendo $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)$ el vector de parámetros desconocidos proveniente del espacio parametral $\boldsymbol{\Theta}$.

Dados dos espacios muestrales $\mathcal{X}$ y $\mathcal{Y}$, los datos observados los veremos en función de $\boldsymbol{x}$; es decir, $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})$, para $\boldsymbol{y} \in \mathcal{Y}$ y $\boldsymbol{x} \in \mathcal{X}$.

Entonces, podemos expresar a $g(\boldsymbol{y}; \boldsymbol{\theta})$ de la siguiente manera:

$$g(\boldsymbol{y};\boldsymbol{\theta}) = \int_{\mathcal{X}(y)} g_c(\boldsymbol{x};\boldsymbol{\theta})dx, \tag{2.1}$$

siendo $\mathcal{X}(y)$ el conjunto definido como $\mathcal{X}(y) := \{\boldsymbol{x} \in \mathcal{X} : \boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})\}$.

Queremos obtener los estimadores máximo verosímiles de los datos observados a través de la función de log-verosimilitud de los datos completos, que denotamos como $l_c(\boldsymbol{\theta};\boldsymbol{x})$. Sin embargo, dado que parte de los datos completos no son observados, usamos la esperanza con respecto de la función de densidad condicional de los datos completos dado los observados, denotada como

$$k(\boldsymbol{x} \mid \boldsymbol{y};\boldsymbol{\theta}) = \frac{g_c(\boldsymbol{x};\boldsymbol{\theta})}{g(\boldsymbol{y};\boldsymbol{\theta})}.$$

Dicha esperanza a la que denotaremos $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ está definida como

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) = \mathbb{E}\left[l_c(\boldsymbol{\theta};\boldsymbol{x}) \mid \boldsymbol{y};\boldsymbol{\theta}^{(t)}\right], \tag{2.2}$$

es decir, es la esperanza de la log-verosimilitud dados los datos completos, con respecto de la función de densidad condicional $k(\boldsymbol{x} \mid \boldsymbol{y};\boldsymbol{\theta}^{(t)})$, donde el superíndice (t) hace referencia al valor de $\boldsymbol{\theta}$ en la iteración t del algoritmo.

Esta es la esperanza que calculamos en el paso E del algoritmo, la cual procedemos a maximizar en $\boldsymbol{\Theta}$ en el paso M. La formulación general del algoritmo EM se muestra en el siguiente algoritmo:

Algoritmo 1 Algoritmo EM

1: Inicializar parámetros $\boldsymbol{\theta}^{(0)}$, $t = 0$.
2: **Paso E** Calcular $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) = \mathbb{E}\left[l_c(\boldsymbol{\theta};\boldsymbol{x}) \mid \boldsymbol{y};\boldsymbol{\theta}^{(t)}\right]$.
3: **Paso M** Maximizar $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ y obtener siguiente estimador como

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}).$$

4: $t = t + 1$ y regresar a 2.

El algoritmo se repite hasta que los estimadores converjan, entendiéndose por convergencia a que, dada cierta tolerancia $\epsilon > 0$ se tenga

$$\mid l_c(\boldsymbol{\theta}^{(t+1)};\boldsymbol{x}) - l_c(\boldsymbol{\theta}^{(t)};\boldsymbol{x}) \mid < \epsilon.$$

A continuación se presenta la formulación del algoritmo EM para el caso en el que la función de densidad de los datos completos sigue una distribución perteneciente a la familia exponencial.

2.1.1. Algoritmo EM para familias exponenciales

En el caso donde la función de densidad conjunta de los datos observados $g_c(\boldsymbol{x};\theta)$ es de la forma

$$g_c(\boldsymbol{x};\theta) = a(\boldsymbol{x})e^{\theta t(\boldsymbol{x})-c(\theta)}; \tag{2.3}$$

donde θ es un escalar. Es decir, cuando pertence a la familia exponencial regular en forma canónica, el procedimiento del algoritmo se simplifica mucho. Recordemos del capítulo anterior que

$$\mathbb{E}\left[t(\boldsymbol{x});\theta\right] = \frac{\partial c(\theta)}{\partial \theta}. \tag{2.4}$$

Además, notemos que

$$\begin{aligned} Q(\theta;\theta^{(t)}) &= \mathbb{E}\left[l_c(\theta;\boldsymbol{x}) \mid \boldsymbol{y};\theta^{(t)}\right] \\ &= \mathbb{E}\left[\log\left(g_c(\boldsymbol{x};\theta)\right) \mid \boldsymbol{y};\theta^{(t)}\right] \\ &= \theta\mathbb{E}\left[t(\boldsymbol{x}) \mid \boldsymbol{y};\theta^{(t)}\right] - c(\theta) + \text{cte}. \end{aligned} \tag{2.5}$$

Maximizando con respecto de θ obtenemos la igualdad

$$\mathbb{E}\left[t(\boldsymbol{x}) \mid \boldsymbol{y};\theta^{(t)}\right] = \frac{\partial c(\theta)}{\partial \theta}. \tag{2.6}$$

Entonces, dado (2.4), el paso M para obtener el estimador $\theta^{(t+1)}$ consiste en resolver

$$\mathbb{E}\left[t(\boldsymbol{x};\theta) \mid \boldsymbol{y};\theta^{(t)}\right] = \mathbb{E}\left[t(\boldsymbol{x});\theta\right] \tag{2.7}$$

para θ. Por lo tanto, el algoritmo EM para familias exponenciales lo definimos del siguiente modo:

Algoritmo 2 Algoritmo EM para Familias Exponenciales

1: Inicializar parámetros $\theta^{(0)}$, $t = 0$.
2: **Paso E** Calcular $\mathbb{E}\left[t(\boldsymbol{x}) \mid \boldsymbol{y};\theta^{(t)}\right]$.
3: **Paso M** Obtener $\theta^{(t+1)}$ resolviendo para θ la ecuación

$$\mathbb{E}\left[t(\boldsymbol{x}) \mid \boldsymbol{y};\theta^{(t)}\right] = \mathbb{E}\left[t(\boldsymbol{x});\theta\right].$$

4: $t = t + 1$ y regresar a 2.

La explicación antes mencionada para el caso multivariado $\boldsymbol{\theta}$ es análoga.

A continuación se presenta un poco de la historia del algoritmo, mencionando algunos artículos relevantes e ideas expuestas en ellos.

2.2. Historia e interpretación

Si bien el título del algoritmo fue concebido en el artículo hecho por Dempster, Laird y Rubin en 1977 [12], la idea de este algoritmo como tal ya había sido pensada y expuesta por distintos autores en una amplia gama de artículos que precedieron al DLR, (de ahora en adelante nos referiremos al artículo de Dempster, Laird y Rubin de 1977 con estas siglas). Sin embargo, la formulación general sí se la atribuímos a éste.

A continuación se presentan algunos trabajos que precedieron al DLR y que bien resultan ser algoritmos EM en contextos especiales.

En [14], Hartley menciona el problema de tener que lidiar con datos "incompletos" a la hora de hacer estimación máximo verosímil, dando como ejemplos los casos de muestras truncadas o censuradas. También expone que el cálculo de estimadores máximo verosímiles de información incompleta puede ser simplificada por medio de la estimación máximo verosímil de una muestra "completa". En este artículo, Hartley introduce un método que puede ser implementado en cualquier situación o problema en el cual el procedimiento para encontrar estimadores máximo verosímiles de la muestra completa es fácil de conseguir o está disponible, tratándose de casos de distribuciones discretas (datos de conteo) con frecuencias faltantes. Básicamente, el método iterativo que muestra con distintos ejemplos contiene la misma idea base del algoritmo EM, consistiendo en la actualización de valores estimados para la información faltante haciendo uso de los datos observados, para así, una vez teniendo los datos completos, proseguir con la estimación máximo verosímil, realizando estos dos pasos iterativamente hasta llegar a la convergencia.

El problema de estimación de valores faltantes también es tratado por S.F. Buck en [9]. Sin embargo, el tema se concentra solamente en la imputación, dejando a un lado la estimación de parámetros. El método propuesto por Buck consistía en, dada una muestra de datos p-variados, si para una observación ocurría que faltaban valores en k de las p variables, el valor esperado de dichos valores se calculaba por medio de una regresión múltiple con las demás $p - k$ variables del conjunto de observaciones completas.

Por otro lado, Efron [13] introduce en su trabajo el concepto de autoconsistencia para estimadores máximo verosímiles no paramétricos de funciones de supervivencia de datos censurados a la derecha. En [6] en 1970, Blight considera el problema de estimación por máxima verosimilitud de datos censurados del tipo I, pertenecientes a la familia exponencial multiparamétrica. Menciona también que un procedimiento iterativo para la estimación de los parámetros, así como propiedades de su convergencia. No provee una demostración de lo anterior; sin embargo, menciona una condición necesaria y suficiente para la convergencia, y es que el máximo de los eigenvalores de la matriz dada por $B_j(\hat{\theta}) = \frac{\partial E_j(\theta)}{\partial \theta}$ debe ser menor a 1. Ejemplifica el método con un conjunto de datos de mortalidad (doblemente) censurados distribuidos $N \sim (\mu, \sigma^2)$. Trata a la familia exponencial general y reconoce explícitamente la interpretación de los dos pasos de cada iteración EM.

Por otra parte, la gran aportación en el trabajo que realizaron conjuntamente Orchard y Woodbury [21] en 1972, fue justamente el principio de información faltante. Explican que los datos faltantes pueden ser reemplazados por valores muestrales provenientes de la función de

distribución apropiada. Dado un vector $X = (Y, Z)$ con Y representando a los datos observados y Z los faltantes, se tiene que

$$f(x \,|\, \theta) = f(y, z \,|\, \theta) = f(y \,|\, \theta) f(z \,|\, y; \theta),$$

siendo la segunda densidad del lado derecho la función de distribución requerida para simular o muestrear a los datos faltantes. Sin embargo, θ es desconocido, por lo que algún valor estimado $\hat{\theta}$ debe ser usado. Dado esto, se puede entonces obtener una relación entre la verosimilitud de los datos faltantes con la de los datos completos.

Años más tarde, en 1976 Turnbull [28] extiende el concepto de autoconsistencia de Efron para casos de datos doblemente censurados (a la derecha e izquierda) y muestra además que la idea de datos incompletos puede aplicarse para casos de información truncada y agrupada. Ese mismo año, Sundberg presenta en [26] un método iterativo aplicable a distribuciones de datos incompletos provenientes de familias exponenciales. Usa el término de "datos incompletos" para referirse a que solamente se observa el valor de una función $y = y(x)$, donde x representa el conjunto de información completa que no se obtuvo en su totalidad. Demuestra la convergencia de dicho método y aplica éste con ejemplos de datos de una muestra normal censurada, una muestra normal agrupada, una mezcla de dos muestras normales y un ejemplo de convolución de distribuciones normales y exponenciales. Dada la densidad característica de la familia exponencial

$$P_\alpha(x) = C(\alpha)^{-1} \exp\{\alpha t(x)\},$$

la iteración es entonces, dado un valor inicial α_0,

$$\alpha_{k+1} = m_t^{-1}(m_{t \,|\, y}(\alpha_k))$$

con $m_t(\alpha)$ la esperanza del vector de estadísticos suficientes t y $m_{t \,|\, y}(\alpha)$ la esperanza dado el vector de observaciones y. Para que este método sea de valor práctico, como bien aclaran los autores, $m_t(\alpha)$ debe ser invertible.

DLR 1977

Este famoso y celebrado artículo titulado "Maximum likelihood from incomplete data via the EM Algorithm" [12] fue presentado ante la Real Sociedad Estadística en 1976 y publicado en su revista en 1977. En él se sintetizaron todas las ideas y aplicaciones expuestas con anterioridad, estableciendo sus propiedades básicas y el desarrollo de la formulación general, así como la teoría de este algoritmo, presentando también una gran variedad de ejemplos y aplicaciones. Además, se le atribuyen otras acciones, como darle el nombre de EM-uno muy apropiado dada la naturaleza de sus pasos y procedimiento de Esperanza y Maximización; demostraron que puede ser aplicado en un sinnúmero de problemas, algunos de los cuales no se habían considerado relacionados a dicho algoritmo previamente, como el hecho de ver a las variables latentes como información faltante. También se le atribuye la idea de ver a los datos observados como un mapeo $\mathcal{X} \to \mathcal{Y}$ (como lo mecionó Sundberg [25] años antes) y el resultado sobre la convergencia monótona de la verosimilitud, sobre la cual hablaremos en siguientes capítulos.

A partir de su publicación, el interés hacia este algoritmo ha ido creciendo con el paso de los años y muchas extensiones han surgido, las cuales presentaremos en otros capítulos. Este algoritmo fue ganando popularidad y abriéndose paso en el repertorio de herramientas de un gran número de estadísticos.

2.3. Ejemplos e implementación en R

2.3.1. Muestra Normal Univariada

Consideremos a $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ una muestra aleatoria con distribución $N(\mu, \sigma^2)$, y supongamos que de $\boldsymbol{x}$ observamos una submuestra $\boldsymbol{y} = (x_1, ..., x_m)$ $m < n$ generada por muestreo simple sin reemplazo. Del Ejemplo 1.3.1 del capítulo anterior, sabemos que los estadísticos suficientes para $\boldsymbol{\theta} = (\mu, \sigma^2)$ están dados por $(\sum_{i=1}^m x_i, \sum_{i=1}^m x_i^2)$. Dado que la distribución normal pertence a la familia exponencial, el algoritmo se reduce a lo siguiente:

Algoritmo 3 Algoritmo EM para muestra normal univariada

1: Inicializar parámetros $\boldsymbol{\theta}^{(0)} = (\mu^{(0)}, \sigma^{2(0)})$, $t = 0$.

2: **Paso E** Calcular

$$s_1^{(t)} := \mathbb{E}\left[\sum_{i=1}^n x_i \,|\, \boldsymbol{y}; \boldsymbol{\theta}^{(t)}\right] = \sum_{i=1}^m x_i + (n-m)\mu^{(t)}$$

.

$$s_2^{(t)} := \mathbb{E}\left[\sum_{i=1}^n x_i^2 \,|\, \boldsymbol{y}; \boldsymbol{\theta}^{(t)}\right] = \sum_{i=1}^m x_i^2 + (n-m)(\sigma^{2(t)} + \mu^{2(t)})$$

3: **Paso M** Actualizar estimadores

$$\mu^{(t+1)} = \frac{s_1^{(t)}}{n}.$$

$$\sigma^{2(t+1)} = \frac{s_2^{(t)}}{n} - \left(\mu^{(t+1)}\right)^2$$

4: $t = t + 1$ y regresar a 2.

Se implementó un programa en R, haciendo uso del código (A.1) simulando una muestra de tamaño $n = 1000$ de la cual generamos una submuestra de tamaño $m = 500$, la cual hace referencia a los datos observados del problema. Se inicializaron los valores de los parámetros (μ, σ^2) de tres maneras distintas:

Valor inicial	Valor a iteración 20
$\mu^{(0)}$	$\mu^{(20)}$
0.5	2.989279
5	2.989284
10	2.989288
$\sigma^{2(0)}$	$\sigma^{2(20)}$
0.5	0.03984177
1	0.03984019
2	0.03988416

Tabla 2.1: Comparación de estimadores para μ y σ^2 obtenidos en la iteración 20 del algoritmo EM para el ejemplo de la muestra normal univariada, con distintos valores iniciales.

Comparando los resultados con los estimadores máximo verosímiles de (μ, σ^2) de la submuestra $\boldsymbol{y}$, cuyos valores son $(2.989282, 0.03983542)$, podemos notar que los resultados se aproximan bastante a dichos valores, como se muestra también en la Figura 2.1, donde la línea punteada representa el valor del estimador máximo verosímil.

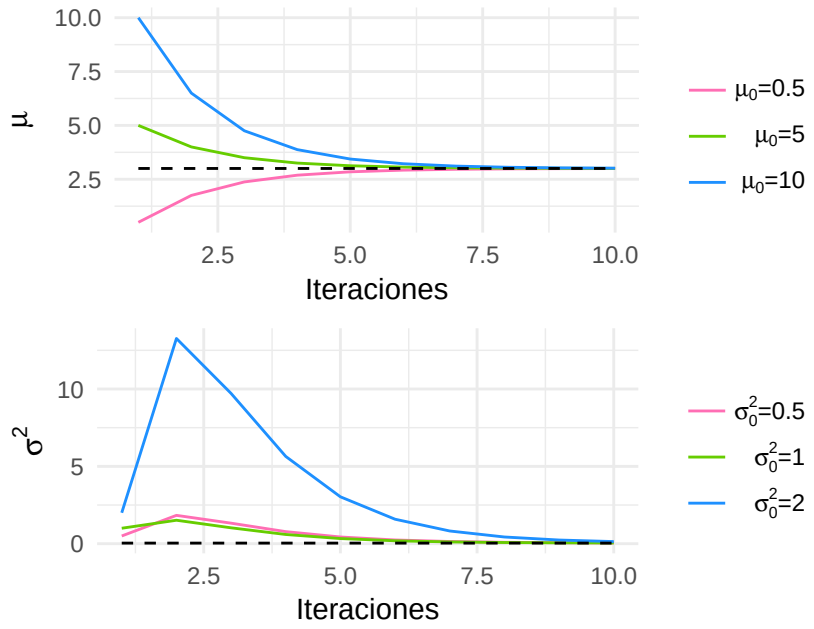

Figura 2.1: Convergencia de secuencia de valores generados por el algoritmo EM para obtener los estimadores máximo verosímiles de la muestra normal univariada observada. La línea punteada representa el valor del estimador máximo verosímil para μ y σ^2, respectivamente, de izquierda a derecha, que son $\hat{\mu} = 3.0013$ y $\hat{\sigma}^2 = 0.0393$, respectivamente. Software: R

2.3.2. Muestra de población multinomial

Ahora presentamos un ejemplo ilustrativo de datos observados provenientes de una población multinomial. Dichas observaciones son las frecuencias $\boldsymbol{y} = (38, 34, 125)$ con vector de probabilidades $\left(\frac{1}{2} - \frac{1}{2}\theta, \frac{1}{4}\theta, \frac{1}{4}\theta + \frac{1}{2}\right)$. Supondremos que los datos están incompletos y que los datos completos corresponden a una población con distribución multinomial con cuatro categorías $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$ y con probabilidades $\boldsymbol{p} = \left(\frac{1}{2} - \frac{1}{2}\theta, \frac{1}{4}\theta, \frac{1}{4}\theta, \frac{1}{2}\right)$, respectivamente. Notemos que los datos observados están en función de los datos completos del siguiente modo:

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_3 \\ y_3 &= x_3 + x_4 \end{aligned} \tag{2.8}$$

Entonces, la log-verosimilitud de los datos completos está dada por

$$l_c(\theta; \boldsymbol{x}) = \sum_{i=1}^{4} x_i \log(p_i) + \text{cte.}$$

Claramente, los estadísticos suficientes están dados por x_1, x_2, x_3, x_4 y dado que la distribución multinomial pertenece a la familia exponencial el paso E se reduce a obtener:

$$\begin{aligned} \mathbb{E}\left[x_1 \,|\, \boldsymbol{y}\right] &= x_1 \\ \mathbb{E}\left[x_2 \,|\, \boldsymbol{y}\right] &= x_2 \\ \mathbb{E}\left[x_3 \,|\, \boldsymbol{y}\right] &= 125\frac{\frac{1}{4}\theta}{\frac{1}{4}\theta + \frac{1}{2}} \end{aligned}$$

$$\mathbb{E}\left[x_4 \,|\, \boldsymbol{y}\right] = 125\frac{\frac{1}{2}}{\frac{1}{4}\theta + \frac{1}{2}}.$$

Las últimas dos esperanzas se deben a que los valores x_3 y x_4 condicionados a los datos observados, que es la suma de $x_3 + x_4$, siguen una distribución binomial con parámetros $(125, p)$, donde p vale

$$p = \frac{\frac{1}{4}\theta}{\frac{1}{4}\theta + \frac{1}{2}}$$

para x_3 y $1 - p$ para x_4.

Del paso M obtenemos que el estimador máximo verosímil de θ de los datos completos es

$$\hat{\theta} = \frac{x_2 + x_3}{x_1 + x_2 + x_3^{(t)}}, \tag{2.9}$$

pues

$$\frac{\partial Q(\theta; \theta^{(t)})}{\partial \theta} = -\frac{x_1}{1-\theta} + \frac{x_2}{\theta} + \frac{x_3^{(t)}}{\theta}$$

con $x_3^{(t)} = \mathbb{E}\left[x_3 \,|\, \boldsymbol{y}\right] = 125\frac{\frac{1}{4}\theta}{\frac{1}{4}\theta + \frac{1}{2}}$. Por lo tanto, maximizando sobre θ en $\Theta = (0, 1)$, obtenemos el valor dado en (2.9). Entonces, el algoritmo para este ejemplo es:

Algoritmo 4 Algoritmo EM para ejemplo de Población Multinomial

1: Inicializar parámetro $\theta^{(0)}$, $t = 0$.

2: **Paso E** Calcular $x_3^{(t)} = 125\frac{\frac{1}{4}\theta^{(t)}}{\frac{1}{4}\theta^{(t)} + \frac{1}{2}}$.

3: **Paso M** Obtener

$$\theta^{(t+1)} = \frac{x_2 + x_3^{(t)}}{x_1 + x_2 + x_3^{(t)}}$$

4: $t = t + 1$ y regresar a 2.

En la Figura 2.2 mostramos los resultados de convergencia al estimador máximo verosímil dado por $\theta_{MLE} = 0.626821497$, después de 9 iteraciones y con cuatro parámetros iniciales distintos. También presentamos en la Tabla 2.2 las primeras cinco iteraciones del algoritmo EM para los distintos valores iniciales usados.

2.3.3. Mezcla de distribuciones Binomial/Poisson

La Tabla 2.3 muestra el número de hijos de N viudas que recibirán cierta pensión.

Dado que la información obtenida resultó no ser consistente con el hecho de venir de una distribución Poisson (siendo muy grande el número de viudas que no tenían hijos), se propone adoptar el siguiente modelo como alternativa. Supongamos que una variable aleatoria discreta se distribuye como la mezcla de dos poblaciones, por lo que definimos a las poblaciones A y B como:

- **Población A:** Con probabilidad ξ, la variable aleatoria toma el valor de cero, y

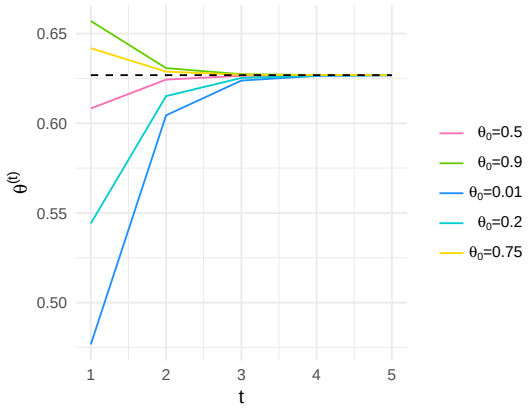

Figura 2.2: Convergencia de secuencias generadas por medio del algoritmo EM al estimador máximo verosímil del parámetro θ (línea punteada) del ejemplo de la población multinomial, usando distintos valores iniciales. Software: R

$\theta^{(0)}$	0.5	0.9	0.01	0.2	0.75
$\theta^{(1)}$	0.6082474	0.6570184	0.4767418	0.5441658	0.6418166
$\theta^{(2)}$	0.6243211	0.6307438	0.6044178	0.6151352	0.6287908
$\theta^{(3)}$	0.6264889	0.6273408	0.6237969	0.6252563	0.6270826
$\theta^{(4)}$	0.6267773	0.6268904	0.6264190	0.6266134	0.6268562
$\theta(5)$	0.6268156	0.6268306	0.6267680	0.6267939	0.6268261

Tabla 2.2: Convergencia de secuencia de estimadores que genera el algoritmo EM para θ del ejemplo de población multinomial

Número de hijos:	0	1	2	3	4	5	6
# Observado de viudas:	y_0	y_1	y_2	y_3	y_4	y_5	y_6

Tabla 2.3: Número observado de viudas por número de hijos

- **Población B:** Con probabilidad $1-\xi$, la variable aleatoria sigue una distribución Poisson con media λ.

Reformularemos el problema a uno con datos incompletos, suponiendo que el número de viudas sin hijos resulta de la suma de los casos que vienen de cada una de las poblaciones A o B, por lo que:

- $y_0 : x_A + x_B$
- x_A: Número de viudas sin hijos de la población A.
- x_B: Número de viudas sin hijos de la población B.

Notamos que este ejemplo es muy parecido al anterior en el sentido que uno de nuestros datos observados (y_0) proviene de una suma de dos de los valores de los datos completos $\boldsymbol{x} = (x_A, x_B, x_1, x_2, x_3, x_4, x_5, x_6)$, con lo cual la función de probabilidad de los datos completos queda del siguiente modo:

$$g_c(\boldsymbol{x};\boldsymbol{\theta}) = \binom{N}{x_A x_B x_1 x_2 x_3 x_4 x_5 x_6} (\xi + (1-\xi)e^{-\lambda})^{x_A + x_B} \prod_{i=1}^{6} \left((1-\xi)e^{-\lambda}\frac{\lambda^i}{i!} \right)^{x_i},$$

con $\boldsymbol{\theta} = (\xi, \lambda)$.

Por lo tanto, la función de log-verosimilitud de los datos completos está dada por:

$$\begin{aligned} l_c(\boldsymbol{\theta};\boldsymbol{x}) = \log(N) - \log(x_A!) - \log(x_B!) - \sum_{i=1}^{6} \log(x_i!) + (x_A + x_B)\log\left(\xi + (1-\xi)e^{-\lambda}\right) \\ + \sum_{i=1}^{6} x_i \left(\log(1-\xi) - \lambda\right) + \sum_{i=1}^{6} x_i \left(i\log(\lambda) - \sum_{k=1}^{i} \log(k) \right), \end{aligned} \tag{2.10}$$

con $\boldsymbol{\theta} = (\xi, \lambda)$.

Obteniendo de (2.10) los estadísticos suficientes $(x_A, x_B, x_1, x_2, x_3, x_4, x_5, x_6)$ y como podemos expresar la función de (2.10) de la forma mostrada en la Sección 1.3 el paso E está dado por

$$\mathbb{E}\left[x_i \,|\, \boldsymbol{y}\right] = x_i,$$

para $i = 1, 2, ..., 6$ y

$$\mathbb{E}\left[x_A \,|\, \boldsymbol{y}\right] = \frac{y_0 \xi}{\xi + (1-\xi)e - \lambda},$$

pues $x_A \,|\, \boldsymbol{y} \sim Bin(y_0, p_1)$, con $p_1 = \frac{p_A}{p_A + p_B}$ con $p_A = \xi$ y $p_B = (1-\xi)\exp\{-\lambda\}$. Análogamente, $x_B \,|\, \boldsymbol{y} \sim Bin(y_0, p_2)$ con $p_2 = \frac{p_B}{p_A + p_B}$, por lo que a cada iteración, el paso E se reduce al cálculo de:

$$x_A^{(t)} = \frac{y_0 \xi^{(t)}}{\xi^{(t)} + (1-\xi^{(t)})e^{-\lambda^{(t)}}}.$$

Ahora, para el paso M, necesitamos obtener los estimadores máximo verosímiles para ξ y λ. Para ello, observemos que $x_A \sim Bin(N, \xi)$ y los valores $(x_B, x_1, x_2, x_3, x_4, x_5, x_6)$ son frecuencias observadas de una distribución Poisson de parámetro λ. Por lo tanto,

$$\xi^{(t+1)} = \frac{x_A^{(t)}}{N}$$

$$\lambda^{(t+1)} = \frac{\sum_i i x_i}{x_B^{(t)} + \sum_i x_i},$$

con $x_B = y_0 - x_A$. El algoritmo para este ejemplo queda expresado del siguiente modo:

Algoritmo 5 Algoritmo EM para ejemplo de Mezclas Binomial/Poisson

1: Inicializar parámetros $\xi^{(0)}$, $\lambda^{(0)}$ $t = 0$.
2: **Paso E** Calcular

$$x_A^{(t)} = \frac{y_0 \xi^{(t)}}{\xi^{(t)} + (1 - \xi^{(t)}) e^{-\lambda^{(t)}}}$$

.
3: **Paso M** Obtener

$$\xi^{(t+1)} = \frac{x_A^{(t)}}{N},$$
$$\lambda^{(t+1)} = \frac{\sum_i i x_i}{x_B^{(t)} + \sum_i x_i}$$

4: $t = t + 1$ y regresar a 2.

Implementamos el algoritmo en R con los datos mostrados en la Tabla 2.4, obtenidos de [27] y mostramos los resultados obtenidos usando distintos valores iniciales en la Tabla 2.5 así como en la Figura 2.3 (ver código A.3 en Apéndice).

Número de hijos:	0	1	2	3	4	5	6
# Observado de viudas:	3,062	587	284	103	33	4	2

Tabla 2.4: Datos observados

Valor inicial	Convergencia en iteración	Valor a iteración t
$\xi^{(0)}$	t	$\xi^{(t)}$
0.75	36	0.6150566
0.4	63	0.6150566
0.1	74	0.6150566
0.86	52	0.6150568
$\lambda^{(0)}$	t	$\lambda^{(t)}$
0.4	36	1.037839
0.3	63	1.0378388
0.7	74	1.0378388
0.8	52	1.037839

Tabla 2.5: Convergencia de estimadores para ξ y λ

2.3.4. Componentes de varianza

En este ejemplo hacemos uso del algoritmo EM para la estimación de los componentes de varianza en un modelo de efectos aleatorios. Dado un vector de observaciones $\boldsymbol{y}$ de tamaño $N \times 1$, el modelo con un factor se expresa de la forma

$$\boldsymbol{y} = \mathbf{Z}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{2.11}$$

donde $\mathbf{Z}$ hace referencia a la matriz de diseño y $\boldsymbol{\alpha}$ es el vector de efectos aleatorios que cumple

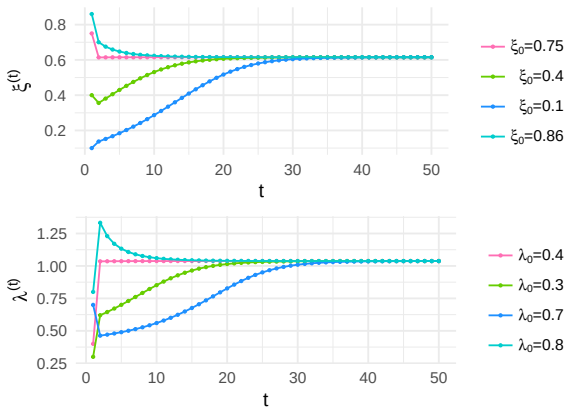

Figura 2.3: Convergencia de secuencia de estimadores obtenidos por medio del algoritmo EM para ξ, λ para el ejemplo de las viudas, con distintos valores iniciales. Software: R

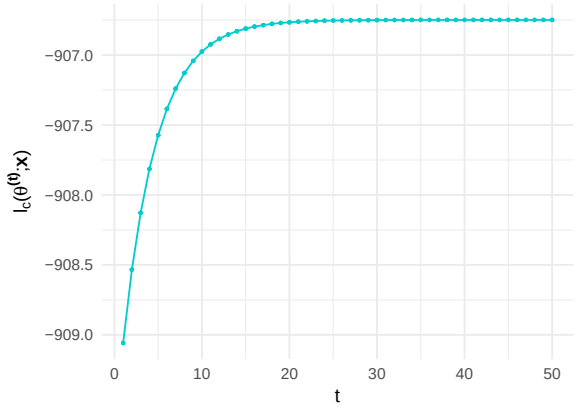

Figura 2.4: Convergencia de la función de log-verosimilitud. Software: R

$$\mathbb{E}\left[\boldsymbol{\alpha}\right] = \mathbf{1}_k \mu \tag{2.12}$$

$$\text{Var}\left(\boldsymbol{\alpha}\right) = \sigma_\alpha^2 \mathbf{I}_k, \tag{2.13}$$

con $\mathbf{1}_k$ siendo el vector de tamaño $k \times 1$ cuyas entradas valen todas 1 e $\mathbf{I}_k$ es la matriz identidad de tamaño $k \times k$, donde k es el número de efectos ó niveles. Además, $\boldsymbol{\epsilon}$ corresponde al vector de errores cuyas entradas están definidas como $\epsilon_{ij} = y_{ij} - \mathbb{E}\left[y_{ij} \,|\, \alpha_i\right] = y_{ij} - \alpha_i$ y es tal que

$$\mathbb{E}\left[\boldsymbol{\epsilon}\right] = \mathbf{0} \tag{2.14}$$

$$\text{Var}\left(\boldsymbol{\epsilon}\right) = \sigma_\epsilon^2 \mathbf{I}_N \tag{2.15}$$

$$\text{Cov}\left(\boldsymbol{\alpha}\,,\, \boldsymbol{\epsilon}\right) = \mathbf{0}. \tag{2.16}$$

Usaremos en este ejemplo la notación y_{ij} para referirnos a la $j-$ésima observación del $i-$ésimo efecto; es decir no representa la entrada i, j de una matriz, como la notación sugiere en un principio, sino que hace referencia al número de observación j del efecto i.

Por ejemplo, dados los datos que se muestran en la Tabla 2.6, el modelo (2.11) se ve como en (2.17).

Nivel i		y_{ij}		n_i
$i=1$	y_{11}	y_{12}	y_{13}	3
$i=2$	y_{21}	y_{22}		2

Tabla 2.6: Ejemplo para aclarar la notación.

$$\boldsymbol{y} = \begin{bmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{22}, \end{bmatrix} \tag{2.17}$$

con lo cual se cumple que

$$\begin{aligned} \mathbb{E}[\boldsymbol{y}] &= \mathbf{1}_N \mu & (2.18) \\ \mathbf{V} &= \mathrm{Var}(\boldsymbol{y}) \\ &= \mathrm{Var}(\mathbf{Z}\boldsymbol{\alpha} + \boldsymbol{\epsilon}) \\ &= \mathbf{Z}\mathrm{Var}(\boldsymbol{\alpha})\mathbf{Z}^T + \mathrm{Var}(\boldsymbol{\epsilon}) \\ &= \mathbf{Z}\sigma_\alpha^2 \mathbf{I}_k \mathbf{Z}^T + \sigma_\epsilon^2 \mathbf{I}_N. & (2.19) \end{aligned}$$

Dado que asumimos normalidad, el modelo se distribuye normal con parámetros $(\mathbf{1}_N\mu, \mathbf{V})$.

En este caso, la implementación del algoritmo parte de ver a las observaciones- las cuales queremos explicar por medio de este modelo, como el conjunto de datos incompletos, siendo entonces los datos completos aquellos compuestos por las observaciones y los efectos aleatorios no observados. Por lo tanto, necesitaremos la distribución conjunta de los datos completos, es decir, del vector $\boldsymbol{x} = (\boldsymbol{y}, \boldsymbol{\alpha})^T$. Su función de densidad $g_c(\boldsymbol{x}; \boldsymbol{\theta})$ la podemos ver como el siguiente producto de funciones

$$\begin{aligned} g_c(\boldsymbol{x}; \boldsymbol{\theta}) = (2\pi\sigma_\epsilon^2)^{-N/2} \exp\left\{ -\frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\alpha_i)^2}{2\sigma_\epsilon^2} \right\} \times \\ (2\pi\sigma_\alpha^2)^{-k/2} \exp\left\{ -\frac{\sum_{i=1}^{k}(\alpha_i-\mu)^2}{2\sigma_\alpha^2} \right\}. \end{aligned} \tag{2.20}$$

Notemos que los estadísticos suficientes para $\theta = (\mu, \sigma_\alpha^2, \sigma_\epsilon^2)$ son respectivamente

$$\begin{aligned} s_1 &= \sum_{i=1}^{k} \alpha_i \\ s_2 &= \sum_{i=1}^{k} \alpha_i^2 \\ s_3 &= \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\alpha_i)^2 \\ &= \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2 + \sum_{i=1}^{k} n_i(\bar{y}_i-\alpha_i)^2. & (2.21) \end{aligned}$$

Dicha función conjunta de densidad, notando además que

$$\begin{aligned}\mathrm{Cov}\left(\boldsymbol{y}\,,\,\boldsymbol{\alpha}\right) &= \mathrm{Cov}\left(\mathbf{Z}\boldsymbol{\alpha}+\boldsymbol{\epsilon}\,,\,\boldsymbol{\alpha}\right)\\ &= \mathbf{Z}\mathrm{Var}\left(\boldsymbol{\alpha}\right)\\ &= \mathbf{Z}\sigma_\alpha^2\mathbf{I}_k\end{aligned} \tag{2.22}$$

corresponde a una distribución normal con parámetros $(\boldsymbol{\mu}, \Sigma)$, donde $\boldsymbol{\mu}$ y $\boldsymbol{\Sigma}$ son un vector y una matriz de tamaño $(N+k)\times 1$ y $(N+k)\times(N+k)$, respectivamente y son de la forma

$$\boldsymbol{\mu} = \begin{bmatrix}\mathbf{1}_N\mu\\ \mathbf{1}_k\mu\end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix}\mathbf{V} & \mathbf{Z}\sigma_\alpha^2\mathbf{I}_k\\ \left(\mathbf{Z}\sigma_\alpha^2\mathbf{I}_k\right)^T & \sigma_\alpha^2\mathbf{I}_k\end{bmatrix}. \tag{2.23}$$

También necesitaremos para el paso E la esperanza condicional de los estadísticos suficientes, dados los datos observados, $\boldsymbol{y}$. Notemos de (2.18) que $\mathbf{V}$ es una matriz diagonal por bloques, cuyos bloques de la diagonal $\mathbf{V}_i$ son matrices de la forma $\mathbf{V}_i = \sigma_\epsilon^2\mathbf{I}_{n_i} + \sigma_\alpha^2\mathbf{J}_{n_i}$, siendo $\mathbf{J}_{n_i}$ la matriz de dimensiones $n_i \times n_i$ cuyas entradas valen todas 1. Por lo tanto, su matriz inversa $\mathbf{V}^{-1}$ es una matriz diagonal cuyas entradas en la diagonal son matrices de la forma $\mathbf{V}_i^{-1} = \frac{1}{\sigma_\epsilon^2}\left(\mathbf{I}_{n_i} - \frac{\sigma_\alpha^2}{\sigma_\epsilon^2+n_i\sigma_\alpha^2}\mathbf{J}_{n_i}\right)$.

Usando un resultado presentado en [29], la distribución condicional de $\boldsymbol{\alpha}$ dado $\boldsymbol{y}$ es

$$\boldsymbol{\alpha}\,|\,\boldsymbol{y} \sim N(\mathbf{1}_k\mu + \sigma_\alpha^2\mathbf{Z}^T\mathbf{V}^{-1}(\boldsymbol{y}-\mathbf{1}_N\mu), \sigma_\alpha^2\mathbf{I}_k - \sigma_\alpha^4\mathbf{Z}^T\mathbf{V}^{-1}\mathbf{Z}) \tag{2.24}$$

con lo cual cada a_i se distribuye $N\left(\frac{\sigma_\epsilon^2\mu + n_i\sigma_a^2\bar{y}_i}{\sigma_\epsilon^2+n_i\sigma_a^2}, \frac{\sigma_\epsilon^2\sigma_a^2}{\sigma_\epsilon^2+n_i\sigma_a^2}\right)$ con $\bar{y}_i = \frac{1}{n}\sum_{j=1}^{n_i} y_{ij}$.

Por lo tanto, el algoritmo EM se plantea de la siguiente manera:

Toro(i)	Porcentajes de concepción	n_i
1	$46, 31, 37, 62, 30$	5
2	$70, 59$	2
3	$52, 44, 57, 40, 67, 64, 70$	7
4	$47, 21, 70, 46, 14$	5
5	$42, 64, 50, 69, 77, 81, 87$	7
6	$35, 68, 59, 38, 57, 76, 57, 29, 60$	9
Total		35

Tabla 2.7: Datos de porcentajes de concepción

Algoritmo 6 Algoritmo EM para Componentes de Varianza

1: Inicializar parámetros $\boldsymbol{\theta}^{(0)} = (\mu^{(0)}, \sigma_\alpha^{2(0)}, \sigma_\epsilon^{2(0)})$, $t = 0$.

2: **Paso E** Calcular

$$\begin{aligned} s_1 &= \mathbb{E}\left[\sum_{i=1}^{k} \alpha_i \,|\, \boldsymbol{y}; \boldsymbol{\theta}^{(t)}\right] \\ &= \sum_{i=1}^{k} \frac{\sigma_\epsilon^{2(t)}\mu^{(t)} + n_i\sigma_\alpha^{2(t)}\bar{y}_i}{\sigma_\epsilon^{2(t)} + n_i\sigma_\alpha^{2(t)}} \\ s_2 &= \mathbb{E}\left[\sum_{i=1}^{k} \alpha_i^2 \,|\, \boldsymbol{y}; \boldsymbol{\theta}^{(t)}\right] \\ &= \sum_{i=1}^{k} \frac{\sigma_\epsilon^{2(t)}\sigma_a^{2(t)}}{\sigma_\epsilon^{2(t)} + n_i\sigma_\alpha^{2(t)}} + \sum_{i=1}^{k}\left(\frac{\sigma_\epsilon^{2(t)}\mu^{(t)} + n_i\sigma_a^{2(t)}\bar{y}_i}{\sigma^{2(t)} + n_i\sigma_\alpha^{2(t)}}\right)^2 \\ s_3 &= \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i)^2 + \sum_{i=1}^{k} n_i\left(\left(u_i^{(t)}\right)^2(\mu^{(t)} - \bar{y}_i)^2 + v_i^{(t)}\right) \end{aligned}$$

con $u_i^{(t)} = \frac{\sigma_\epsilon^{2(t)}}{\sigma_\epsilon^{2(t)} + n_i\sigma_\alpha^{2(t)}}$, $v_i = u_i^{(t)}\sigma_\alpha^{2(t)}$

3: **Paso M** Obtener

$$\mu^{(t+1)} = \frac{s_1^{(t)}}{k}$$

$$\sigma_\alpha^{2(t+1)} = \frac{s_2^{(t)}}{k} - \left(\mu^{(t+1)}\right)^2$$

$$\sigma_\epsilon^{2(t+1)} = \frac{s_3^{(t)}}{N}$$

4: $t = t + 1$ y regresar a 2.

A continuación se presentan los resultados de la implementación en R de dicho modelo (ver Código A.4), haciendo uso de los datos presentados en Snedecor y Cochran (1967) [24], sobre porcentajes de concepción de distintas muestras de semen de toros para experimentos de inseminación artificial, variando el número de muestras por toro, como se muestra en la Tabla 2.7

En la Figura 2.5 mostramos los resultados de la convergencia de los estimadores generados por el algoritmo, usando distintos valores iniciales como se muestra en las leyendas de cada gráfica. Obtuvimos como estimadores $\hat{\mu} = 53.59$, $\hat{\sigma}_a^2 = 57.42$ y $\hat{\sigma}_\epsilon^2 = 248.39$, comparados con

los estimadores obtenidos por medio del análisis ANOVA, cuyos resultados son $\sigma_a^2 = 73.41$ y $\sigma_\epsilon^2 = 248.29$. Refiérase al código (A.4) en el Apéndice.

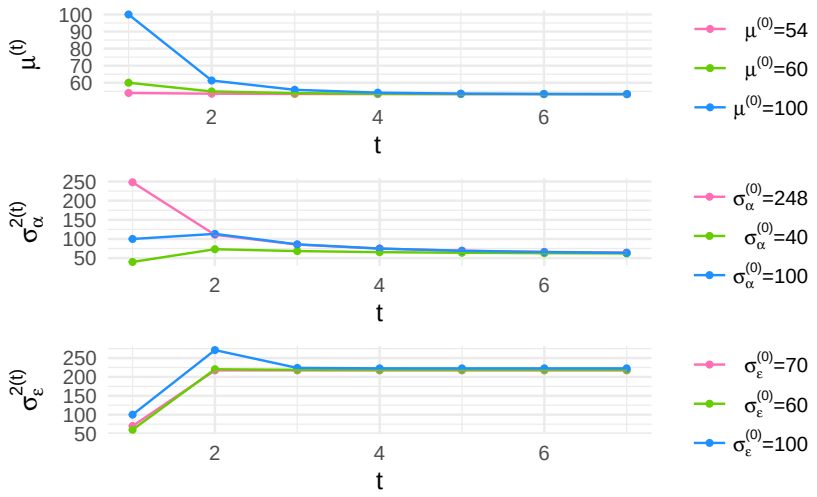

Figura 2.5: Convergencia de estimadores para $(\mu, \sigma_\alpha^2, \sigma_\epsilon^2)$ con distintos valores iniciales. Software: R

2.4. Aplicaciones

2.4.1. Densidades mixtas

Una variable o vector aleatorio Y perteneciente a una mezcla de M distibuciones tiene función de densidad

$$p_Y(\boldsymbol{y} \mid \boldsymbol{\theta}) = \sum_{k=1}^{M} \alpha_k p_k(\boldsymbol{y} \mid \boldsymbol{\theta}_k) \tag{2.25}$$

donde las α_k tales que $\sum_{k=1}^{M} \alpha_k = 1$ son los pesos asignados a las densidades $p_k(\boldsymbol{y} \mid \boldsymbol{\theta}_k)$. La función de log-verosimilitud está dada por

$$l(\boldsymbol{\theta} \mid \boldsymbol{y}) = \sum_{i=1}^{N} \log \sum_{k=1}^{M} \alpha_k p_k(\boldsymbol{y} \mid \boldsymbol{\theta}_k). \tag{2.26}$$

Podemos simplificar este problema si consideramos variables no observadas las cuales, de tenerlas nos dirían de qué distribución proviene cada una de las observaciones $\boldsymbol{y}_i$. Sea $\boldsymbol{z} = z_1, ..., z_N$ un vector de realizaciones de la variable $\mathcal{Z}$, tales que si $z_i = k$, para $k = 1, ..., M$ entonces la observación $\boldsymbol{y}_i$ habrá sido generada por la k-ésima distribución. Dado esto, la log-verosimilitud de la información completa $(\boldsymbol{y}, \boldsymbol{z})$ nos queda:

$$\begin{aligned} l_c(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{z}) &= \log\left(g_c(\boldsymbol{y}, \boldsymbol{z}; \boldsymbol{\theta})\right) \\ &= \sum_{i=1}^{N} \log\left(f_1(y_i \mid z_i; \theta) f_2(z_i; \theta)\right) \end{aligned} \tag{2.27}$$

donde $f_1(y_i \mid z_i; \theta) = p_{z_i}(\boldsymbol{y} \mid \boldsymbol{\theta}_{z_i})$ es el equivalente a la función $p_k(\boldsymbol{y} \mid \boldsymbol{\theta}_k)$ en (2.25), solamente que ahora vista en presencia de datos no observados que por ende traducimos a volver aleatorio el

valor de k, por lo cual los pesos α_k se convertirán en una función de probabilidad; es decir, $f_2(z_i;\theta) = \alpha_{z_i}$.

Entonces 2.27 la expresamos como

$$l_c(\boldsymbol{\theta} \,|\, \boldsymbol{y}, \boldsymbol{z}) = \sum_{i=1}^{N} \log\left(p_{z_i}(x_i;\theta)\alpha_{z_i}\right). \tag{2.28}$$

Recordemos que la esperanza calculada en el paso E del algoritmo EM es aquella con respecto de la función de densidad condicional de los datos completos dado los observados. Para ello, primero obtengamos la siguiente probabilidad

$$\begin{aligned}\mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right) &= \frac{f_1(y_i \,|\, z_i;\theta) f_2(z_i;\theta)}{p_Y(y_i;\boldsymbol{\theta})} \\ &= \frac{\alpha_{z_i} p_{z_i}(x_i;\boldsymbol{\theta}_{z_i})}{\sum_{k=1}^{M} \alpha_k p_k(y_i;\boldsymbol{\theta}_k)}.\end{aligned} \tag{2.29}$$

Notemos que la esperanza que se calculará en el paso E del algoritmo dependerá de la probabilidad posterior de la variable $\mathcal{Z}$, dadas las observaciones $\boldsymbol{x}_i$, con $i = 1, ..., n$.

Por lo tanto, la densidad conjunta $k(\boldsymbol{x} \,|\, \boldsymbol{y};\boldsymbol{\theta})$ es

$$\begin{aligned}k(\boldsymbol{x} \,|\, \boldsymbol{y};\boldsymbol{\theta}) &= p(\boldsymbol{z} \,|\, \boldsymbol{y};\boldsymbol{\theta}) \\ &= \prod_{i=1}^{N} \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right).\end{aligned} \tag{2.30}$$

Entonces, el paso E nos queda como

$$\begin{aligned}Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) &= \sum_{\boldsymbol{z}\in\mathcal{Z}} \sum_{i=1}^{N} \log \alpha_{z_i} p_{z_i}(y_i;\boldsymbol{\theta}) k(\boldsymbol{x} \,|\, \boldsymbol{y};\boldsymbol{\theta}) \\ &= \sum_{z_1=1}^{M} \sum_{z_2}^{M} ... \sum_{z_n=1}^{M} \sum_{i=1}^{N} \sum_{k=1}^{M} \mathbb{1}(z_i = k) \log \alpha_k p(y_i;\boldsymbol{\theta}) \prod_{i=1}^{N} \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right) \\ &= \sum_{i=1}^{N} \sum_{k=1}^{M} \log\left(\alpha_k p_k(y_i;\boldsymbol{\theta})\right) \sum_{z_1=1}^{M} \sum_{z_2=1}^{M} ... \sum_{z_N=1}^{M} \mathbb{1}(z_i = k) \prod_{i=1}^{N} \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right)\end{aligned} \tag{2.31}$$

Veamos ahora que $\sum_{z_1=1}^{M} \sum_{z_2=1}^{M} ... \sum_{z_N=1}^{M} \mathbb{1}(z_i = k) \prod_{i=1}^{N} \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right)$ es igual a:

$$\begin{aligned}\sum_{z_1=1}^{M} ... \sum_{z_{i-1}=1}^{M} \sum_{z_{i+1}=1}^{M} ... \sum_{z_N=1}^{M} \prod_{\substack{j=1 \\ j\neq i}}^{N} p(z_j \,|\, y_j;\boldsymbol{\theta}) p(z_i = k \,|\, y_i;\boldsymbol{\theta}) &= \prod_{\substack{j=1 \\ j\neq i}}^{N} \Big(\sum_{z_j=1}^{M} p(z_j \,|\, y_j;\boldsymbol{\theta})\Big) p(z_i = k \,|\, y_i;\boldsymbol{\theta}) \\ &= \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right)\end{aligned} \tag{2.32}$$

Por lo cual la esperanza se simplifica a $\sum_{i=1}^{N} \sum_{k=1}^{M} \log \alpha_k p_k(y_i;\boldsymbol{\theta}_k) \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right)$ y por lo tanto:

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{N} \sum_{k=1}^{M} \log \alpha_k \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right) + \sum_{i=1}^{N} \sum_{k=1}^{M} \log p(y_i \,|\, z_i = k;\boldsymbol{\theta}) \mathbb{P}\left(z_i = k \,|\, y_i;\boldsymbol{\theta}\right). \tag{2.33}$$

Esto nos facilita mucho las cosas, pues la esperanza que deseamos maximizar en el paso M la dividimos en dos términos independientes, con lo cual podremos optimizar los parámetros α y θ de manera separada.

Usemos el primer término de 2.33 para maximizar con respecto de α_k para $k = 1, 2, ..., M$. Dada la restricción de que $\sum_{k=1}^{M} \alpha_k = 1$ y derivando el lagrangiano, el parámetro óptimo en la iteración $t + 1$ que obtenemos es:

$$\alpha_k^{(t+1)} = \frac{\sum_{i=1}^{N} \alpha_k^{(t)} p(x_i \mid z_i = k, \theta^t)}{\sum_{k=1}^{M} \alpha^{(t)} p(x_i \mid z_i = k; \theta^{(t)}) n}. \tag{2.34}$$

Ahora, para obtener el parámetro θ de manera explícita, dado que la clasificación que haremos en el ejemplo que se mostrará se basa en el modelo de gaussianas mixtas, supondremos que las observaciones $\boldsymbol{y}_i$, $i = 1, ..., n$ siguen una distribución normal multivariada de d dimensiones, con parámetros (μ_k, Σ_k) dependientes de que la observación provenga de la clase k, es decir,

$$p(\boldsymbol{y}_i \mid z_i = k; \boldsymbol{\theta}) = (2\pi)^{-d/2} \mid \Sigma_k \mid^{-1/2} \exp\left\{-\tfrac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)\right\}. \tag{2.35}$$

Entonces en este contexto, $\boldsymbol{\theta}_k = (\mu_k, \Sigma_k)$, por lo cual para encontrar los parámetros óptimos haremos uso del segundo término de (2.33). Derivando (2.33) con respecto de μ_k e igualando a cero, obtenemos:

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{N} \boldsymbol{y}_i \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right)}{\sum_{i=1}^{N} \mathbb{P}\left(z_i = k \mid \boldsymbol{y}_i; \boldsymbol{\theta}^{(t)}\right)}. \tag{2.36}$$

Por último para obtener el valor óptimo de Σ_k de la $t + 1$ iteración, notemos primero que:

$$\begin{aligned}
&\sum_{k=1}^{M} \sum_{i=1}^{N} \log(\mid \Sigma_k \mid^{-1/2} \exp\left\{-\tfrac{1}{2}(y_i - \mu_k)^T \Sigma_k^{-1} (y_i - \mu_k)\right\}) \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right) \\
&= \sum_{k=1}^{M} \sum_{i=1}^{N} (\tfrac{-1}{2} \log \mid \Sigma_k \mid -\tfrac{1}{2}(y_i - \mu_k)^T \Sigma_k^{-1} (y_i - \mu_k)) \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right) \\
&= \sum_{k=1}^{M} \left[\sum_{i=1}^{N} \tfrac{-1}{2} \log \mid \Sigma_k \mid \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right) - \tfrac{1}{2} \sum_{i=1}^{N} \operatorname{tr}(\Sigma_k^{-1} (y_i - \mu_k)(y_i - \mu_k)^T) \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right) \right] \\
&= \sum_{k=1}^{M} \left[\sum_{i=1}^{N} \tfrac{1}{2} \log \mid \Sigma_k^{-1} \mid \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right) - \tfrac{1}{2} \sum_{i=1}^{N} \operatorname{tr}(\Sigma_k^{-1} (y_i - \mu_k)(y_i - \mu_k)^T) \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right) \right]
\end{aligned} \tag{2.37}$$

Derivando la última expresión con respecto de Σ_k, el óptimo en la iteración $t + 1$ nos queda:

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{N} \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right) (y_i - \mu_k^{(t)})(y_i - \mu_k^{(t)})^T}{\sum_{i=1}^{n} \mathbb{P}\left(z_i = k \mid y_i; \boldsymbol{\theta}^{(t)}\right)} \tag{2.38}$$

Por lo tanto, el algoritmo EM para este ejemplo queda expresado del siguiente modo

Algoritmo 7 Algoritmo EM para mezcla de densidades

1: Inicializar parámetros $\boldsymbol{\theta}_i^{(0)} = (\alpha_i^{(0)}, \mu_i^{(0)}, \Sigma_i^{(0)})$ para i en $\{1, 2, ..., M\}$, $t = 0$

2: **Paso E** Calcular $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{N} \sum_{k=1}^{M} \log \alpha_k \mathbb{P}\left(z_i = k \,|\, y_i; \boldsymbol{\theta}^{(T)}\right) + \sum_{i=1}^{N} \sum_{k=1}^{M} \log p(y_i \,|\, z_i = k; \boldsymbol{\theta}^{(t)}) \mathbb{P}\left(z_i = k \,|\, y_i; \boldsymbol{\theta}^{(t)}\right)$.

3: **Paso M** Actualizar parámetros

$$\alpha_k^{(t+1)} = \frac{1}{n} \frac{\sum_{i=1}^{N} \alpha_k^{(t)} p(y_i \,|\, z_i = k; \boldsymbol{\theta}^{(t)})}{\sum_{k=1}^{M} \alpha^{(t)} p(x_i \,|\, z_i = k, \boldsymbol{\theta}^{(t)})}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{n} \mathbb{P}\left(z_i = k \,|\, y_i; \boldsymbol{\theta}^{(t)}\right) (y_i - \mu_k^{(t)})(y_i - \mu_k^{(t)})^T}{\sum_{i=1}^{n} \mathbb{P}\left(z_i = k \,|\, y_i; \boldsymbol{\theta}^{(t)}\right)}$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} y_i \mathbb{P}\left(z_i = k \,|\, y_i; \boldsymbol{\theta}\right)^{(t)}}{\sum_{i=1}^{n} \mathbb{P}\left(z_i = k \,|\, y_i; \boldsymbol{\theta}\right)}.$$

4: Mientras $\mid Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t+1)}) - Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) \mid >$ tolerancia, $t = t + 1$ y volver a 2.

Utilizaremos ahora este algoritmo y su aplicación en densidades mixtas para crear un modelo de clasificación al famoso conjunto de datos de flores iris de Ronald Fisher (disponible en el software R). Dicho conjunto de datos consiste en 50 muestras de cuatro atributos de tres distintas especies de flor Iris: Iris versicolor, Iris setosa e Iris virginica. Los atributos son: ancho y largo del sépalo y ancho y largo del pétalo.

Para poder ilustrar el problema haremos uso de dos de las cuatro variables: ancho y largo del pétalo y supondremos que cada una de las observaciones provienen de una distribución gaussiana mixta, de la cual nuestro objetivo es obtener los parámetros óptimos que mejor clasifiquen a nuestros datos.

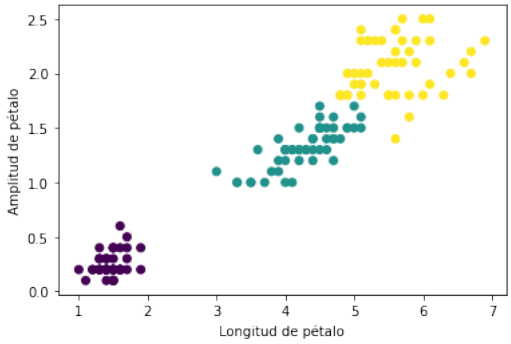

Figura 2.6: Visualización del conjunto de datos "Iris" de las variables longitud y ancho de pétalo. Software: Python

Inicializando los parámetros con los valores $\boldsymbol{\alpha} = (1/3, 1/3, 1/3)$ $\boldsymbol{\mu} = (1.4, 0.2), (4.5, 1.6), (4.8, 1.8)$ y matrices identidad para las matrices de covarianzas, después de 35 iteraciones se llegó a los valores mostrados en la Tabla 2.8 y matrices de covarianzas

$$\Sigma_1 = \begin{bmatrix} 0.03 & 0.006 \\ 0.006 & 0.011 \end{bmatrix} \tag{2.39}$$

$$\Sigma_2 = \begin{bmatrix} 0.256 & 0.092 \\ 0.092 & 0.051 \end{bmatrix} \tag{2.40}$$

Variable	Valor final
α_1	0.33
α_2	0.37
α_3	0.30
μ_1	$(1.46, 0.24)$
μ_2	$(4.33, 1.36)$
μ_3	$(5.60, 2.06)$

Tabla 2.8: Estimadores finales de la secuencia generada por el algoritmo EM

$$\Sigma_3 = \begin{bmatrix} 0.292 & 0.037 \\ 0.037 & 0.070 \end{bmatrix} \tag{2.41}$$

con una precisión del 97 %. En las Figuras (2.7) y (2.8) se muestran las gráficas de la distribución de mezclas gaussianas, usando los estimadores a los que convergieron las secuencias generadas por el algoritmo EM, mostrados en la Tabla 2.8 siguiendo los pasos del Algoritmo 7.

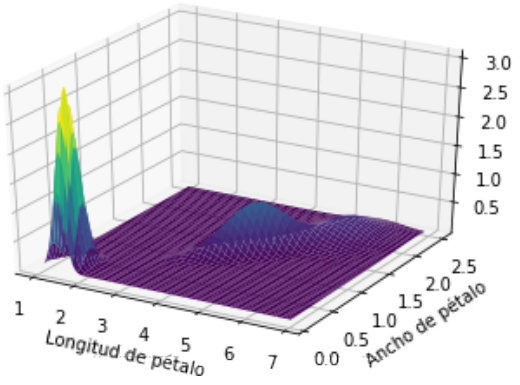

Figura 2.7: Modelo de gaussianas mixtas para el conjunto de datos “Iris”. Software: Python.

2.4.2. Clasificación de un modelo de riesgo modulado

Continuaremos con la formulación del algoritmo EM para estimación de parámetros de densidades mixtas, aplicado a una versión modificada del modelo clásico de riesgo que se enuncia a continuación.

A inicios del siglo XX, el matemático y actuario sueco Filip Lundberg propone en su tesis doctoral un modelo para la evolución del capital de una compañía aseguradora, con lo cual introduce el proceso de Poisson compuesto. Años más tarde, Harald Cramér formaliza dicho modelo, al cual conocemos actualmente como modelo de Cramér-Lundberg, dado por el proceso estocástico del capital a tiempo t, $\boldsymbol{C} = \{C_t\}_{t\geq 0}$

$$C_t = c + p - \sum_{i=1}^{N_t} X_i, \tag{2.42}$$

donde c es el capital inicial, p es el incremento en primas, N_t es el número de reclamaciones hasta el tiempo t, X_i es una variable aleatoria con soporte en los reales positivos, la cual se

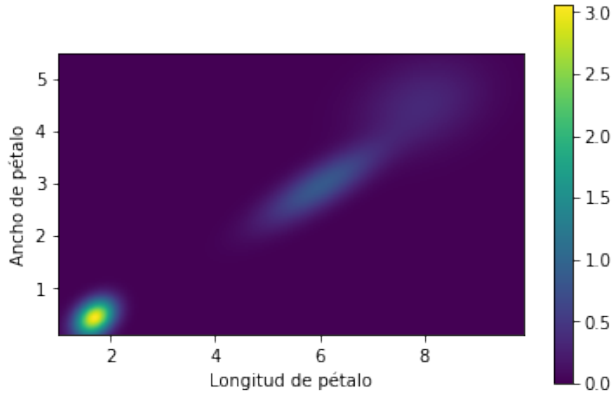

Figura 2.8: Gráfica de contorno de la densidad gaussiana mixta con parámetros óptimos. Software: Python.

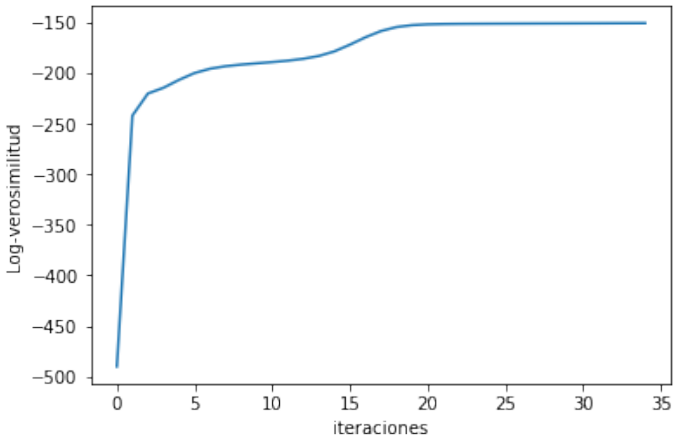

Figura 2.9: Log-verosimilitud de los datos completos. Software: Python.

interpreta como el monto de la i-ésima reclamación y $\sum_{i=1}^{N_t} X_i$ hace referencia a un proceso Poisson compuesto con parámetro λ.

Años más tarde, Reinhard [23] y Asmussen [2] presentan un modelo en el cual el proceso (2.42) es modulado por un proceso de saltos de Markov, $\boldsymbol{S} = \{S_t\}_{t\geq 0}$, con espacio de estados E finito, irreducible y con generador infinitesimal Λ, (ver detalles en el Apéndice). Dicho proceso es interpretado como las condiciones económicas que influyen en el comportamiento del capital de la compañía a lo largo del tiempo. El proceso subyacente incide en el modelo de riesgo del siguiente modo: si $S_t = i$, con $t \in [0, T]$, la frecuencia de las reclamaciones tendrá intensidad λ_i y los montos de reclamación en dicho estado seguirán una distribución G_i.

El uso del algoritmo EM en este modelo radica en, dado un horizonte de tiempo $[0, T]$, clasificar a las parejas (X_m, Y_m), según el estado que las modula, donde X_m es el monto de la m-ésima reclamación y Y_m el tiempo de interarribo entre la m-ésima reclamación y la anterior. Los datos observados del proceso son las parejas (X_m, Y_m), mientras que los datos no observados es la trayectoria del proceso a tiempo discreto $\{S_{W_i}\}$, donde $W_i = \sum_{j=1}^{i} Y_j$; es decir, el valor que toma el proceso en cada momento que hubo una reclamación.

Podemos suponer que las parejas (X_m, Y_m) se distribuyen de acuerdo a una mezcla de densidades dada por una combinación convexa como se explicó en (2.25),

$$f_{X_m,Y_m}(x,y) = \sum_{i=1}^{k} \omega_i f_i(x,y). \tag{2.43}$$

Para este ejemplo, supondremos que los montos de reclamación X_m se distribuyen Gamma con parámetros α_i, β_i, si $S_{W_m} = i$.

Dado que el estado subyacente se encuentra en el estado i, los montos y tiempos de interarribo de las reclamaciones son independientes, por lo cual la función de densidad de las parejas (X_m, Y_m), dado $S_t = i$ con $t = W_m$, se expresa como

$$f_i(x,y) = \lambda_i e^{-\lambda_i y} \frac{\beta_i^{\alpha_i} x^{\alpha_i - 1} e^{-\beta_i x}}{\Gamma(\alpha_i)}. \tag{2.44}$$

Dada la expresión para $p(S_{W_j} = i \,|\, x_j, y_j, \boldsymbol{\theta})$ en (2.29), donde $\boldsymbol{\theta} = (\omega_1, \omega_2, \lambda_1, \lambda_2, \alpha_1, \alpha_2, \beta_1, \beta_2)$ y $(X, Y, S) = (x_1, y_1, S_{W_1} ..., x_n, y_n, S_{W_n})$, para n reclamaciones ocurridas en el horizonte de tiempo $[0, T]$ el paso E definido en el Algoritmo 7 para este ejemplo queda como

$$E(\log \mathcal{L}(\boldsymbol{\theta} \,|\, X, Y, S)) = \sum_{j=1}^{n} \sum_{i=1}^{k} \log(\omega_i) p(s_j = i \,|\, x_j, y_j, \boldsymbol{\theta}) + \sum_{j=1}^{n} \sum_{i=1}^{k} \log(p(x_j, y_j \,|\, S_{W_j} = i, \boldsymbol{\theta})) p(S_{W_j} = i \,|\, x_j, y_j, \boldsymbol{\theta}). \tag{2.45}$$

De los cuatro vectores de parámetros a estimar por medio del paso M del algoritmo, se tienen las soluciones analíticas de las probabilidades ω_i, las tasas de intensidad λ_i, y el parámetro de escala β_i para i en $\{1, 2, ...k\}$, las cuales son:

$$\omega_i^{t+1} = \frac{1}{n} \sum_{j=1}^{n} p(S_{W_j} = i \,|\, x_j, y_j, \boldsymbol{\theta}^{(t)}), \tag{2.46}$$

$$\lambda_i^{t+1} = \frac{\sum_{j=1}^{n} p(S_{W_j} = i \,|\, x_j, y_j, \boldsymbol{\theta}^{(t)})}{\sum_{j=1}^{n} y_j p(S_{W_j} = i \,|\, x_j, y_j, \boldsymbol{\theta}^{(t)})} \tag{2.47}$$

y

$$\beta_i^{(t+1)} = \frac{\omega_i^{(t)} \sum_{j=1}^{n} p(S_{W_j} = i \,|\, _j, y_j, \boldsymbol{\theta}^{(t)})}{\sum_{j=1}^{n} x_j p(S_{W_j} = i \,|\, x_j, y_j, \boldsymbol{\theta}^{(t)})}. \tag{2.48}$$

Sin embargo, para el caso de los parámetros α_i, el paso M implica solucionar la ecuación

$$\sum_{j=1}^{n} \left[\log(\beta_i) + \log(x_j) - \psi(\alpha_i)\right] p(S_{W_j} = i \,|\, x_j, y_j, \boldsymbol{\theta}^{(t)}) = 0 \tag{2.49}$$

donde $\psi(\alpha_i) = \frac{\partial \log(\Gamma(\alpha_i))}{\partial \alpha_i}$, para el cual no se tiene solución cerrada.

Valores iniciales	Convergencia en iteración:	Valor a iteración t	Distancia a valor real
ω_0	t	$\omega^{(t)}$	$\mid \omega^{(t)} - \omega^* \mid$
(0.8,0.2)	5	(0.5016063,0.4983937)	(0.001606306,0.001606306)
(0.65,0.35)	10	(0.5016063,0.49839369)	(0.001606306,0.001606306)
(0.3,0.7)	10	(0.50160631,0.4983937)	(0.001606306,0.001606306)
α_0	t	$\alpha^{(t)}$	$\mid \alpha^{(t)} - \alpha^* \mid$
(3,30)	4	(10.20398,19.65945)	(0.2039795,0.3405497)
(12,35)	8	(10.20398,19.65945)	(0.2039795,0.3405497)
(1,2.5)	8	(10.20398,19.65945)	(0.2039795,0.3405497)
β_0	t	$\beta^{(t)}$	$\mid \beta^{(t)} - \beta^* \mid$
(4,1)	4	(6.095321,1.9739)	(0.09532137,0.02609977)
(2,2)	8	(6.0953214,1.9739)	(0.09532137,0.02609977)
(1,2.5)	8	(6.095321,1.9739002)	(0.09532137,0.02609977)
λ_0	t	$\lambda^{(t)}$	$\mid \lambda^{(t)} - \lambda^* \mid$
(1,4)	5	(0.2554094,2.05108)	(0.005409394,0.051080425)
(5,10)	10	(0.2554094,2.05108)	(0.005409394,0.051080425)
(2,3)	9	(0.25540939,2.0510804)	(0.005409394,0.051080425)

Tabla 2.9: Convergencia de parámetros

Implementación en R y resultados

Se presenta un ejemplo con dos estados moduladores y datos simulados que representarán los montos y tiempos de interarribo de reclamaciones. Usaremos parámetros que sean congruentes con lo que describe el modelo. Supondremos que el estado modulador 1 es más favorable que el estado 2, lo cual traducimos a que los montos de reclamación modulados por el estado 1 sean menores y menos frecuentes que aquellos modulados por el estado 2. Dicho esto, se proponen los siguientes valores:

- $\alpha_1 = 10$, $\alpha_2 = 20$
- $\beta_1 = 6$, $\beta_2 = 2$
- $\lambda_1 = 0.25$, $\lambda_2 = 2$
- $\omega_1 = 0.5$, $\omega_2 = 0.5$

Se hicieron 50 iteraciones y los parámetros inciales se muestran en la Tabla 2.9.

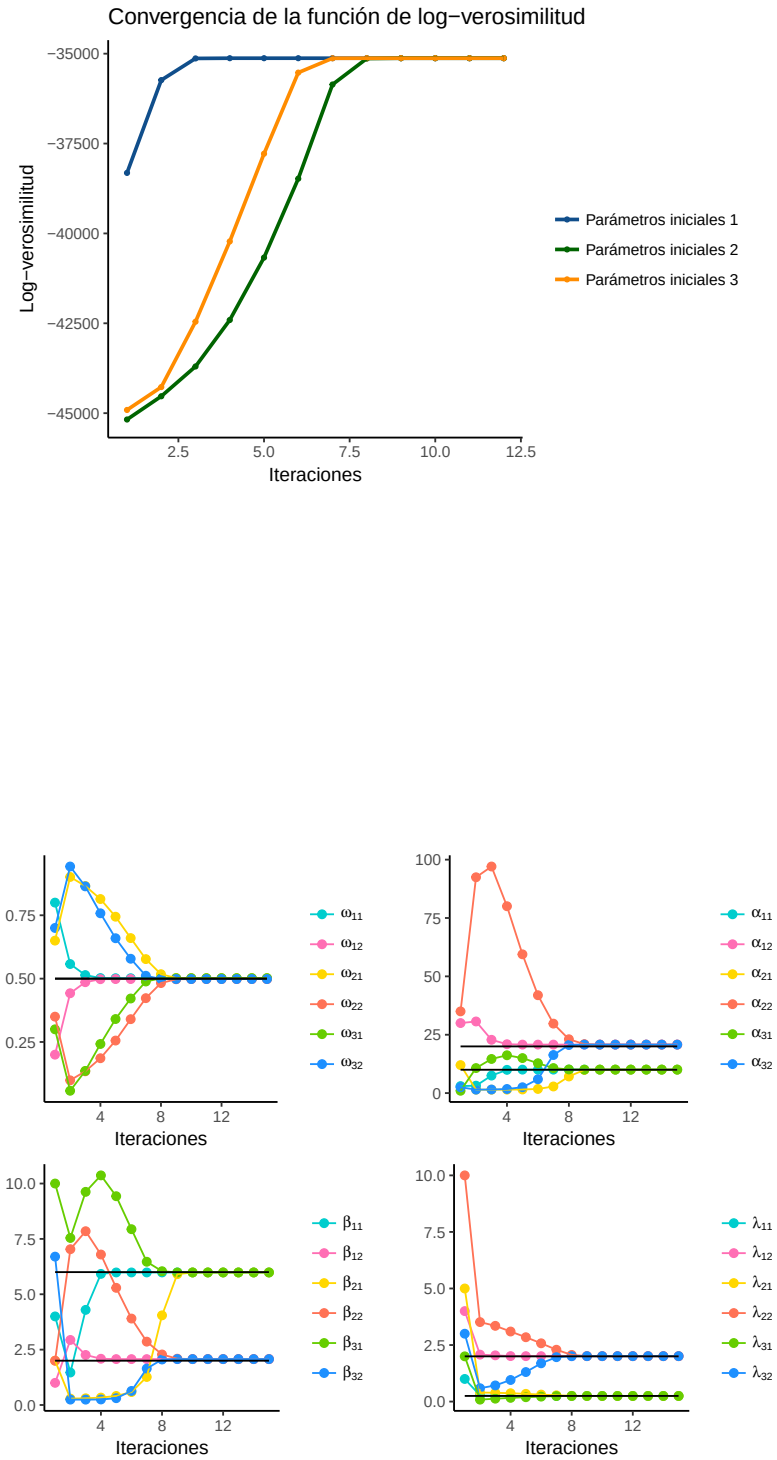

Figura 2.10: Convergencia de parámetros. Software: R

Capítulo 3

Teoría básica del algoritmo EM

El propósito de este capítulo es presentar los resultados más relevantes de la teoría básica que hay detrás del algoritmo EM. Comenzamos mostrando la monotonicidad no decreciente de la secuencia de valores de la función de log-verosimilitud, así como condiciones para asegurar la convergencia a valores máximos. También enunciamos el Teorema de Convergencia Global, el cual generaliza bajo ciertos supuestos los resultados de convergencia del algoritmo EM.

Asimismo, se presentan resultados respectivos a la tasa de convergencia del algoritmo EM y su relación con el principio de información faltante y con las matrices de información de los datos faltantes y los datos completos.

Fianlmente, se habla sobre el cálculo de errores estándar y se muestran algunos ejemplos respectivos a algunas de las implemetnaciones vistas en el capítulo anterior. Además se mencionan algunos métodos de aceleración de convergencia junto con un ejemplo.

3.1. Convergencia del algoritmo EM

Una de las propiedades más importantes de este algoritmo demostradas por Dempster, Laird y Rubin, es el comportamiento no decreciente de la función de verosimilitud después de una iteración, es decir:

$$L(\boldsymbol{\theta}^{(k+1)}) \geq L(\boldsymbol{\theta}^{(k)}) \tag{3.1}$$

para $k = 1, 2, ...$ Partimos de definir las funciones

$$k(\boldsymbol{X} \mid \boldsymbol{Y}; \boldsymbol{\theta}) = \frac{f(\boldsymbol{X}; \boldsymbol{\theta})}{g(\boldsymbol{Y}; \boldsymbol{\theta})} \tag{3.2}$$

como la función de densidad condicional de $\boldsymbol{X}$ dada la información incompleta $\boldsymbol{Y}$, y

$$H(\boldsymbol{\theta'} \mid \boldsymbol{\theta}) = \mathbb{E}\left[\log k(\boldsymbol{X} \mid \boldsymbol{Y}; \boldsymbol{\theta'}) \mid \boldsymbol{Y}; \boldsymbol{\theta}\right]. \tag{3.3}$$

Con esto, procedemos a demostrar el siguiente lema, necesario para demostrar la desigualdad en (3.1).

Lema 3.1.1. *Para cualesquiera* $(\theta', \theta) \in \boldsymbol{\Theta} \times \boldsymbol{\Theta}$*,*

$$H(\boldsymbol{\theta} \mid \boldsymbol{\theta}) \geq H(\boldsymbol{\theta'} \mid \boldsymbol{\theta}).$$

Demostración. Sean $(\boldsymbol{\theta'}, \boldsymbol{\theta}) \in \boldsymbol{\Theta} \times \boldsymbol{\Theta}$. Entonces,

$$\begin{aligned} H(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}) - H(\boldsymbol{\theta'} \,|\, \boldsymbol{\theta}) &= \mathbb{E}\left[\log k(\boldsymbol{X} \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) \,|\, \boldsymbol{Y}; \boldsymbol{\theta}\right] - \mathbb{E}\left[\log k(\boldsymbol{X} \,|\, \boldsymbol{Y}; \boldsymbol{\theta'}) \,|\, \boldsymbol{Y}; \boldsymbol{\theta}\right] \\ &= \int_{x \in \mathcal{X}(y)} \log k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) dx - \int_{x \in \mathcal{X}(y)} \log k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta'}) k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) dx \\ &= \int_{x \in \mathcal{X}(y)} k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) \log \frac{k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta})}{k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta'})} dx \\ &= -\int_{x \in \mathcal{X}(y)} k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) \log \frac{k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta'})}{k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta})} dx \\ &\geq -\log \int_{x \in \mathcal{X}(y)} k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) \frac{k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta'})}{k(x \,|\, \boldsymbol{Y}; \boldsymbol{\theta})} dx \\ &= 0. \end{aligned}$$

La desigualdad se da por la desigualdad Jensen y la concavidad de la función logaritmo. □

A la secuencia de iteraciones definida por el algoritmo $\boldsymbol{\theta}^{(0)} \to \boldsymbol{\theta}^{(1)} \to \boldsymbol{\theta}^{(2)}$... la podemos pensar como un mapeo $M : \Theta \to \Theta$ tal que

$$\boldsymbol{\theta}^{(k+1)} = M(\boldsymbol{\theta}^{(k)}),$$

para $k = 0, 1, 2, ...$

Definición 4. Un algoritmo iterativo con mapeo $M(\boldsymbol{\theta})$ es un algoritmo EM generalizado (GEM por sus siglas en inglés) si

$$Q(M(\boldsymbol{\theta}) \,|\, \boldsymbol{\theta}) \geq Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}).$$

Emplear un algoritmo EM generalizado puede resultar de gran utilidad, cuando la solución del paso M del algoritmo no es cerrada y el valor $\boldsymbol{\theta}$ tal que maximiza la función $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta^{(k)}})$ resulta difícil de obtener.

Teorema 3.1.1. *Para todo algoritmo EM generalizado (GEM),*

$$L(M(\boldsymbol{\theta})) \geq L(\boldsymbol{\theta}),$$

para todo $\boldsymbol{\theta}$ en Θ, cumpliéndose la iguladad si y sólo si

$$Q(M(\boldsymbol{\theta}) \,|\, \boldsymbol{\theta}) = Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta})$$

y

$$k(\boldsymbol{X} \,|\, \boldsymbol{Y}; M(\boldsymbol{\theta})) = k(\boldsymbol{X} \,|\, \boldsymbol{Y}; \boldsymbol{\theta}).$$

Demostración. Notemos que

$$\begin{aligned} L(\theta) &= \log g(\boldsymbol{Y}; \boldsymbol{\theta}) \\ &= \log f(\boldsymbol{X}; \boldsymbol{\theta}) - \log k(\boldsymbol{X} \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) \end{aligned}$$

cuya esperanza con respecto de la distribuciòn condicional de $\boldsymbol{X}$ dado $\boldsymbol{Y}$ con los parámetros $\boldsymbol{\theta}^{(k)}$ es

$$\begin{aligned} L(\boldsymbol{\theta}) &= \mathbb{E}\left[\log f(\boldsymbol{X}; \boldsymbol{\theta}) \,|\, \boldsymbol{Y}; \boldsymbol{\theta}^{(k)}\right] - \mathbb{E}\left[\log k(\boldsymbol{X} \,|\, \boldsymbol{Y}; \boldsymbol{\theta}) \,|\, \boldsymbol{Y}; \boldsymbol{\theta}^{(k)}\right] \\ &= Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{(k)}) - H(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{(k)}). \end{aligned}$$

Entonces,

$$L(\boldsymbol{\theta}^{(k+1)}) - L(\boldsymbol{\theta}^{(k)}) = Q(\boldsymbol{\theta}^{(k+1)} \,|\, \boldsymbol{\theta}^{(k)}) - H(\boldsymbol{\theta}^{(k+1)} \,|\, \boldsymbol{\theta}^{(k)}) - Q(\boldsymbol{\theta}^{(k)} \,|\, \boldsymbol{\theta}^{(k)}) + H(\boldsymbol{\theta}^{(k)} \,|\, \boldsymbol{\theta}^{(k)})$$
$$= \{Q(\boldsymbol{\theta}^{(k+1)} \,|\, \boldsymbol{\theta}^{(k)}) - Q(\boldsymbol{\theta}^{(k)} \,|\, \boldsymbol{\theta}^{(k)})\} - \{H(\boldsymbol{\theta}^{(k+1)} \,|\, \boldsymbol{\theta}^{(k)}) - H(\boldsymbol{\theta}^{(k)} \,|\, \boldsymbol{\theta}^{(k)})\}.$$

Claramente la primera diferencia del lado derecho de la igualdad es no negativa, por definición del paso M del Algoritmo. Por el Lema 3.1.1, la segunda diferencia es no positiva y por lo tanto, la log-verosimilitud es no decreciente y claramente se da la igualdad si y sólo si $Q(M(\boldsymbol{\theta}) \,|\, \boldsymbol{\theta}) = Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta})$ y $k(\boldsymbol{X} \,|\, \boldsymbol{Y}; M(\boldsymbol{\theta})) = k(\boldsymbol{X} \,|\, \boldsymbol{Y}; \boldsymbol{\theta})$.

□

Corolario 3.1.1. *Sea $\boldsymbol{\theta}^* \in \Theta$ tal que $L(\boldsymbol{\theta}^*) \geq L(\boldsymbol{\theta})$ para todo $\boldsymbol{\theta} \in \Theta$. Entonces para todo algoritmo EM generalizado,*

(a) $L(M(\boldsymbol{\theta}^*)) = L(\boldsymbol{\theta}^*)$

(b) $Q(M(\boldsymbol{\theta}^*) \,|\, \boldsymbol{\theta}^*) = Q(\boldsymbol{\theta}^* \,|\, \boldsymbol{\theta}^*)$

(c) $k(\boldsymbol{x} \,|\, \boldsymbol{y}; M(\boldsymbol{\theta}^*)) = k(\boldsymbol{x} \,|\, \boldsymbol{y}; \boldsymbol{\theta}^*)$.

Demostración. Del Teorema 3.1.1 junto con los supuestos tenemos lo siguiente:

(I) $L(\boldsymbol{\theta}^*) \geq L(\boldsymbol{\theta})$, para todo $\boldsymbol{\theta} \in \Theta$

(II) $L(M(\boldsymbol{\theta}^*)) \geq L(\boldsymbol{\theta})$, para todo $\boldsymbol{\theta} \in \Theta$,

lo cual implica que tanto $L(M(\boldsymbol{\theta}^*)) \geq L(\boldsymbol{\theta}^*)$ como $L(M(\boldsymbol{\theta}^*)) \leq L(\boldsymbol{\theta}^*)$ se cumpla. Por lo tanto, $L(M(\boldsymbol{\theta}^*)) = L(\boldsymbol{\theta}^*)$. Además del Teorema 3.1.1 también tenemos $Q(M(\boldsymbol{\theta}^*) \,|\, \boldsymbol{\theta}^*) = Q(\boldsymbol{\theta}^* \,|\, \boldsymbol{\theta}^*)$ y $k(\boldsymbol{x} \,|\, \boldsymbol{y}; M(\boldsymbol{\theta}^*)) = k(\boldsymbol{x} \,|\, \boldsymbol{y}; \boldsymbol{\theta}^*)$.

□

Corolario 3.1.2. *Sea $\boldsymbol{\theta}^* \in \Theta$ tal que $L(\boldsymbol{\theta}^*) > L(\boldsymbol{\theta})$ para todo $\boldsymbol{\theta} \in \Theta$ tal que $\boldsymbol{\theta}^* \neq \boldsymbol{\theta}$, entonces para todo algoritmo EM generalizado,*

$$M(\boldsymbol{\theta}^*) = \boldsymbol{\theta}^*.$$

Demostración. Supongamos que $M(\boldsymbol{\theta}^*) \neq \boldsymbol{\theta}^*$. Del Corolario 3.1.1 tenemos que

$$L(M(\boldsymbol{\theta}^*)) = L(\boldsymbol{\theta}^*), \tag{3.4}$$

y de la hipótesis tenemos

$$L(\boldsymbol{\theta}^*) > L(\boldsymbol{\theta}), \tag{3.5}$$

para todo $\boldsymbol{\theta} \in \Theta$ tal que $\boldsymbol{\theta}^* \neq \boldsymbol{\theta}$. Pero supusimos que $M(\boldsymbol{\theta}^*) \neq \boldsymbol{\theta}^*$ lo cual contradice que tanto (3.4) como (3.5) se cumplan. Por lo tanto, $M(\boldsymbol{\theta}^*) = \boldsymbol{\theta}^*$.

□

Los Corolarios 3.1.1 y 3.1.2 nos dicen que si el algoritmo EM llega a un maximizador global, la log-verosimilitud ya no cambia en las siguientes iteraciones. También nos muestran que si contamos un único maximizador global, el algoritmo tiene un punto fijo.

Los Teoremas 2 y 3 de DLR exponen condiciones para que la secuencia de valores $\boldsymbol{\theta}^{(p)}$ producidos por el algoritmo EM pueda converger. Sin embargo, la demostración es incorrecta en unos pasos (Boyles [7] muestra un contraejemplo de ello), por lo que estos teoremas no son válidos.

Boyles [7] presenta dos teoremas importantes, uno siendo una versión corregida del Teorema 2 de DLR y el segundo concerniente a la convergencia de los parámetros a puntos estacionarios.

Teorema 3.1.2. *Supongamos que:*

1. $\|\boldsymbol{\theta}^{(p+1)} - \boldsymbol{\theta}^{(p)}\| \to 0$ *cuando* $p \to \infty$.
2. *El conjunto* $\{L \geq \lambda_0\} = \{\boldsymbol{\theta} \in \Theta : L(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}^{(0)})\}$, *con* $\lambda_0 = L(\boldsymbol{\theta}^{(0)})$, *es compacto.*
3. L *es continua en* $\{L \geq \lambda_0\}$.

Entonces existe $\lambda^* \in [\lambda_0, \infty)$ *tal que* $L(\boldsymbol{\theta}^{(p)}) \to \lambda^*$ *cuando* $p \to \infty$. *Además,* $\{\boldsymbol{\theta}^{(p)}\}$ *converge a algún subconjunto compacto y conexo de* $\{L = \lambda^*\}$.

Teorema 3.1.3. *Supongamos que:*

1. $\|\boldsymbol{\theta}^{(p+1)} - \boldsymbol{\theta}^{(p)}\| \to 0$ *cuando* $p \to \infty$.
2. *El conjunto* $\{L \geq \lambda_0\}$ *es compacto en el interior de* Θ.
3. *Las funciones* L *y* $Q(\cdot \,|\, \boldsymbol{\theta})$ *con* $\boldsymbol{\theta} \in \{L \geq \lambda_0\}$ *son diferenciables en un conjunto abierto que contiene a* $\{L \geq \lambda_0\}$.
4. M *es continua en* $\{L \geq \lambda_0\}$.
5. $\boldsymbol{D^{10}} Q(M(\boldsymbol{\theta}) \,|\, \boldsymbol{\theta}) = 0$, *para todo* $\boldsymbol{\theta} \in \{L \geq \lambda_0\}$.

   Entonces $\{\boldsymbol{\theta}\}$ *converge a un subconjunto compacto y conexo de* $S \cap \{L = \lambda^*\}$, *donde* λ^* *es el valor dado en el Teorema 3.1.2 y* $S = \{\boldsymbol{\theta} \in \Theta^0 : DL(\boldsymbol{\theta}) = 0\}$.

3.1.1. Teorema de Convergencia Global

Definición 5. Sea $X \subseteq \mathbb{R}^n$. Una función A que manda puntos de X a subconjuntos de X se dice que es cerrada en $\boldsymbol{x}$ si para toda sucesión $\{\boldsymbol{x}_n\}$ tal que $\boldsymbol{x}_n \to \boldsymbol{x}, \quad \boldsymbol{y}_n \to \boldsymbol{y}$, con $\boldsymbol{y}_n \in A(\boldsymbol{x}_n)$ implica que $\boldsymbol{y} \in A(\boldsymbol{x})$.

Teorema 3.1.4. *Sea* $\{\boldsymbol{x}_k\}_{k=0}^{\infty}$ *la sucesión generada por* $\boldsymbol{x}_{k=1} \in M(\boldsymbol{x}_k)$, *donde* M *es una función que manda puntos de* X *a subconjuntos de* X. *Sea* $\Gamma \subset X$ *un conjunto de soluciones dado, y supongamos que:*

1. *todos los puntos* $\boldsymbol{x}_k$ *están contenidos en un conjunto compacto* $S \subset X$;
2. M *es cerrada en el complemento de* Γ;
3. *Existe una función continua* α *en* X *tal que:*

   a) *si* $\boldsymbol{x} \notin \Gamma, \quad \alpha(\boldsymbol{y}) > \alpha(\boldsymbol{x})$ *para toda* $\boldsymbol{y} \in M(\boldsymbol{x})$

   b) *si* $\boldsymbol{x} \in \Gamma, \quad \alpha(\boldsymbol{y}) \geq \alpha(\boldsymbol{x})$ *para toda* $\boldsymbol{y} \in M(\boldsymbol{x})$.

Entonces todos los puntos límite de $\{\boldsymbol{x}_k\}$ *están en el conjunto de soluciones* Γ *y* $\alpha(\boldsymbol{x}_k)$ *converge monótonamente a* $\alpha(\boldsymbol{x})$ *para alguna* $\boldsymbol{x} \in \Gamma$.

Demostración. Supongamos que $\boldsymbol{x}^*$ es un punto límite de la sucesión $\{\boldsymbol{x}_k\}_{k=0}^{\infty}$. Entonces, por ser S compacto existe una subsucesión $\{\boldsymbol{x}_{k_i}\}_{i=0}^{\infty}$ que converge a $\boldsymbol{x}^*$. Ahora, como α es continua, $\alpha(\boldsymbol{x}_{k_i}) \to \alpha(\boldsymbol{x}^*)$, cuando $i \to \infty$. Es decir, para toda $\epsilon > 0$, existe una i_ϵ tal que para $i \geq i_\epsilon$

$$\alpha(\boldsymbol{x}^*) - \alpha(\boldsymbol{x}_{k_i}) < \epsilon. \tag{3.6}$$

Además, por la condición 3, tenemos que α es monótonamente no decreciente en $\{\boldsymbol{x}_k\}_{k=0}^{\infty}$, entonces para cualquier k,

$$\alpha(\boldsymbol{x}_k) \leq \alpha(\boldsymbol{x}^*) \tag{3.7}$$

y también, para $j \geq k_{i_\epsilon}$

$$\alpha(\boldsymbol{x}_j) \geq \alpha(\boldsymbol{x}_{k_{i_\epsilon}}) \tag{3.8}$$

Con las ecuaciones anteriores, llegamos a que

$$\mid \alpha(\boldsymbol{x}_k) - \alpha(\boldsymbol{x}^*) \mid < \epsilon \tag{3.9}$$

para toda $k \geq k_{i_\epsilon}$.

Consideremos ahora la subsucesión $\{\boldsymbol{x}_{k_i+1}\}_{i=0}^{\infty}$ tal que $\boldsymbol{x}_{k_i+1}$ Como S es compacto, existe un subsucesión de $\{\boldsymbol{x}_{k_i+1}\}_{i=0}^{\infty}$ que converge a algún punto $\boldsymbol{x}^{**} \in S$. Análogo a lo que se hizo anteriormente, se llega a que

$$\lim_{k\to\infty} \alpha(\boldsymbol{x}_k) = \alpha(\boldsymbol{x}^{**}). \tag{3.10}$$

Por (3.9) y (3.10) tenemos

$$\alpha(\boldsymbol{x}^{**}) = \alpha(\boldsymbol{x}^*) \tag{3.11}$$

y por tanto la sucesión original $\{\boldsymbol{x}_k\}_{k=0}^{\infty}$ converge a $\boldsymbol{x}^*$. Falta demostrar que $\boldsymbol{x}^*$ es solución. Supongamos que no lo es. De las subsucesiones convergentes definidas anteriormente tenemos lo siguiente:

1. $\boldsymbol{x}_{k_i+1} \to \boldsymbol{x}^{**}$, para $i \in \mathcal{I} \subset \mathbb{N}$
2. $\boldsymbol{x}_{k_i} \to \boldsymbol{x}^*$, para $i \in \mathcal{N}$
3. $\boldsymbol{x}_{k_i+1} \in M(\boldsymbol{x}_{k_i})$.

Entonces como M es cerrada en Γ^c y $\boldsymbol{x}^* \notin \Gamma$ se llega a que

$$\boldsymbol{x}^{**} \in M(\boldsymbol{x}^*). \tag{3.12}$$

Además, por 3a) tenemos que

$$\alpha(\boldsymbol{x}^{**}) > \alpha(\boldsymbol{x}^*). \tag{3.13}$$

Sin embargo, por (3.11) tenemos que $\alpha(\boldsymbol{x}^{**}) = \alpha(\boldsymbol{x}^*)$ por lo que llegamos a una contradicción. Por lo tanto, $\boldsymbol{x}^*$ debe ser solución. □

Wu [31] hace uso de los siguientes supuestos, los cuales consideraremos válidos a lo largo del resto de esta sección:

Supuesto 1. Θ *es un subconjunto de* $\mathbb{R}^n$,

Supuesto 2. $\Theta_{\theta_0} = \{\theta \in \Theta : L(\theta) \geq L(\theta_0)\}$ *es compacto para todo* $L(\theta_0) > -\infty$,

Supuesto 3. L *es continua en* Θ *y diferenciable en el interior de* Θ,

Supuesto 4. $\{L(\theta)_p\}_{p\geq 0}$ *es acotada por arriba para todo* $\theta_0 \in \Theta$,

Supuesto 5. Θ_{θ_0} *está en el interior de* Θ *para* $\theta_0 \in \Theta$.

Un caso especial del Teorema 3.1.4 sería definir a la función M como el mapeo generado por las iteraciones del algoritmo generalizado, explicado en la Definición 4; a α como la función de log-verosimilitud L y al conjunto de soluciones Γ como el conjunto de puntos estacionarios $\mathcal{S}$. Formalmente, el teorema diría lo siguiente:

Teorema 3.1.5. *Sea* $\{\theta_k\}_{k=0}^{\infty}$ *la sucesión del algoritmo GEM, generada por* $\theta_{k+1} \in M(\theta_k)$, *donde* M *es una función que manda puntos de* X *a subconjuntos de* X. *Supongamos que:*

1. M *es cerrada en* $\mathcal{S}^c$.

2. $L(\theta_{k+1}) > L(\theta_k)$ *para todo* $\theta_k \notin \mathcal{S}$.

Entonces todos los puntos límite de $\{\theta_k\}_{k=0}^{\infty}$ *son puntos estacionarios de* L, *y* $L(\theta_k)$ *converge monótonamente a* $L^* = L(\theta^*)$, *para algún* $\theta^* \in \mathcal{S}$.

La condición 1 del Teorema 3.1.4 no se enuncia en el Teorema 3.1.5, puesto que $\{L(\theta_k)\}_{k\geq 0}$ es no decreciente y por el supuesto 4 es acotada, dado cualquier valor inicial θ_0. Tampoco se enuncia la condición $3b$ del Teorema 3.1.4, ya que para toda $k = 1, 2, ...$ se cumple que $L(\theta^{(k+1)}) \geq L(\theta^{(k)})$.

En el caso particular del algoritmo EM, la condición 2 del Teorema 3.1.5 siempre se cumple. Consideremos a $\theta^{(k)}$ tal que no es punto estacionario, entonces

$$\left.\frac{\partial L(\theta)}{\partial \theta}\right|_{\theta=\theta^{(k)}} = \left.\frac{\partial Q(\theta \,|\, \theta^{(k)})}{\partial \theta}\right|_{\theta=\theta^{(k)}} - \left.\frac{\partial H(\theta \,|\, \theta^{(k)})}{\partial \theta}\right|_{\theta=\theta^{(k)}}. \tag{3.14}$$

Por el Lema 3.1.1, sabemos que $\theta = \theta^{(k)}$ maximiza a $H(\theta \,|\, \theta^{(k)})$ y por tanto

$$\begin{aligned}\left.\frac{\partial L(\theta)}{\partial \theta}\right|_{\theta=\theta^{(k)}} &= \left.\frac{\partial Q(\theta \,|\, \theta^{(k)})}{\partial \theta}\right|_{\theta=\theta^{(k)}} \\ &\neq 0,\end{aligned} \tag{3.15}$$

pues $\theta^{(k)}$ no es punto estacionario y por la definición del paso M del algoritmo, se cumple

$$Q(\theta^{(k+1)}) > Q(\theta(k)), \tag{3.16}$$

lo cual implica que $L(\theta(k+1)) > L(\theta(k))$.

Una condición necesaria para que se cumpla el supuesto 1 del Teorema 3.1.5 para el caso del algoritmo EM, es que $Q(\theta \,|\, \theta^{(k)})$ sea continua en ambos argumentos, con lo cual se formula el siguiente teorema.

Teorema 3.1.6. *Supongamos que la función* $Q(\omega \,|\, \theta)$ *es continua en* ω *y* θ. *Entonces todos los puntos límite de* $\{\theta_k\}_{k=0}^{\infty}$ *son puntos estacionarios de* L, *y* $L(\theta_k)$ *converge monótonamente a* $L^* = L(\theta^*)$, *para algún* $\theta^* \in \mathcal{S}$.

Demostración. Veamos que se cumple el primer supuesto del Teorema 3.1.5 dada la hipótesis que $Q(\theta \,|\, \theta^k)$ es continua en ambos argumentos. Primero, probaremos que continuidad de $Q(\theta \,|\, \theta^k)$ en θ^k implica la cerradura de Q. Sea $Q(\theta \,|\, \theta^k)$ continua en θ^k y supongamos que no es cerrada. Es decir, debe existir una sucesión $\{\theta_n^{(k)}\}_{n\geq 0}$ tal que $\theta_n^k \to \theta^{k*}$, $Q(\theta \,|\, \theta_n^k) \to Q(\theta \,|\, \theta_n^{k**})$, pero $Q(\theta \,|\, \theta_n^{k**}) \neq Q(\theta \,|\, \theta_n^{k*})$.

Sea $\epsilon > 0$, por continuidad de $Q(\theta \,|\, \theta^k)$ en θ_k, existe $N \in \mathbb{N}$ tal que para $n \geq N$, $\mid Q(\theta \,|\, \theta_n^k) - Q(\theta \,|\, \theta^{k*}) \mid < \epsilon/2$. Por otro lado sabemos que $\mid Q(\theta \,|\, \theta_n^k) - Q(\theta \,|\, \theta^{k**}) \mid < \epsilon/2$, entonces

$$\begin{aligned}\mid Q(\theta \,|\, \theta_n^{k**}) - Q(\theta \,|\, \theta^{k*}) \mid &= \mid Q(\theta \,|\, \theta_n^{k**}) - Q(\theta \,|\, \theta_n^k) + Q(\theta \,|\, \theta_n^k) - Q(\theta \,|\, \theta^{k*}) \mid \\ &\leq \mid Q(\theta \,|\, \theta^{k**}) - Q(\theta \,|\, \theta_n^k) \mid + \mid Q(\theta \,|\, \theta_n^k) - Q(\theta \,|\, \theta^{k**}) \mid \\ &< \epsilon/2 + \epsilon/2 = \epsilon.\end{aligned}$$

Sin embargo, supusimos que $Q(\theta \,|\, \theta^k)$ no era cerrada y $Q(\theta \,|\, \theta_n^{k**}) \neq Q(\theta \,|\, \theta_n^{k*})$, por lo cual llegamos a una contradicción. Por lo tanto, $Q(\theta \,|\, \theta^k)$ debe ser cerrada.

Ahora, probemos que la continuidad de $Q(\theta \mid \theta^k)$ en θ implica la cerradura de M en $\mathcal{S}^c$. Supongamos que M no es cerrada en $\mathcal{S}^c$. Entonces, existe una sucesión $\{\theta_k\}_{k\geq 0}$ en $\mathcal{S}^c$, tal que converge a un punto $\theta^* \in \mathcal{S}^c$, y $M(\theta^k) \to \theta^{**}$, pero $\theta^{**} \neq M(\theta^*)$, con

$$M(\theta) = \arg\max_{\theta \in \mathcal{S}^c} Q(\theta \mid \theta^k). \tag{3.17}$$

□

Sin embargo, la condición 2 del Teorema 3.1.5 para el caso de maximizadores locales no se cumple. Basta tomar $\theta^{(k)} \in \mathcal{S} \setminus \mathcal{M}$, con lo cual

$$\begin{aligned} \left.\frac{\partial L(\theta)}{\partial \theta}\right|_{\theta=\theta^{(k)}} &= \left.\frac{\partial Q(\theta \mid \theta^{(k)})}{\partial \theta}\right|_{\theta=\theta^{(k)}} \\ &= 0. \end{aligned} \tag{3.18}$$

Es decir, la función $Q(\theta \mid \theta^{(k)})$ se maximiza en $\theta^{(k)}$ y el algoritmo termina, rompiendo la condición 2 de $L(\theta^{(k+1)}) > L(\theta^{(k)})$, convergiendo a un punto silla y no a un maximizador local, el teorema sugiere.

Por lo tanto, Wu impone la condición adicional

$$\sup_{\theta' \in \Theta} Q(\theta' \mid \theta) > Q(\theta \mid \theta), \tag{3.19}$$

para toda $\theta \in \mathcal{S} \setminus \mathcal{M}$. Con esta condición, aseguramos que ningún punto silla maximice a la función $Q(\theta \mid \theta^{(k)})$ y por tanto, incremente la log-verosimilitud hasta converger a un maximizador local. Dicho eso, tenemos el siguiente teorema.

Teorema 3.1.7. *Supongamos que la función $Q(\omega \mid \theta)$ es continua en ω y θ y que la siguiente condición se cumple*

$$\sup_{\theta' \in \Theta} Q(\theta' \mid \theta) > Q(\theta \mid \theta). \tag{3.20}$$

Entonces todos los puntos límite de $\{\theta_k\}_{k=0}^{\infty}$ son maximizadores locales de L, y $L(\theta_k)$ converge monótonamente a $L^* = L(\theta^*)$, para algún $\theta^* \in \mathcal{M}$.

Refiérase a [31].

3.1.2. Convergencia de la secuencia de parámetros

Definimos los conjuntos

$$\mathcal{S}(L^*) = \{\theta \in \mathcal{S} : L(\theta) = L^*\} \tag{3.21}$$

$$\mathcal{M}(L^*) = \{\theta \in \mathcal{M} : L(\theta) = L^*\}. \tag{3.22}$$

Teorema 3.1.8. *Sea $\{\theta_k\}_{k=0}^{\infty}$ una sucesión generada por el algoritmo GEM que cumple las condiciones del Teorema 3.1.5. Si $\mathcal{S}(L^*) = \{\theta^*\}$ (o $\mathcal{M}(L^*)$), con $L(\theta_k) \to L^*$, entonces $\theta_k \to \theta^*$.*

Demostración. Por el Teorema 3.1.5 tenemos que todos los puntos límite de $\{\theta_k\}_{k=0}^{\infty}$ están en $\mathcal{S}$ (o en $\mathcal{M}$) y además, $L(\theta_k)$ converge a $L^* = L(\theta^*)$, para algún θ^* en $\mathcal{S}(L^*)$. Entonces es claro que si $\mathcal{S}(L^*) = \{\theta^*\}$ (o $\mathcal{M}$) entonces $\{\theta_k\}_{k=0}^{\infty}$ converge a ese único punto límite. □

Una alternativa del Teorema 3.1.8 considera una condición más relajada que el supuesto de $\mathcal{S}(L^*) = \{\theta^*\}$ y necesaria para asegurar el resultado $\theta_k \to \theta^*$.

Teorema 3.1.9. *Sea $\{\theta_k\}_{k=0}^{\infty}$ una sucesión generada por el algoritmo GEM tal que cumple las condiciones del Teorema 3.1.5. Si $\|\theta_{k+1} - \theta_k\| \to 0$ cuando $k \to \infty$, entonces todos los puntos límite de $\{\theta_k\}$ están contenidos en un subconjunto compacto y conexo de $\mathcal{S}(L^*)$, con $L(\theta_k) \to L^*$. En particular, si $\mathcal{S}(L^*)$ es discreto, entonces $\theta_k \to \theta^*$, para algún $\theta^* \in \mathcal{S}(L^*)$.*

Su demostración se sigue del Teorema 28.1 de Ostrowski [22], pues por la suposición 2, la sucesión $\{\theta_k\}_{k=0}^{\infty}$ es acotada y por hipótesis, cumple que $\|\theta_{k+1} - \theta_k\| \to 0$ cuando $k \to \infty$, entonces por [22] , los puntos límites de la sucesión convergen a un conjunto conexo y compacto. Por otro lado, por el Teorema 3.1.5, los puntos límites están en $\mathcal{S}(L^*)$. El Teorema 3.1.9 también es válido para el conjunto $\mathcal{M}(L^*)$.

3.2. Tasa de convergencia

Un punto muy importante a tratar es analizar la tasa de convergencia del algoritmo, puesto que de este modo podemos medir su desempeño y eficiencia.

Primero, introduzcamos notación, términos y resultados que usaremos a lo largo de esta sección. Denotamos como

$$\boldsymbol{S}_{obs}(\theta) = \frac{\partial}{\partial\theta} l(\theta; y) \tag{3.23}$$

$$\boldsymbol{S}_{c}(\theta) = \frac{\partial}{\partial\theta} l_c(\theta; x) \tag{3.24}$$

al "score" de los datos observados y al de los datos completos, respectivamente, donde $l(\theta; y)$ hace referencia a la log-verosimilitud de los datos observados y $l_c(\theta; x)$ es la log-versimilitud con respecto de los datos completos.

Suponiendo válido el intercambio de la derivada y la integral, podemos expresar a $\boldsymbol{S}_{obs}$ de la siguiente manera:

$$\begin{aligned}
\boldsymbol{S}_{obs}(\theta) &= \frac{\partial}{\partial\theta} l(\theta; y) \\
&= \frac{\partial}{\partial\theta} \log(g(y;\theta)) \\
&= \frac{\frac{\partial}{\partial\theta} g(y;\theta)}{g(y;\theta)} \\
&= \frac{\frac{\partial}{\partial\theta} \int_{x\in\mathcal{X}} f(x;\theta)dx}{g(y;\theta)} \\
&= \frac{\int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x;\theta)dx}{g(y;\theta)} \\
&= \int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} \log(f(x;\theta)) k(x \,|\, y;\theta)dx \\
&= \mathbb{E}\left[\frac{\partial}{\partial\theta} \log(f(x;\theta)) \,|\, y \right] \\
&= \mathbb{E}\left[\boldsymbol{S}_c(\theta) \,|\, y \right]
\end{aligned} \tag{3.25}$$

con $k(x \,|\, y;\theta) = \frac{f(x;\theta)}{g(y;\theta)}$, la densidad condicional de $\boldsymbol{X} \mid \boldsymbol{Y}$, con $\mathcal{X} = \{x : y(x) = y\}$, $g(y;\theta)$ la función de densidad correspondiente al vector de datos observados y $f(x;\theta)$ la función de

densidad de los datos completos.

Por otro lado, denotemos como I_{obs} y I_c a la información de Fisher de los datos observados y a la de los datos completos, respectivamente; es decir

$$I_{obs}(\theta) = -\frac{\partial^2}{\partial\theta} l(\theta; y) \tag{3.26}$$

$$I_c(\theta) = -\frac{\partial^2}{\partial\theta} l_c(\theta; x). \tag{3.27}$$

Notemos que

$$l(\theta; y) = l_c(\theta; x) - \log(k(x \,|\, y; \theta)).$$

Entonces,

$$I_{obs}(\theta) = I_c(\theta) + \frac{\partial^2}{\partial\theta} \log(k(x \,|\, y; \theta)).$$

Tomando esperanza sobre la distribucióin condicional $k(x \,|\, y; \theta)$ obtenemos la siguiente expresión

$$I_{obs}(\theta) = \mathcal{I}_c(\theta \,|\, y) - \mathcal{I}_m(\theta \,|\, y),$$

donde $\mathcal{I}_c(\theta \,|\, y) = -\mathbb{E}\left[\frac{\partial^2}{\partial\theta} l(\theta; x) \,|\, y\right]$ y $\mathcal{I}_m(\theta \,|\, y) = -\mathbb{E}\left[\frac{\partial^2}{\partial\theta} \log(k(x \,|\, y; \theta)) \,|\, y\right]$. Es decir, a la información de los datos observados la expresamos en términos de la información esperada de los datos completos menos la información de los datos faltantes, embas condicionadas al valor de los datos observados. A esto es a lo que Orchard y Woodbury [21] llamaron principio de información faltante.

Podemos expresar a $\mathcal{I}_m(\theta \,|\, y)$ en términos de $\boldsymbol{S}_c(\theta)$, notando primero que

$$\begin{aligned}
I_{obs}(\theta) &= -\frac{\partial}{\partial\theta}\left(\frac{\partial}{\partial\theta} l(\theta; y)\right) \\
&= -\frac{\partial}{\partial\theta} \frac{\int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x; \theta) dx}{g(y; \theta)} \\
&= \frac{-\int_{x\in\mathcal{X}} \frac{\partial^2}{\partial\theta} f(x; \theta) g(y; \theta) dx + \int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x; \theta) \int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x; \theta)}{g(y; \theta)^2} \\
&= \frac{-\int_{x\in\mathcal{X}} \frac{\partial^2}{\partial\theta} f(x; \theta) g(y; \theta) dx + \int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x; \theta) dx \int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x; \theta) dx}{g(y; \theta)^2} \\
&= -\int_{x\in\mathcal{X}} \frac{\partial^2}{\partial\theta} f(x; \theta)/g(y; \theta) dx + \int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x; \theta)/g(y; \theta) dx \int_{x\in\mathcal{X}} \frac{\partial}{\partial\theta} f(x; \theta)/g(y; \theta) dx \\
&= -\int_{x\in\mathcal{X}} \frac{\partial^2}{\partial\theta} f(x; \theta)/g(y; \theta) dx + \mathbb{E}\left[\boldsymbol{S}_c(\theta) \,|\, y\right] \mathbb{E}\left[\boldsymbol{S}_c(\theta) \,|\, y\right].
\end{aligned} \tag{3.28}$$

Ahora, notemos que

$$
\begin{aligned}
-\int_{x\in\mathcal{X}} \frac{\partial^2}{\partial\theta} f(x;\theta)/g(y;\theta)dx &= -\int_{x\in\mathcal{X}} \frac{\partial^2}{\partial\theta} \log(f(x;\theta)) f(x;\theta)/g(y;\theta)dx \\
&\quad -\int_{x\in\mathcal{X}} \frac{\frac{\partial}{\partial\theta} f(x;\theta)}{f(x;\theta)} \frac{\frac{\partial}{\partial\theta} f(x;\theta)}{f(x;\theta)} \frac{f(x;\theta)}{g(y;\theta)} dx \\
&= \mathcal{I}_c(\theta \,|\, y) - \mathbb{E}\left[\boldsymbol{S}_c(\theta)\boldsymbol{S}_c(\theta) \,|\, y\right].
\end{aligned}
$$

Por lo tanto,

$$
\begin{aligned}
I_{obs}(\theta) &= \mathcal{I}_c(\theta \,|\, y) - \mathbb{E}\left[\boldsymbol{S}_c(\theta)\boldsymbol{S}_c(\theta) \,|\, y\right] + \mathbb{E}\left[\boldsymbol{S}_c(\theta) \,|\, y\right] \mathbb{E}\left[\boldsymbol{S}_c(\theta) \,|\, y\right] \\
&= \mathcal{I}_c(\theta \,|\, y) - \mathrm{Var}\left(\boldsymbol{S}_c(\theta) \,|\, y\right). \qquad (3.29)
\end{aligned}
$$

El algoritmo EM converge aproximadamente linealmente, puesto que, suponiendo que la sucesión generada por el algoritmo converge a un punto fijo $\boldsymbol{\theta}^*$ y que la función $M(\boldsymbol{\theta})$ es diferenciable, haciendo la expansión de Taylor de la función $M(\boldsymbol{\theta}^{(t)})$ alrededor de $\boldsymbol{\theta}^*$ es

$$
M(\boldsymbol{\theta}^{(t)}) \approx M(\boldsymbol{\theta}^*) + J(\boldsymbol{\theta}^*)(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^*) \qquad (3.30)
$$

donde $J(\boldsymbol{\theta}) = \left(\frac{\partial M_j(\boldsymbol{\theta})}{\partial\theta_i}\right)_{i,j\in\{1,\ldots,d\}}$. Por lo tanto,

$$
\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^* \approx J(\boldsymbol{\theta}^*)(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^*). \qquad (3.31)
$$

Por lo tanto, la tasa de convergencia estará dada por el radio espectral de la matriz $J(\boldsymbol{\theta})$, por lo cual a ésta la llamaremos matriz de convergencia. Si su radio espectral es menor que 1 entonces podemos asegurar que el método iterativo del EM es convergente.

De nuevo, por medio de una expansión de Taylor de $\boldsymbol{S}_{obs}(\theta)$ alrededor de $\boldsymbol{\theta}^{(t)}$ nos da

$$
\boldsymbol{S}_{obs}(\theta) = \boldsymbol{S}_{obs}(\theta^{(t)}) - I_{obs}(\theta^{(t)})(\theta - \theta^{(t)}), \qquad (3.32)
$$

evaluando en $\theta = \theta^*$ y notando que $\boldsymbol{S}_{obs}(\theta^*) = 0$ nos queda

$$
\theta^* = \theta^{(t)} + I_{obs}^{-1}(\theta^{(t)})\boldsymbol{S}_{obs}(\theta^{(t)}). \qquad (3.33)
$$

Análogo a lo que se hizo en (3.30), hacemos la expansión de Taylor de la función $\frac{\partial Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})}{\partial\boldsymbol{\theta}}$ alrededor de $\boldsymbol{\theta}^{(t)}$ y evaluada en $\boldsymbol{\theta}^{(t+1)}$, cumpliéndose que

$$
\begin{aligned}
0 &= \left.\frac{\partial Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})}{\partial\boldsymbol{\theta}}\right|_{\theta=\theta^{(t+1)}} \\
&\approx \left.\frac{\partial Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})}{\partial\boldsymbol{\theta}}\right|_{\theta=\theta^{(t)}} + \left.\frac{\partial^2 Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})}{\partial\boldsymbol{\theta}^2}\right|_{\theta=\theta^{(t)}} (\theta^{(t+1)} - \theta^{(t)}). \qquad (3.34)
\end{aligned}
$$

Asumiendo válido el intercambio de la derivada con la integral, (3.34) resulta ser

$$0 \approx \mathbb{E}\left[\boldsymbol{S}_c(\theta^{(t)}) \,|\, y\right] - \mathcal{I}_c(\theta^{(t)} \,|\, y)(\theta^{(t+1)} - \theta^{(t)}). \tag{3.35}$$

Además, junto con el resultado de (3.25) obtenemos que

$$\boldsymbol{S}_{obs}(\theta^{(t)}) \approx \mathcal{I}_c(\theta^{(t)} \,|\, y)(\theta^{(t+1)} - \theta^{(t)}). \tag{3.36}$$

Sustituyendo en (3.33) y restándole $\theta^{(t)}$ obtenemos la siguiente expresión para θ^*

$$\begin{aligned}
\theta^* - \theta^{(t)} &\approx I_{obs}^{-1}(\theta^{(t)})\mathcal{I}_c(\theta^{(t)} \,|\, y)(\theta^{(t+1)} - \theta^{(t)}) \\
&= I_{obs}^{-1}(\theta^{(t)})\mathcal{I}_c(\theta^{(t)} \,|\, y)(\theta^{(t+1)} - \theta^* + \theta^* - \theta^{(t)}) \\
&= I_{obs}^{-1}(\theta^{(t)})\mathcal{I}_c(\theta^{(t)} \,|\, y)(\theta^{(t+1)} - \theta^*) + I_{obs}^{-1}(\theta^{(t)})\mathcal{I}_c(\theta^{(t)} \,|\, y)(\theta^* - \theta^{(t)}),
\end{aligned} \tag{3.37}$$

que por tanto nos da que

$$\begin{aligned}
\theta^{(t+1)} - \theta^* &\approx I_d - \mathcal{I}_c^{-1}(\theta^{(t)} \,|\, y) I_{obs}^{-1}(\theta^{(t)})(\theta^{(t)} - \theta^*) \\
&\approx I_d - \mathcal{I}_c^{-1}(\theta^* \,|\, y) I_{obs}^{-1}(\theta^*)(\theta^{(t)} - \theta^*).
\end{aligned} \tag{3.38}$$

Entonces, la tasa de convergencia dada en (3.31) la podemos expresar de la manera

$$J(\theta^*) = I_d - \mathcal{I}_c^{-1}(\theta^* \,|\, y) I_{obs}(\theta^*). \tag{3.39}$$

De (3.28) tenemos que

$$\mathcal{I}_m(\theta^* \,|\, y) = \mathcal{I}_c(\theta^* \,|\, y) - I_{obs}(\theta^*), \tag{3.40}$$

mientras que de (3.39) obtenemos lo siguiente

$$-I_{obs}(\theta^*) = -\mathcal{I}_c(\theta^* \,|\, y)(I_d - J(\theta^*)). \tag{3.41}$$

Sustituyendo en (3.40) nos queda

$$\mathcal{I}_m(\theta^* \,|\, y) = \mathcal{I}_c(\theta^* \,|\, y) - \mathcal{I}_c(\theta^* \,|\, y)(I_d - J(\theta^*)) \tag{3.42}$$

y por lo tanto, llegamos a la expresión

$$J(\theta^*) = \mathcal{I}_c^{-1}(\theta^* \,|\, y)\mathcal{I}_m(\theta^* \,|\, y). \tag{3.43}$$

Esto nos quiere decir que la tasa de convergencia depende de la proporción de información faltante que tengamos, conforme más información faltante haya, más lenta será la convergencia.

Del Ejemplo 2.3.1, comparando el número de iteraciones que el algoritmo necesita para llegar a la convergencia, con 0.01 % de tolerancia para distintos porcentajes de datos no faltantes, observamos el comportamiento de la tasa de convergencia por medio de la Figura 3.1.

Como podemos notar, conforme el número de datos faltantes disminuye, el número de iteraciones necesarias para que algoritmo termine va disminuyendo también, lo cual coincide con el resultado de (3.43).

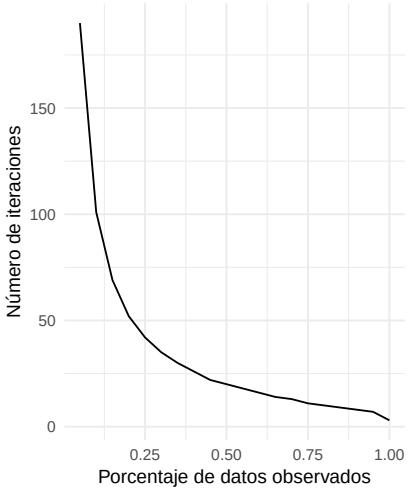

Figura 3.1: Tasa de convergencia con distintos porcentajes de datos observados. Software: R

3.3. Error estándar y aceleración de la convergencia

Una de las principales desventajas del algoritmo EM es que no calcula de manera directa la matriz de covarianzas del estimador máximo verosímil. Como se explicó en el Capítulo 1, podemos estimar a dicha matriz por medio de la matriz de información de Fisher.

Ejemplo 3.3.1. Retomando Ejemplo 2.3.2: Muestra de población multinomial

Retomando el Ejemplo 2.3.2 del Capítulo 2, calcularemos el error estándar de los parámetros obtenidos por medio del algoritmo. Obtendremos la información de Fisher directamente de los datos observados para compararla con aquella obtenida por medio del procedimiento que acabamos de enunciar.

Dada la log-verosimilitud

$$\begin{aligned} l(\theta; y) &= g(y; \theta) \\ &= \text{cte.} + y_1 \log\left(\frac{1}{2} - \frac{1}{2}\theta\right) + y_2 \log(\theta) + y_3 \log\left(\frac{1}{4}\theta + \frac{1}{2}\right) \end{aligned} \tag{3.44}$$

la información de Fisher es

$$I_{obs}(\theta) = \frac{y_1}{(1-\theta)^2} + \frac{y_2}{\theta^2} + \frac{y_3}{(\theta+2)^2}. \tag{3.45}$$

Evaluando en $\theta_{MLE} = 0.626815$ obtenemos

$$I_{obs}(\theta_{MLE}) = 377.5169. \tag{3.46}$$

Consideramos ahora la log-verosimilitud de los datos completos

$$\begin{aligned} l_c(\theta; x) &= f(x; \theta) \\ &= \text{cte} + y_1 \log\left(\frac{1}{2} - \frac{1}{2}\theta\right) + (y_2 + y_{31}) \log(\theta), \end{aligned} \tag{3.47}$$

y la información de Fisher correspondiente

$$I_c(\theta) = \frac{y_2 + y_{31}}{\theta^2} + \frac{y_1}{(1-\theta)^2}. \tag{3.48}$$

Dado que no conocemos a y_{31}, procedemos a obtener la información esperada dada por

$$\begin{aligned} \mathcal{I}_c(\theta) &= \frac{y_2 + \mathbb{E}\left[y_{31} \,|\, \boldsymbol{y}_{obs}\right]}{\theta^2} + \frac{y_1}{(1-\theta)^2} \\ &= \frac{y_2 + \frac{125\theta}{2+\theta}}{\theta^2} + \frac{y_1}{(1-\theta)^2}, \end{aligned} \tag{3.49}$$

pues $y_{31} \mid \boldsymbol{y}_{obs} \sim Bin\left(125, \frac{\frac{1}{4}\theta}{\frac{1}{4}\theta + \frac{1}{2}}\right)$, que se explicó en el ejemplo del capítulo anterior.

Evaluando (3.49) en el valor $\theta_{EM} = 0.020815$ obtenido por el algoritmo, visto en la Tabla 2.2, obtenemos el valor de

$$I_c(\theta_{EM}) = 435.3179. \tag{3.50}$$

Por otro lado, se calcula $\mathcal{I}_m(\theta)$ como

$$\begin{aligned} \mathcal{I}_m(\theta) &= \text{Var}\left(\boldsymbol{S}_c(\theta) \,|\, \boldsymbol{y}_{obs}\right) \\ &= \text{Var}\left(\frac{y_2 + y_{31}}{\theta} - \frac{y_1}{1-\theta} \,|\, \boldsymbol{y}_{obs}\right) \\ &= \theta^{-2}\text{Var}\left(y_{31} \,|\, \boldsymbol{y}_{obs}\right) \end{aligned} \tag{3.51}$$

con lo cual, evaluando en $\theta = \theta_{EM}$ obtenemos

$$I_m(\theta_{EM}) = 57.80095. \tag{3.52}$$

Por lo tanto,

$$\begin{aligned} I_{obs}(\theta_{EM}) &= 435.3179 - 57.80095 \\ &= 377.5169 \end{aligned} \tag{3.53}$$

que coincide con (3.46).

3.3.1. Aceleración del algoritmo vía el método de Aitken

Uno de los métodos más usados para acelerar algoritmos cuya tasa de convergencia es lineal es el método de aceleración de Aitken. Dados tres valores consecutivos de una sucesión convergente, el acelerador nos genera una aproximación a la solución, haciendo uso de la iteración

$$\hat{x}_n = x_{n-1} + \frac{x_n - x_{n-1}}{1 - \frac{x_{n+1}-x_n}{x_n - x_{n-1}}}. \tag{3.54}$$

Generalizando al caso multivariado $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)$, notemos que para $\boldsymbol{\theta}^*$ tal que $\boldsymbol{\theta}^{(t)} \to \boldsymbol{\theta}^*$ podemos expresarlo del siguiente modo

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(t)} + \sum_{k=0}^{\infty} \boldsymbol{\theta}^{(k+t+1)} - \boldsymbol{\theta}^{(k+t)}. \tag{3.55}$$

Además, por medio de una aproximación por expansión de Taylor de primer orden de $\boldsymbol{M}(\boldsymbol{\theta}^{(k+t)})$ alrededor de $\boldsymbol{\theta}^{(k+t-1)}$ obtenemos

$$\boldsymbol{M}(\boldsymbol{\theta}^{(k+t)}) \approx \boldsymbol{M}(\boldsymbol{\theta}^{(k+t-1)}) + \boldsymbol{J}(\boldsymbol{\theta}^{k+t-1})(\boldsymbol{\theta}^{(k+t)} - \boldsymbol{\theta}^{(k+t-1)}), \tag{3.56}$$

por lo cual obtenemos la siguiente expresión:

$$\begin{aligned} \boldsymbol{\theta}^{(k+t+1)} - \boldsymbol{\theta}^{(k+t)} &= \boldsymbol{M}(\boldsymbol{\theta}^{(k+t)}) - \boldsymbol{M}(\boldsymbol{\theta}^{(k+t-1)}) \\ &\approx \boldsymbol{J}(\boldsymbol{\theta}^{(k+t-1)})(\boldsymbol{\theta}^{(k+t)} - \boldsymbol{\theta}^{(k+t-1)}) \\ &\approx \boldsymbol{J}(\boldsymbol{\theta}^*)(\boldsymbol{\theta}^{(k+t)} - \boldsymbol{\theta}^{(k+t-1)}). \end{aligned} \tag{3.57}$$

Sustituyendo 3.57 en 3.55 nos da

$$\begin{aligned} \boldsymbol{\theta}^* &\approx \boldsymbol{\theta}^{(t)} + \sum_{k=0}^{\infty} \boldsymbol{J}(\boldsymbol{\theta}^*)(\boldsymbol{\theta}^{(k+t)} - \boldsymbol{\theta}^{(k+t-1)}) \\ &= \boldsymbol{\theta}^{(t)} + \sum_{k=0}^{\infty} \boldsymbol{J}(\boldsymbol{\theta}^*)^k(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}) \\ &= \boldsymbol{\theta}^{(t)} + (I_d - \boldsymbol{J}(\boldsymbol{\theta}^*))^{-1}(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}), \end{aligned} \tag{3.58}$$

que por 3.39, el resultado anterior puede expresarse en términos de las matrices de información, como

$$\boldsymbol{\theta}^* \approx \boldsymbol{\theta}^{(t)} + I_{obs}(\theta^*)^{-1}\mathcal{I}_c(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}). \tag{3.59}$$

Otro método de aceleración empleado en el algoritmo EM es el propuesto por Louis [16], derivado de la expresión 3.59, el cual genera la sucesión de valores $\{\boldsymbol{\theta}_{acc}^{(t)}\}$ del modo

$$\boldsymbol{\theta}_{acc}^{(t+1)} = \boldsymbol{\theta}_{acc}^{(t)} + I_{obs}(\boldsymbol{\theta}_{acc}^{(t)})^{-1}\mathcal{I}_c(\boldsymbol{\theta}_{acc}^{(t)})(\boldsymbol{\theta}_{EM}^{(t+1)} - \boldsymbol{\theta}_{acc}^{(t)}), \tag{3.60}$$

donde $\boldsymbol{\theta}_{EM}^{(t+1)}$ es el valor generado por el algoritmo EM usual, usando como valor anterior a $\boldsymbol{\theta}_{acc}^{(t)}$. El algoritmo acelerado queda expresado del siguiente modo:

Algoritmo 8 Método de Louis para aceleración del algoritmo EM

1: Inicializar parámetros $\boldsymbol{\theta}_{EM}^{(0)} = \boldsymbol{\theta}_{acc}^{(0)}$, $t = 0$.
2: Obtener $\boldsymbol{\theta}_{EM}^{(t+1)}$ por medio del algoritmo EM usual, donde $\boldsymbol{\theta}_{EM}^{(t)} = \boldsymbol{\theta}_{acc}^{(t)}$
3: Obtener $\boldsymbol{\theta}_{acc}^{(t+1)}$ por medio de

$$\boldsymbol{\theta}_{acc}^{(t+1)} = \boldsymbol{\theta}_{acc}^{(t)} + I_{obs}(\boldsymbol{\theta}_{acc}^{(t)})^{-1}\mathcal{I}_c(\boldsymbol{\theta}_{acc}^{(t)})(\boldsymbol{\theta}_{EM}^{(t+1)} - \boldsymbol{\theta}_{acc}^{(t)})$$

4: $t = t + 1$ y regresar a 2.

Ejemplo 3.3.2. Aceleración para ejemplo de población Multinomial

Ejemplificamos el método de aceleración haciendo uso del Ejemplo 2.3.2. Inicializando el parámetro θ con los cinco valores propuestos en dicho ejemplo, procedemos con los pasos del Algoritmo 8. Haciendo uso de las expresiones obtenidas del Ejemplo 3.3.1, la actualización de $\boldsymbol{\theta}_{acc}^{(t+1)}$ queda del siguiente modo

$$\begin{aligned}
\theta_{acc}^{(t+1)} &= \theta_{acc}^{(t)} + I_{obs}(\theta_{acc}^{(t)})^{-1}\mathcal{I}_c(\theta_{acc}^{(t)})(\theta_{EM}^{(t+1)} - \theta_{acc}^{(t)}) \\
&= \theta_{acc}^{(t)} + (\mathcal{I}_c(\theta_{acc}^{(t)}) - \mathcal{I}_m(\theta_{acc}^{(t)}))^{-1}\mathcal{I}_c(\theta_{acc}^{(t)})(\theta_{EM}^{(t+1)} - \theta_{acc}^{(t)}) \\
&= \theta_{acc}^{(t)} + \left(\frac{y_2 + \frac{125\theta_{acc}^{(t)}}{2+\theta_{acc}^{(t)}}}{\theta_{acc}^{(t)2}} + \frac{y_1}{(1-\theta_{acc}^{(t)2})} - \theta_{acc}^{(t)^{-2}}\mathrm{Var}\left(y_{31} \mid \boldsymbol{y}_{obs}\right) \right)^{-1} \\
&\left(\frac{y_2 + \frac{125\theta}{2+\theta}}{\theta^2} + \frac{y_1}{(1-\theta_{acc}^{(t)2})} \right)(\theta_{EM}^{(t+1)} - \boldsymbol{\theta}_{acc}^{(t)}).
\end{aligned} \tag{3.61}$$

Implementando en R un programa que realizara el algoritmo (ver código A.11 en Apéndice), se llegaron a los resultados mostrados en la Tabla 3.1.

Iteración (t)	$\theta_1^{(t)}$	$\theta_2^{(t)}$	$\theta_3^{(t)}$	$\theta_4^{(t)}$	$\theta_5^{(t)}$
0	0.5	0.9	0.01	0.2	0.75
1	0.6363636	0.6549375	0.4852347	0.6392103	0.6348552
2	0.6268769	0.6272961	0.6384904	0.6269148	0.6268608
3	0.6268215	0.6268216	0.6269043	0.6268215	0.6268215
4	0.6268215	0.6268215	0.6268215	0.6268215	0.6268215

Tabla 3.1: Convergencia de estimadores para θ con 5 distintos valores iniciales, usando método de aceleración de Louis

Como podemos notar, a diferencia de los resultados del Ejemplo 2.3.2, el algoritmo convergió en 4 iteraciones.

Capítulo 4

Algoritmo EM Monte Carlo (EMMC)

Unas de las desventajas que tiene el algoritmo EM, del cual hemos hablado en los demás capítulos, es que puede converger a puntos silla y su convergencia puede llegar a ser muy lenta. Por ello, presentamos en este capítulo la versión estocástica del algoritmo EM, la cual vence dichos problemas. Dicha versión es de gran utilidad y se recurre a ella en situaciones en las cuales la esperanza condicional del paso E del algoritmo no es fácil de obtener. Además, dado su caracter estocástico, tiene como ventaja que su convergencia no depende de los valores iniciales.

Primero que nada, daremos una breve introducción al método de integración vía Monte Carlo, que estaremos utilizando en los ejemplos que se presentarán más adelante.

4.1. Introducción a los métodos de Monte Carlo

Los métodos de Monte Carlo son métodos de aproximación vía simulación, empleados en distintas tareas necesarias en un sinnúmero de técnicas estadísticas, cuando las soluciones son dificiles o imposibles de obtener de manera analítica. Entre estas tareas está la integración, la cual enunciamos a contiuación.

4.1.1. Integración

Buscamos obtener la integral

$$\int_a^b g(x)dx. \tag{4.1}$$

La idea del método Monte Carlo para integración es expresar a la integral como la esperanza de una variable aleatoria Y adecuada. Una vez definida dicha variable aleatoria, se genera una muestra $y_1, ..., y_m$ con lo cual calculamos el estimador de Monte Carlo dado por

$$\hat{\mathbb{E}}(g(Y)) = \frac{1}{m}\sum_{i=1}^{m} g(y_i), \tag{4.2}$$

el cual, por la Ley de los Grandes Números, converge a $\mathbb{E}(g(Y))$.

Ejemplo 4.1.1. Supongamos que queremos aproximar el valor de π. Para ello, primero calcularemos el área de un cuarto de círculo unitario, dado por la integral

$$\int_0^1 \sqrt{1-x^2}dx. \tag{4.3}$$

Proponemos a la variable aleatoria $X \sim U(0,1)$, con lo cual, la integral (4.3) resulta ser $\mathbb{E}\left[\sqrt{1-X^2}\right]$, cuyo estimador Monte Carlo es, dada una muestra $x_1, ..., x_m$

$$\frac{1}{m}\sum_{i=1}^{m}\sqrt{1-x_i^2}. \tag{4.4}$$

Otro método para estimar el valor de π sería generar variables aleatorias distribuidas de manera uniforme en el cuadro unitario y promediando el número de veces que las variables generadas cayeron dentro del círculo unitario con respecto del número total de simulaciones y multiplicamos dicho promedio por 4. En la Figura 4.1 mostramos el resultado de ambos métodos.

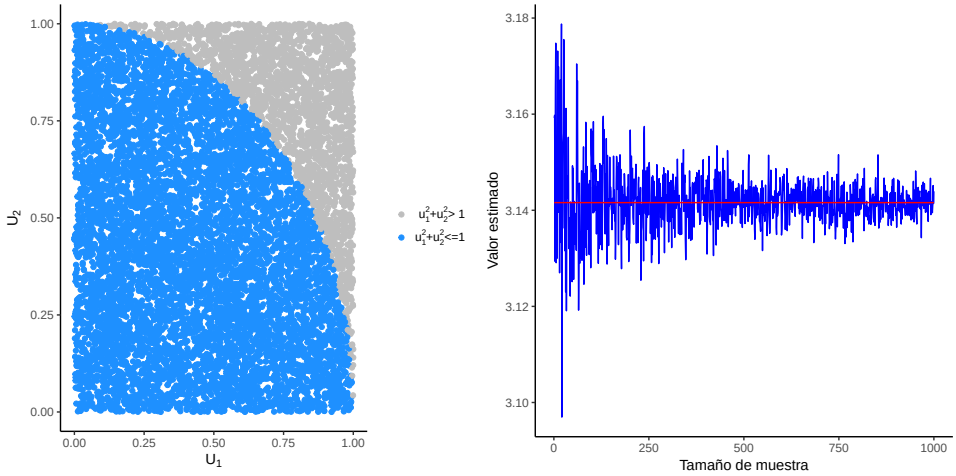

Figura 4.1: Izquierda: Estimación de $\pi/4$ por medio del área de un cuarto de círculo unitario. Derecha: Estimador Monte Carlo visto en (4.4) incrementando el tamaño de muestra, de 100 a 100,000 observaciones. Software: R

4.2. Formulación del algoritmo EMMC

A diferencia del algoritmo EM original, el algoritmo EM Monte Carlo (al cual nos refermos como EMMC), sustituye a la esperanza que se calculaba en el paso E del algoritmo EM original, por el estimador Monte Carlo de dicha esperanza, dada por

$$\hat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) = \frac{1}{m}\sum_{i=1}^{m} l_c(\theta;x), \tag{4.5}$$

donde m es el tamaño de la muestra generada por la función de densidad condicional, $k(\boldsymbol{x}\,|\,\boldsymbol{y})$, es decir, la función de densidad correspondiente a los datos faltantes dados los datos observados.

Una vez obtenido dicho estimador, se procede al paso M y actualizamos $\boldsymbol{\theta}$ como se ha hecho en capítulos anteriores. El algoritmo queda definido entonces del siguiente modo:

Algoritmo 9 Algoritmo EM Monte Carlo

1: Inicializar parámetros $\boldsymbol{\theta}_i$ para i en $\{1, 2, ..., M\}$, $t = 0$
2: **Paso MC E** Generamos una muestra $\boldsymbol{z}_1, ..., \boldsymbol{z}_m$ de tamaño m adecuado, de la distribución condicional de los datos faltantes dados los datos observados.
3: Calcular $\hat{Q}(\theta; \theta^{(t)}) = \frac{1}{m} \sum_{i=1}^{m} l_c(\theta; \boldsymbol{y}, \boldsymbol{z}_i)$.
4: **Paso M** Maximizar $\hat{Q}(\theta; \theta^{(t)})$ y actualizar parámetros.
5: Regresar al paso 2 y repetir hasta que se estabilice el proceso.

Notemos que en el paso 5 mencionamos la estabilidad de la secuencia generada por el algoritmo. Para ello, el criterio que usaremos será identificar visualmente el momento en el cual la cadena de Markov llega a la estacionariedad, para así determinar el periodo de calentamiento. Sin embargo, para las pruebas teóricas de convergencia de estos algoritmos, refiérase a [20].

4.3. Algoritmo EM estocástico (EME)

El algoritmo EM estocástico (al cual nos referimos como EME) difiere del EMMC, o más bien, puede pensarse como un caso específico del EMMC en el cual el tamaño de muestra m es 1, es decir, solamente simulamos una vez a los datos faltantes. El algoritmo queda expresado del siguiente modo:

Algoritmo 10 Algoritmo EM Estocástico

1: Inicializar parámetros $\boldsymbol{\theta}_i$ para i en $\{1, 2, ..., M\}$, $t = 0$
2: **Paso MC E** Generamos un valor $\boldsymbol{z}$ de la distribución condicional de los datos faltantes dados los datos observados.
3: Calcular $\hat{Q}(\theta; \theta^{(t)}) = l_c(\theta; \boldsymbol{y}, \boldsymbol{z})$.
4: **Paso M** Maximizar $\hat{Q}(\theta; \theta^{(t)})$ y actualizar parámetros.
5: Regresar al paso 2 y repetir hasta que se estabilice el proceso.

Definición 6. Definimos al proceso $\tilde{\theta}_n(k)$ como el parámetro estimado de la muestra completa X de tamaño n en la k-ésima iteración del EME.

Intuitivamente, podemos notar que se trata de una cadena de Markov y que es necesario también pensar en un tamaño de muestra factible para tener suficiencia de información y bajo costo de implementación. El objetivo principal es hacer uso de las propiedades de la cadena de Markov antes mencionada. Consideremos primero la siguiente cadena de Markov.

Definición 7. Definimos al proceso $\tilde{X}(k)$ como las observaciones simuladas faltantes en la k-ésima iteración del EME dado el valor de $\tilde{\theta}_n(k)$.

La forma en que se encuentran definidos ambos procesos implica el resultado crucial para la convergencia, la cadena de Markov $\tilde{X}(k)$ proporciona ciertas propiedades a la cadena de Markov $\tilde{\theta}_n(k)$. Los siguientes resultados nos permiten justificar la aplicación del algoritmo.

Lema 4.3.1. *La cadena de Markov $\tilde{X}(k)$ es irreducible y aperiódica.*

Lema 4.3.2. *La cadena de Markov $\tilde{\theta}_n(k)$ cumple la propiedad de Feller.*

Teorema 4.3.1. *Sea $c = \log \int k_{\tilde{\theta}}(\tilde{x} \,|\, y) dv_y(\tilde{x})$, donde $\tilde{\theta}$ es el valor máximo verosímil para $\tilde{x}$. Supongamos que $c < \infty$ y que la función*

$$\theta \to \Delta(\theta) = \tilde{E}(\log f_{\tilde{\theta}(k+1)}(\tilde{X}) - \log f_{\tilde{\theta}(k)}(\tilde{X}) \,|\, \tilde{\theta}(k) = \theta),$$

donde $\tilde{X}$ tiene la distribución condicional a las observaciones conocidas, es más grande que $(1 + \delta)c$ para algún $\delta > 0$.

	$m = 1$	$m = 10$	$m = 100$
$\hat{\mu}_{\hat{\theta}_{MCEM}}$	0.6216783	0.624364	0.6244651
$\hat{\sigma}_{\hat{\theta}_{MCEM}}$	0.004267926	0.00433769	0.004560546
Tiempo CPU promedio	0.0009733268	0.0008787428	0.002172402

Tabla 4.1: Promedios y desviaciones estándar de $\hat{\theta}_{multMCEM}$ para cada tamaño de muestra y tiempo CPU promedio

Entonces la cadena de Markov $\tilde{\theta}_n(k)$ tiene distribución estacionaria, y además, si $\Delta(\theta)$ se distribuye normalmente entonces la cadena es ergódica.

Además, como lo demuestra [20], la distribución estacionaria converge a la distribución normal, con lo cual podemos usar como estimador final el promedio de los valores de la cadena, excluyendo en dicho promedio los valores del periodo de calentamiento.

4.4. Ejemplos e implementación en R

4.4.1. Población Multinomial

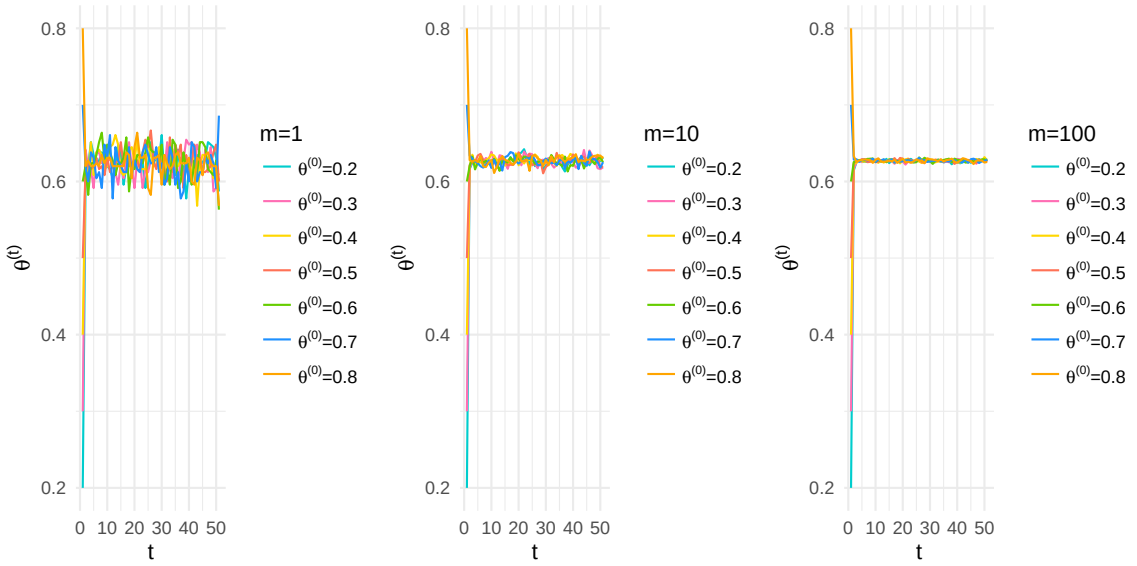

Figura 4.2: Convergencia de parámetros. Software: R

Se presenta el mismo ejemplo visto en el Capítulo 2, pero esta vez haciendo uso de del algoritmo EM Monte Carlo. La log-verosimilitud de los datos completos es

$$\begin{aligned} l_c(\theta; x) &= f(x; \theta) \\ &= \text{cte} + y_1 \log\left(\frac{1}{2} - \frac{1}{2}\theta\right) + (y_2 + y_{31}) \log(\theta), \end{aligned} \tag{4.6}$$

Dado que $Y_{31} \sim \text{Bin}(125, \frac{\theta}{2+\theta})$, procedemos a generar una muestra $z_1, ..., z_m$ proveniente de dicha distibución para obtener el estimador MonteCarlo de la función $Q(\theta; \theta^{(k)})$ como

$$\hat{Q}(\theta;\theta^{(k)}) = y_1 \log\left(\frac{1}{2} - \frac{1}{2}\theta\right) + (y_2 + z_{31})\log(\theta), \tag{4.7}$$

donde $z_{31} = \frac{1}{m}\sum_{i=1}^{m} z_i$. Una vez completado el paso E, realizamos el paso M del mismo modo que se hizo en el capítulo 2.

Se implementó el código A.5 con 7 distintos valores iniciales, como se muestra en la Figura 4.2, para tamaños de muestra de $m = 1, 10$ y 100. Es decir, en total se corrió 21 veces el algoritmo, haciendo 50 iteraciones y de las cuales las primeras 30 las consideramos para el periodo de calentamiento de la cadena. Por el resultado dado en [20] que nos dice que la distribución estacionaria converge a una distribución normal, aplicamos para cada secuencia generada la prueba de Anderson-Darling para probar normalidad, con lo cual confirmamos con un nivel de $\alpha = 5\,\%$ que 20 de las 21 secuencias generadas cumplen que siguen una distribución normal, con lo cual se obtuvieron los estimadores finales definidas como:

$$\hat{\theta}_{MCEM} = \frac{1}{20}\sum_{t=30}^{50} \theta^{(t)}.$$

De la Tabla 4.1 podemos notar que los promedios obtenidos de los 7 distintos valores de $\hat{\theta}_{MCEM}$ para cada tamaño de muestra, son muy parecidos al valor original $\theta_{MLE} = 0.626821497$. Además, comparamos en la misma tabla los tiempos CPU promedio por cada tamaño de muestra con el tiempo CPU promedio del la versión determinista del algoritmo EM implementado para este mismo problema, que resultó ser 0.001579571. Como podemos notar también de la Figura 4.2, conforme aumenta el tamaño de muestra, claramente la varianza se reduce.

4.4.2. Ejemplo de las viudas

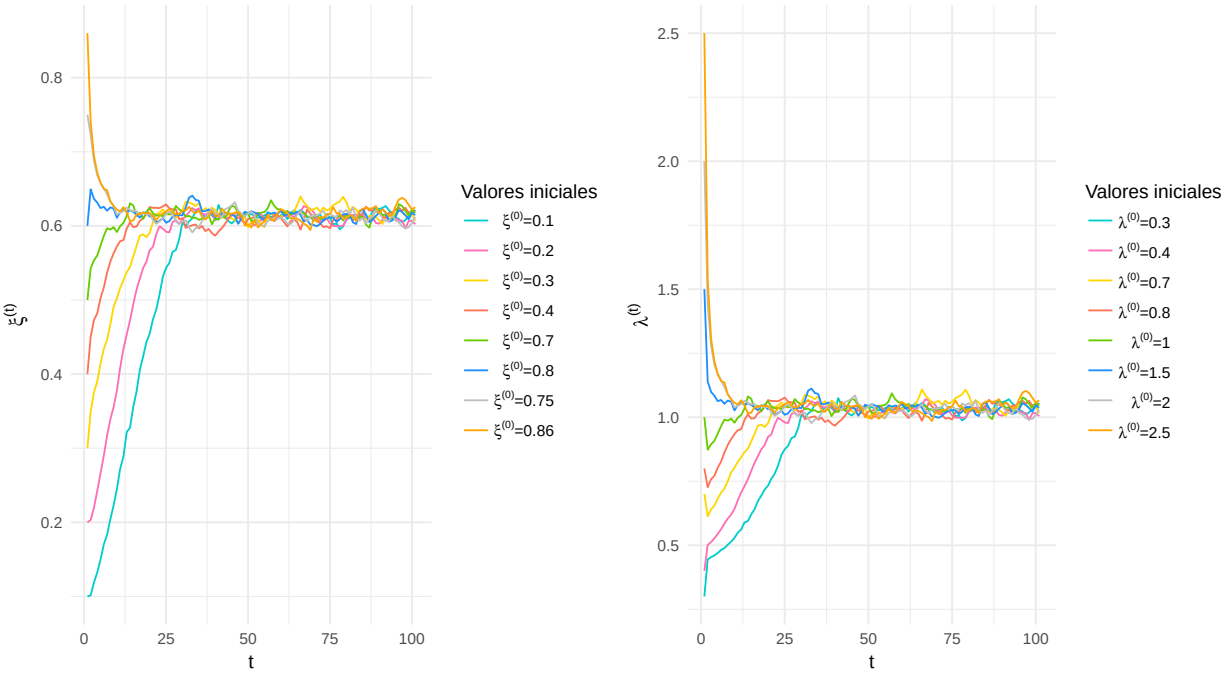

Figura 4.3: Convergencia de parámetros. Software: R

Presentamos el mismo ejemplo de las viudas visto en el capítulo 2, pero ahora implementando el algoritmo EME. La única diferencia entre este ejemplo y el del capítulo 2 es que simulamos una variable aleatoria con distribución binomial de parámetros y_0 y $p1$, definidas en dicho ejemplo. Implementamos el código A.6 incializando los parámetros con 7 valores distintos como se ve en la Figura 4.3. Para esta implementación corrimos el algoritmo con 100 iteraciones, con un periodo de calentamiento de 70 iteraciones. La media de los estimadores finales $\hat{\theta}_{MCEM}$ resultó ser de 0.613675 para ξ y de 1.03457 para λ y desviaciones estándar de 0.004045165 y 0.1513798, respectivamente, y un tiempo CPU promedio de 0.001997858.

4.4.3. Efectos aleatorios, componentes de varianza

Implementamos la versión EMMC para el ejemplo visto en el capítulo 2 de componentes de varianza. Como se mencionó anteriormente, cada $a_i \sim N\left(\frac{\sigma^2\mu + n_i\sigma_a^2\bar{y}_i}{\sigma^2 + n_i\sigma_a^2}, \frac{\sigma^2\sigma_a^2}{\sigma^2 + n_i\sigma_a^2}\right)$, para $i = 1, ..., 6$. Por lo tanto, el algoritmo para este ejemplo se expresa del siguiente modo:

Algoritmo 11 Algoritmo EM MonteCarlo

1: Inicializar parámetros $\mu^{(0)}, \sigma^{2^{(0)}}, \sigma_a^{2^{(0)}}$, $t = 0$.
2: **Paso MC E** Generar una muestra $a_{i,1}^{(t)}, ..., a_{i,m}^{(t)}$ de la distribución $N\left(\frac{\sigma^2\mu + n_i\sigma_a^2\bar{y}_i}{\sigma^2 + n_i\sigma_a^2}, \frac{\sigma^2\sigma_a^2}{\sigma^2 + n_i\sigma_a^2}\right)$ para $i = 1, ..., 6$. Utilizaremos el tamaño $m = 10000$.
3: **Paso M** Actualizar parámetros.

$$\mu^{(t+1)} = \frac{1}{6m}\sum_{k=1}^{m}\sum_{i=1}^{6} a_{i,k}^{(t)}$$

$$\sigma^{2^{(t+1)}} = \frac{1}{6m}\sum_{k=1}^{m}\sum_{i=1}^{6}\left(a_{i,k}^{(t)} - \mu^{(t+1)}\right)$$

$$\sigma_a^{(t+1)} = \frac{1}{m\sum_{i=1}^{6} n_i}\sum_{k=1}^{m}\sum_{i=1}^{6}\sum_{j=1}^{n_i}\left(y_{ij} - a_{i,k}^{(t)}\right)$$

4: $t = t + 1$.
5: Repetir hasta que se estabilice el proceso.

4.5. Aplicaciones

4.5.1. Continuación. Modelo de riesgo modulado

La aplicación que se presenta a continuación de la versión estocástica del algoritmo EM propone una manera para hacer inferencia sobre el proceso modulador explicado en el Capítulo 2. Hacer inferencia en dicho proceso se traduce a obtener estimadores para la matriz de intensidades $Q = \{q_{ij}\}_{i,j\in E}$.

4.5.2. Puentes de Markov

En muchas ocasiones, se da el caso en el que el proceso de saltos de Markov a tratar es observado de manera discreta, es decir, se tienen las observaciones $\boldsymbol{S} = \{S_{t_1}, ..., S_{t_N}\}$, con $t_1 < ... < t_N$ conocidos. Sin embargo, el proceso en sí es uno a tiempo continuo, por lo cual podemos tratar al conjunto de observaciones $\boldsymbol{S} = \{S_{t_1}, ..., S_{t_N}\}$ como el conjunto de datos incompletos

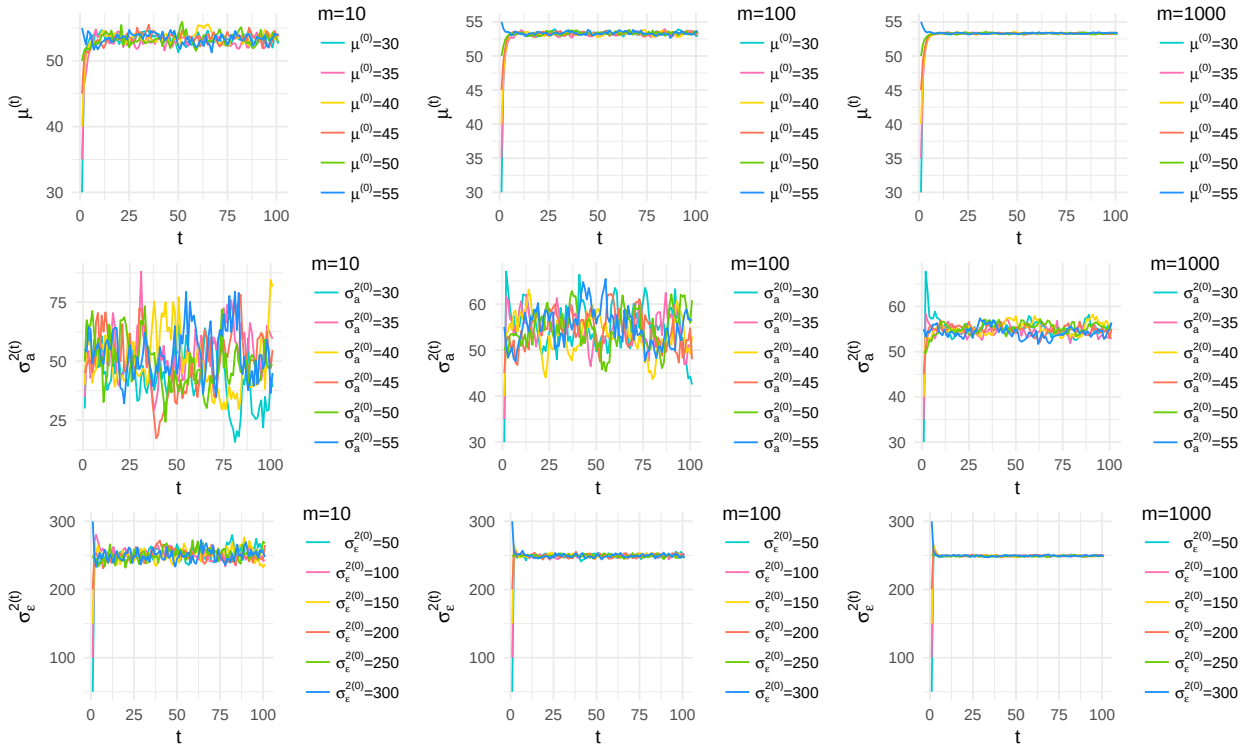

Figura 4.4: Convergencia de parámetros. Software: R

y con este planteamiento hacer uso del algoritmo EM. Notemos que, de tener la trayectoria completa, se reduce el problema a obtener:

- $N_{ij}(T)$: Número de veces que el proceso salta del estado i al j en el intervalo $[0, T]$
- $R_i(T)$: Tiempo total en el estado i, en el intervalo de tiempo $[0, T]$.

Dados esos valores, los estimadores máximo verosímiles para la matriz de intensidades Q están dados por:

$$\hat{q}_{ij} = \frac{N_{ij}(T)}{R_i(T)} \quad j \neq i \tag{4.8}$$

$$\hat{q}_{ii} = -\sum_{j \neq i} \hat{q}_{ij} \tag{4.9}$$

En el caso en el cual no contamos con la trayectoria continua, usando el algoritmo EM, en el paso E a iteración (t) obtenemos las esperanzas condicionales de los estadísticos $R_i(T)$ y $N_{ij}(T)$, dadas las observaciones $\{S_{t_1}, ..., S_{t_N}\}$ y la matriz estimadas de intensidades, $Q^{(t)}$. Es decir, calcular

$$R_i^{(t)} = \mathbb{E}_{Q^{(t)}} \left[R_i(T) \,|\, S_{t_1}, ..., S_{t_N} \right] \tag{4.10}$$

$$N_{ij}^{(t)} = \mathbb{E}_{Q^{(t)}} \left[N_{ij}(T) \,|\, S_{t_1}, ..., S_{t_N} \right]. \tag{4.11}$$

Notemos que, por la propiedad de Markov podemos particionar el problema en $N - 1$ problemas independientes, pues $\{S_{t_1}, ..., S_{t_N}\}$ tiene N observaciones. En lugar de calcular las esperanzas, es conveniente hacer uso de la simulación e implementar el algoritmo EM estocástico,

condicionando a que en los tiempos $\{t_i, t_{i+1}\}$ el proceso fue observado, para $i \in \{1, ..., N-1\}$. El modo en el cual implementamos este procedimiento es por medio de generación de puentes de Markov, los cuales definimos a continuación.

Definición 8. Sea $S = \{S(t) : t \geq 0\}$ un proceso de saltos de Markov con espacio de estados E finito, matriz de intensidades $Q = \{q_{ab}\}_{a,b \in E}$ y matriz de probabilidades de transición $P_{ab}(t) = e^{Qt}$. Un puente de Markov con parámetros (a, b, T) es un proceso estocástico con la distribución de $S = \{S(t) : t \leq T\}$ condicionado a $S(0) = a$, $S(T) = b$.

El método que se usó para simular los puentes fue el método de la bisección, propuesto por Asmussen y Hobolth [3], el cual parte de observar dos casos:

- Si $S(0) = S(T) = a$ y no hubo ningún salto, el procedimiento termina y el puente está dado por $S(t) = a$, para $t \in [0, T]$.
- Si $S(0) = a$ y $S(T) = b$, con $a \neq b$ y solamente hubo un salto, lo único que queda por hacer es simular el tiempo de salto τ, con lo cual, el puente está dado por $S(t) = a$, para $t \in [0, \tau]$ y $S(t) = b$, para $t \in [\tau, T]$.

Partiendo de esta observación, vamos biseccionando el intervalo completo $[0, T]$, eligiendo el estado en el que está el proceso a tiempo $T/2$ y cuántos saltos realizó en los subintervalos $[0, T/2]$ y $[T/2, T]$. Para cada subintervalo generado realizamos el mismo procedimiento y paramos hasta que en cada subintervalo tengamos alguno de los casos mencionados en 4.5.2.

Además, se tiene el siguiente Lema enunciado en [3].

Lema 4.5.1. *Consideremos un intervalo de longitud T con $S(0) = a$ y sea $b \neq a$. La probabilidad de que $S(t) = b$ y que solamente se haya presentado un salto en el intervalo $[0, T]$ está dada por*

$$R_{ab} = q_{ab} \begin{cases} \dfrac{e^{-q_a T} - e^{-q_b T}}{q_b - q_a} & \text{si } q_a \neq q_b \\ T e^{-q_a T} & \text{si } q_a = q_b. \end{cases}$$

La función de densidad del tiempo de salto es

$$f_{ab}(t; T) = \frac{q_{ab} e^{-q_b T}}{R_{ab}(T)} e^{-(q_a - q_b)t}, \quad 0 \leq t \leq T. \tag{4.12}$$

Además, la probabilidad de que $S(T) = b$ y que hayan por lo menos dos saltos en $[0, T]$ es $P_{ab}(T) - R_{ab}(T)$.

Refiérase a [3] para ver la demostración.

Notemos que dependiendo de si $q_a = q_b, q_a > q_b$ o $q_a < q_b$ se tendrá respectivamente que el tiempo de salto se distribuye de manera uniforme en $[0, T]$, de forma exponencial truncada en $[0, T]$ o bien, que $f_{ab}(t; T)$ es la densidad de la variable aleatoria $T - X$, con X distribuida de forma exponencial truncada en $[0, T]$ con parámetro $q_b - q_a$, esto último dada la simetría de $f_{ab}(t; T)$.

Dadas las parejas contiguas de observaciones discretas de nuestro proceso modulador, dependiendo de si en ambos extremos del intervalo el proceso está en el mismo estado o no, se usarán las Tablas 4.2 y 4.3, respectivamente, con $e_a = e^{-q_a T/2}$, $r_{ab} = R_{ab}(T/2)$, $p_{ab} = P_{ab}(T/2)$.

Caso	Saltos en $(0, T/2)$	Saltos en $(T/2, T)$	Probabilidad	Notación
1	0	0	$e_a e_a$	α_1
2	0	≥ 2	$e_a(p_{aa} - e_a)$	α_2
3	≥ 2	0	$(p_{aa} - e_a)e_a$	α_3
4	≥ 2	≥ 2	$(p_{aa} - e_a)(p_{aa} - e_a)$	α_4
5	1	1	$r_{ac} r_{ca}$	$\alpha_{5,c}$
6	1	≥ 2	$r_{ac}(p_{ca} - r_{ca})$	$\alpha_{6,c}$
7	≥ 2	1	$(p_{ac} - r_{ac})r_{ca}$	$\alpha_{7,c}$
8	≥ 2	≥ 2	$(p_{ac} - r_{ac})(p_{ca} - r_{ca})$	$\alpha_{,c}$

Tabla 4.2: Tabla de casos para $S(0) = S(T) = a$

Caso	Saltos en $(0, T/2)$	Saltos en $(T/2, T)$	Probabilidad	Notación
1	0	1	$e_a r_{ab}$	β_1
2	0	≥ 2	$e_a(p_{ab} - r_{ab})$	β_2
3	≥ 2	1	$(p_{aa} - e_a)r_{ab}$	β_3
4	≥ 2	≥ 2	$(p_{aa} - e_a)(p_{ab} - r_{ab})$	β_4
5	1	0	$r_{ab} e_b$	β_5
6	1	≥ 2	$r_{ab}(p_{bb} - e_b)$	β_6
7	≥ 2	0	$(p_{ab} - r_{ab})e_b$	β_7
8	≥ 2	≥ 2	$(p_{ab} - r_{ab})(p_{bb} - e_b)$	β_8
9	1	1	$r_{ac} r_{cb}$	$\beta_{9,c}$
10	1	≥ 2	$r_{ac}(p_{cb} - r_{cb})$	$\beta_{10,c}$
11	≥ 2	1	$(p_{ac} - r_{ac})r_{cb}$	$\beta_{11,c}$
12	≥ 2	≥ 2	$(p_{ac} - r_{ac})(p_{cb} - r_{cb})$	$\beta_{12,c}$

Tabla 4.3: Tabla de casos para $S(0) = a$ y $S(T) = b$, $a \neq b$

Entonces, para generar una trayectoria continua del proceso modulador, dada una trayectoria observada de manera discreta, generaremos un puente que comience en S_{t_i} y termine en $S_{t_{i+1}}$, para cada $i \in \{1, ..., N-1\}$. Finalmente se concatenan dichas trayectorias dando como resultado una trayectoria continua del proceso, con matriz estimada de intensidades $\hat{Q}^{(t)}$. Una vez realizado eso procedemos a obtener los estimadores máximo verosímiles y actualizamos la matriz de intensidades estimada. Dada esa nueva matriz, volvemos a generar los puentes y se repite el proceso hasta que los estimadores se estabilicen.

Implementado dicho procedimiento en R, haciendo uso de los mismo datos proporcionados en la aplicación vista en el Capítulo 2, se realizaron 1000 iteraciones, con lo cual se generó el proceso modulador que se muestra en la Figura 4.5, junto con el proceso de Cramer Lundberg sobre el cual el proceso modulador influye. Además, se muestra la secuencia de valores generados a cada iteración, para los valores de $-q_{ii}$ de la matriz de intensidades Q.

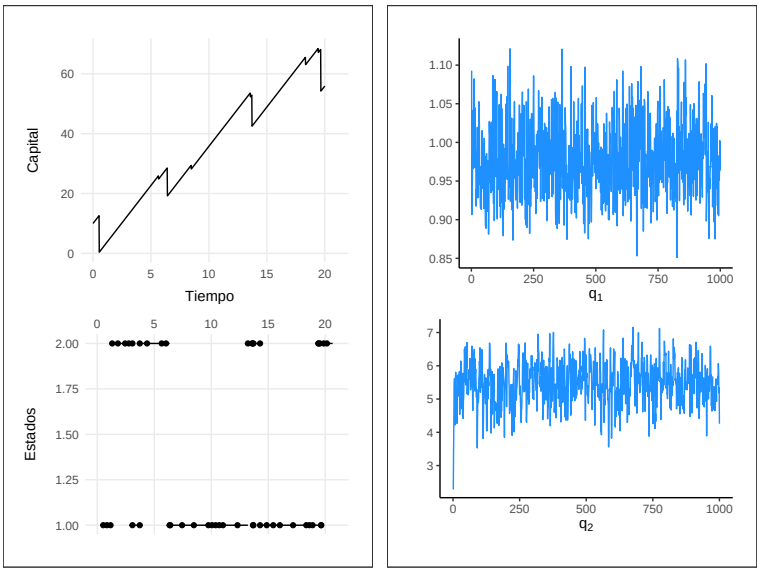

Figura 4.5: Izquierda: Modelo de Cramer-Lundberg modulado con dos estados y el proceso de saltos de Markov de los estados moduladores con estimadores óptimos. Derecha: Secuencia de valores estimados de los elementos diagonales de la matriz de intensidades del proceso, obtenidos por medio de la generación de puentes de Markov. Software: R

Conclusión

En conclusión podemos decir que el algoritmo EM es una buena herramienta para obtener estimadores máximo verosímiles para problemas con información incompleta. Vimos algunos de sus principales resultados y, aunque se presentaron algunas variantes del algoritmo, cabe recalcar que hay muchas más versiones que no se mencionaron en este trabajo. Al día de hoy se siguen desarrollando diversas extensiones del algoritmo EM, muchas de las cuales proponen utilizar simulación vía métodos de Monte Carlo vía Cadenas de Markov.

Algunas otras de las aplicaciones del algoritmo EM que no fueron presentadas en este trabajo son por ejemplo en cadenas ocultas de Markov, en donde el algorimto EM es mejor conocido como algoritmo de Baum Welch [4] , que incluso fue publicado años antes del paper de DLR. Otra aplicación interesante es en redes neuronales, que a pesar de ser un modelo determinista, puede ser entrenado como a una red estocástica, en la cual puede emplearse la metodología estadística y con lo cual puede implementarse el algoritmo EM, ver por ejemplo, [1]. Algunas otras aplicaciones son por ejemplo en reconstrucción de imágenes, procesamiento de señales, genética, entre otras.

Apéndice A

Apéndice

A.1. Muestra Normal Univariada

```
logLike <- function(X,mu,sigma2){
  n = length(X)
  a = -(n/2)*log(2*pi)-(n/2)*log(sigma2)
  b = (X-mu)^2
  c = -1/(2*sigma2)
  verosimilitud = a +(c*sum(b))
  return(verosimilitud)
}


NormalUnivariateEM <- function(y_tot,y_obs,mu0,sigma20,tolerancia){
  n = length(y_tot)
  m = length(y_obs)
  me=mean(y_obs)
  ms = mean(y_obs^2)-mean(y_obs)^2
  mus <- NULL
  sigmas2 <- NULL
  L <- NULL
  #E-step:
  s1 = sum(y_obs)+(n-m)*(mu0)
  s2 = sum(y_obs^2)+(n-m)*(sigma20+mu0^2)
  # M-step:
  j = 1
  mus <- s1/n
  sigmas2 <- s2/n -mus^2
  sd <- sqrt(sigmas2)
  tamano = n-m
  y_fal <- rnorm(tamano,mus,sd)
  y_completa <- c(y_obs,y_fal)
  L <- logLike(y_obs,mus,sigmas2)
  for(j in 2:20){
    s1 = sum(y_obs)+(n-m)*mus[j-1]
    s2 = sum(y_obs^2)+(n-m)*(sigmas2[j-1]+mus[j-1]^2)
    mus[j] = s1/n
    sigmas2[j] = s2/n - mus[j]^2
    sd <- sqrt(sigmas2[j])
```

```
    tamano = n-m
    y_fal <- rnorm(tamano,mus[j],sd)
    y_completa <- c(y_obs,y_fal)
    L[j] = logLike(y_obs,mus[j],sigmas2[j])
  }
  #result <- list(n=n,m=m,mus=mus,sigmas2=sigmas2,L=L,y_tot=y_tot,y_obs=y_obs,
      me=me,ms=ms,info=INFO)
  result <- list(n=n,m=m,mus=mus,sigmas2=sigmas2,L=L,y_tot=y_tot,y_obs=y_obs,me
      =me,ms=ms)
  return(result)
}
```

A.2. Población Multinomial

```
MultinomialEM <- function(y_obs,MLEobs,theta0,tolerancia){
  y3 <- 125*0.25*theta0/(0.5+0.25*theta0)
  thetas <- NULL
  i = 1
  y3 <- 125*0.25*theta0/(0.5+0.25*theta0)
  thetas[i] <- (y_obs[2] + y3)/(y_obs[1]+y_obs[2]+y3)
  while (abs(thetas[i]-MLEobs) > tolerancia) {
    i = i+1
    y3 <- 125*0.25*thetas[i-1]/(0.5+0.25*thetas[i-1])
    thetas[i] <- (y_obs[2] + y3)/(y_obs[1]+y_obs[2]+y3)
  }
  return(thetas = thetas)
}
```

A.3. Mezcla de distribuciones Binomial-Poisson

```
logfactorial<-function(y){
  return(sum(log(1:y)))
}

ini<-function(i){
  return(i*widows[i])
}
logfactni<-function(i){
  return(widows[i]*logfactorial(i))
}

BinPoisMixtureEM <- function(children,widows,xi0,lambda0,eps=0.0001){
  xis <- NULL
  lambdas<- NULL
  nA <- NULL
  nB <- NULL
  lc<-NULL
  # E-step:
```

```
nA[1] = widows[1]*xi0/(xi0+(1-xi0)*exp(-lambda0))
nB[1] = widows[1]-nA[1]
i=1
# M-step:
xis[i] <- nA[i]/sum(widows)
lambdas[i] <- sum(children[-1]*widows[-1])/(nB[i]+sum(widows[-1]))
xi=tail(xis,1)
lambda=tail(lambdas,1)
loglikec<-logfactorial(sum(widows))-sum(sapply(widows,
                                        logfactorial))+widows[1]*log(
                                          xi+(1-xi)*exp(-lambda))+(log(1-xi
                                              )-lambda)*sum(
                                            widows[-1])+log(lambda)*sum(
                                                sapply(2:length(widows),ini)
                                                )-sum(sapply(2:length(widows
                                                ),logfactni))

lc<-c(lc,loglikec)
i=i+1
# E-step:
nA[i] = widows[1]*xi/(xi+(1-xi)*exp(-lambda))
nB[i] = widows[1]-nA[i]
# M-step:
xis[i] <- nA[i]/sum(widows)
lambdas[i] <- sum(children[-1]*widows[-1])/(nB[i]+sum(widows[-1]))
xi=tail(xis,1)
lambda=tail(lambdas,1)
loglikec<-logfactorial(sum(widows))-sum(sapply(widows,
                                        logfactorial))+widows[1]*log(
                                          xi+(1-xi)*exp(-lambda))+(log(1-xi
                                              )-lambda)*sum(
                                            widows[-1])+log(lambda)*sum(
                                                sapply(2:length(widows),ini)
                                                )-sum(sapply(2:length(widows
                                                ),logfactni))

lc<-c(lc,loglikec)


while (abs(lc[i]-lc[i-1])>eps) {
  i=i+1
  # E-step:
  nA[i] = widows[1]*xi/(xi+(1-xi)*exp(-lambda))
  nB[i] = widows[1]-nA[i]

  # M-step:
  xis[i] <- nA[i]/sum(widows)
  lambdas[i] <- sum(children[-1]*widows[-1])/(nB[i]+sum(widows[-1]))


  xi=tail(xis,1)
```

```
    lambda=tail(lambdas,1)

    loglikec<-logfactorial(sum(widows))-sum(sapply(widows,
                                              logfactorial))+widows[1]*log(
                                                xi+(1-xi)*exp(-lambda))+(log(1-
                                                    xi)-lambda)*sum(
                                                  widows[-1])+log(lambda)*sum(
                                                      sapply(2:length(widows),
                                                      ini))-sum(sapply(2:length(
                                                      widows),logfactni))

    lc<-c(lc,loglikec)

  }

  results <- list(xis = xis, lambdas = lambdas, nA=nA, nB=nB,i=i,lc=lc)
  return(results)
}
```

A.4. Efectos aleatorios

```
a1 <- c(46,31,37,62,30)
a2 <- c(70,59)
a3 <- c(52,44,57,40,67,64,70)
a4 <- c(47,21,70,46,14)
a5 <- c(42,64,50,69,77,81,87)
a6 <- c(35,68,59,38,57,76,57,29,60)

a = list(a1,a2,a3,a4,a5,a6)
n = NULL
for(i in 1:6){

  n[i]=length(a[[i]])
}
N = sum(n)

yBari <- NULL
for(i in 1:6){
  yBari[i] <- mean(a[[i]])
}

yBar <- sum(a1,a2,a3,a4,a5,a6)/N

SSE_0 <- NULL
SSA_0 <- NULL
for(i in 1:6){
  squareA <- n[i]*(yBari[i]-yBar)^2
  SSA_0 <- c(SSA_0,squareA)
  for(j in 1:length(a[[i]])){
    squareE <- (a[[i]][j]-yBari[i])^2
```

```
    SSE_0 <- c(SSE_0,squareE)
  }

}
SSE_0 <- sum(SSE_0)
SSA_0 <- sum(SSA_0)

MSA <- SSA_0/5
MSE <- SSE_0/9

sigma2a_ANOVA = 5*(MSA-MSE)/(N-sum(n^2)/N)
sigma2e_ANOVA = MSE




VcEM <- function(mu0,sig2a0,sig2e0){
  w0 <- c(0,0,0,0,0,0)
  v0 <- c(0,0,0,0,0,0)
  T10 <- c(0,0,0,0,0,0)
  T20 <- c(0,0,0,0,0,0)
  T30 <- c(0,0,0,0,0,0)

  SSA = array(0,6)
  for(i in 1:6){
    ni = n[i]
    for(j in 1:ni){
      SSA[i] = SSA[i] + (a[[i]][j]-yBari[i])^2}
    }


  for(i in 1:6){
    w0[i] <- sig2e0/(sig2e0+n[i]*sig2a0)
    v0[i] <- w0[i]*sig2a0
    T10[i] <- w0[i]*mu0+(1-w0[i])*yBari[i]
    T20[i] <- T10[i] + v0[i]
    T30[i] <- SSA[i] + n[i]*(w0[i]^2*(mu0-yBari[i])^2+v0[i])

  }

  T1_0 = sum(T10)
  T2_0 = sum(T20)
  T3_0 = sum(T30)


  mus <- array(0,10)
  sig2a <- array(0,10)
  sig2e <- array(0,10)
  # M-step:
  mus[1] = T1_0/6
```

```
  sig2a[1] = T2_0/6 - mus[1]^2
  sig2e[1] = T3_0/N

  w <- vector("list", 10)
  v <- vector("list", 10)

  T1 <- vector("list", 10)
  T2 <- vector("list", 10)
  T3 <- vector("list", 10)

  E1 <- array(0,10)
  E2 <- array(0,10)
  E3 <- array(0,10)


  for(k in 1:10){
    w[[k]] = array(0,6)
    v[[k]] = array(0,6)
    T1[[k]] = array(0,6)
    T2[[k]] = array(0,6)
    T3[[k]] = array(0,6)


  }

  for (t in 1:9) {
    for(i in 1:6){
      w[[t]][i] = sig2e[t]/(sig2e[t]+n[i]*sig2a[t])
      v[[t]][i] = w[[t]][i]*sig2a[t]

      T1[[t]][i] = w[[t]][i]*mus[t]+(1-w[[t]][i])*yBari[i]
      T2[[t]][i] = T1[[t]][i]^2 + v[[t]][i]
      T3[[t]][i] = SSA[i] + n[i]*(w[[t]][i]^2*(mus[t]-yBari[i])^2+v[[t]][i])
    }
    # E-step:
    E1[t] = sum(T1[[t]])
    E2[t] = sum(T2[[t]])
    E3[t] = sum(T3[[t]])

    # M-step:(t+1)
    mus[t+1] = E1[t]/6
    sig2a[t+1] = E2[t]/6-mus[t+1]^2
    sig2e[t+1] = E3[t]/N
  }
  results = list(mus = mus, siga = sig2a, sige = sig2e)
  return(results)

}
```

A.5. Población Multinomial EMMC

```
multMCEM<-function(y1,y2,y3,m,iter,theta0){

  theta=theta0
  z<-rbinom(m,y3,(theta/(2+theta)))
  y=mean(z)
  theta=(y2+y)/(y1+y2+y)
  loglike<-y1*log(0.5-0.5*theta)+(y2+y)*log(theta)
  thetas=c(theta0)


  for(i in 1:iter){
    z<-rbinom(m,y3,(theta/(2+theta)))
    y=mean(z)
    theta=(y2+y)/(y1+y2+y)
    loglike<-c(loglike,y1*log(0.5-0.5*theta)+(y2+y)*log(theta))
    thetas=c(thetas,theta)
  }
  return(list(loglike=loglike,thetas=thetas))
}
```

A.6. Viudas StEM

```
ViudasStEM<-function(obs,xi0,lambda0,iters){
  xis<-xi0
  lambdas<-lambda0
  pA<-xi0
  pB<-(1-xi0)*exp(-lambda0)
  xA<-rbinom(1,obs[1],pA/(pA+pB))
  xB<-obs[1]-xA
  xi<-xA/sum(obs)
  lambda<-sum(1:(length(obs)-1)*obs[-1])/(xB+sum(obs[-1]))
  xis<-c(xis,xi)
  lambdas<-c(lambdas,lambda)
  for(i in 2:iters){
    pA<-xi
    pB<-(1-xi)*exp(-lambda)
    xA<-rbinom(1,obs[1],pA/(pA+pB))
    xB<-obs[1]-xA
    xi<-xA/sum(obs)
    lambda<-sum(1:(length(obs)-1)*obs[-1])/(xB+sum(obs[-1]))
    xis<-c(xis,xi)
    lambdas<-c(lambdas,lambda)
  }
  return(list(xis=xis,lambdas=lambdas))
}
```

A.7. Efectos aleatorios MCEM

```
toros<-rbind(c(46,31,37,62,30,0,0,0,0),c(70,59,0,0,0,0,0,0,0),c
    (52,44,57,40,67,64,70,0,0),
            c(47,21,70,46,14,0,0,0,0),c(42,64,50,69,77,81,87,0,0),
            c(35,68,59,38,57,76,57,29,60))


ni<-function(i){
  return(max(which(toros[i,]!=0)))
}


nis<-sapply(1:6,ni)
Normali<-function(sige2,siga2,mu,m){
  mus=(sige2*mu+nis*siga2*(apply(toros,1,sum)/nis))/(sige2+nis*siga2)
  sigs2=sige2*siga2/(sige2+nis*siga2)
  return(mapply(rnorm,rep(m,6),mus,sqrt(sigs2)))

}
LMEM<-function(toros,siga20,sige20,mu0,m,iters){

  mus=c(mu0)
  siges2=c(sige20)
  sigas2=c(siga20)
  sige2<-sige20
  siga2<-siga20
  mu<-mu0

  for(h in 1:iters){
    e<-Normali(sige2,siga2,mu,m)

    if(m==1){
      mu=sum(e)/6
      siga20=sum((e-mu)^2)/6
      sige2=0
      for(j in 1:6){
        for(i in 1:nis[j]){
          sige2=sige2+((toros[j,i]-e[j])^2)

        }
      }

    }
    else{
      mu=sum(apply(e,1,sum))/(6*m)

      siga2=sum(apply((e-mu)^2,1,sum))/(6*m)
      sige2=0
      for(k in 1:m){
        for(j in 1:6){
```

```
        for(i in 1:nis[j]){
          sige2=sige2+((toros[j,i]-e[k,j])^2)

        }
       }
     }
   }
   sige2=sige2/(m*sum(nis))

   mus=c(mus,mu)
   sigas2=c(sigas2,siga2)
   siges2=c(siges2,sige2)

 }
 return(list(mus=mus,sa2=sigas2,se2=siges2))
}
```

A.8. Clasificación de Estados Moduladores

```
steps=function(montos,tiempos,ome,alp,bet,lambd){
 n=length(montos)
 iden=numeric(n) #etiqueta actual a que mezcla pertenece cada obs
 data_eval=cbind(dexp(tiempos,lambd[1])*dgamma(montos,alp[1],bet[1]),dexp(
    tiempos,lambd[2])*dgamma(montos,alp[2],bet[2]))
 #eval_nor=data_eval*ome/rowSums(data_eval*ome) # densidad de pesos
    condicionada (2.47), pra calcular los pesos dim 100*3
 probs=cbind(ome[1]*data_eval[,1],ome[2]*data_eval[,2])
 eval_nor=probs/apply(probs,1,sum) #poterior probs
 lik=c(0,0)

 m1=NULL
 #t1=NULL
 m2=NULL
 #t2=NULL


 for(i in 1:n)
 {
   iden[i]=which(eval_nor[i,]==max(eval_nor[i,]))
   #iden[i]=sample(c(1,2,3),1,prob=eval_nor[i,])
   if(iden[i]==1){
     m1=c(m1,montos[i])
     #t1=c(t1,tiempos[i])
   }

   else{
     m2=c(m2,montos[i])
     #t2=c(t2,tiempos[i])
     #lik[3]=lik[3]+log(dgamma(data[i],alp[3],bet[3]))
   }
```

```
  }

  #fin paso E
  #paso M
  fitgm1 <- fitdist(m1, "gamma"); pargpo1<-summary(fitgm1)$estimate
  fitgm2 <- fitdist(m2, "gamma"); pargpo2<-summary(fitgm2)$estimate

  #fitgt1 <- fitdist(t1, "exp"); parexp1<-summary(fitgt1)$estimate
  #fitgt2 <- fitdist(t2, "exp"); parexp2<-summary(fitgt2)$estimate

  omegas=colSums(eval_nor)/n
  a1=c(pargpo1[1],pargpo2[1])
  b1=c(pargpo1[2],pargpo2[2])
  l1=c(sum(eval_nor[,1]/sum(tiempos*eval_nor[,1])),sum(eval_nor[,2]/sum(tiempos
     *eval_nor[,2])))
  #b1 = a1*colSums(eval_nor)/colSums(data*eval_nor)
  for(i in 1:n)
  {

    if(iden[i]==1){

      lik[1]=lik[1]+log(dgamma(montos[i],a1[1],b1[1]))+log(dexp(tiempos[i],l1
          [1]))+log(omegas[1])
    }

    else{

      lik[2]=lik[2]+log(dgamma(montos[i],a1[2],b1[2]))+log(dexp(tiempos[i],l1
          [2]))+log(omegas[2])
    }
  }
  like=sum(lik)
  return(list(o=omegas,a1=a1,b1=b1,l1=l1,l=like,class=iden))
}


EM_clas=function(num_iter,montos,tiempos,ome,alp,bet,lam){
  omegas=matrix(0,num_iter,2)
  alphas=matrix(0,num_iter,2)
  betas=matrix(0,num_iter,2)
  lambdas=matrix(0,num_iter,2)
  accuracy=array(0,num_iter)

  like=numeric(num_iter-1)
  accuracy[1] = 0
  omegas[1,]=ome
  alphas[1,]=alp
  betas[1,]=bet
  lambdas[1,]=lam

  for(i in 2:num_iter)
```

```
  {
    res=steps(montos,tiempos,omegas[i-1,],alphas[i-1,],betas[i-1,],lambdas[i
        -1,])
    omegas[i,]=res$o
    alphas[i,]=res$a1
    betas[i,]=res$b1
    lambdas[i,]=res$l1
    like[i-1]=res$l

    accuracy[i]=sum(as.numeric(res$class==clasificacion_original))/length(res$
        class)
  }


  return(list(om=omegas,al=alphas,be=betas,la=lambdas,l=like,acc=accuracy,class
      =res$class))
}
```

A.9. Método de Bisección para puentes de Markov

```
Rab<- function(a,b,Q,t){
  qa<- -Q[a,a]
  qb<- -Q[b,b]
  if(qa!=qb){return(Q[a,b]*((exp(-qa*t)-exp(-qb*t))/(qb-qa)))}
  else{return(t*exp(-qa*t))}
}



Bisection<-function(a,b,T1,T2,Q1){
  a=a
  b=b
  T1=T1
  T2=T2
  media1=NULL
  media2=NULL
  HT=T2-T1
  HT0=HT/2
  n=length(Q1[1,])
  estados=1:n
  Po=expm(Q1*HT0)
  if(HT0<0){
    HT=-T2+T1
    HT0=HT/2
    T1=T2
    T2=T1
  }
  if(length(a)!=1){
    a=a[,1][1]
  }
```

```
if(is.na(a)==T){
  a=b
}
if(a==b){
  C=estados[-a]
  probab=NULL
  opciones1=NULL
  probba=c(exp(Q1[a,a]*HT0)*exp(Q1[a,a]*HT0),
           exp(Q1[a,a]*HT0)*(Po[a,a]-exp(Q1[a,a]*HT0)),
           exp(Q1[a,a]*HT0)*(Po[a,a]-exp(Q1[a,a]*HT0)),
           (Po[a,a]-exp(Q1[a,a]*HT0))*(Po[a,a]-exp(Q1[a,a]*HT0)))
  probab=c(probab,probba)
  opciones1=list(c(0,0),c(0,2),c(2,0),c(2,2))
  for(i in C ){
    probbc=c(Rab(a,i,Q1,HT0)*Rab(i,a,Q1,HT0),
             Rab(a,i,Q1,HT0)*(Po[i,a]-Rab(i,a,Q1,HT0)),
             Rab(i,a,Q1,HT0)*(Po[a,i]-Rab(a,i,Q1,HT0)),
             (Po[a,i]-Rab(a,i,Q1,HT0))*(Po[i,a]-Rab(i,a,Q1,HT0)))
    probab=c(probab,probbc)
    opcionesc=list(c(1,1),c(1,2),c(2,1),c(2,2))
    opciones1=cbind(opciones1,opcionesc)
  }
  complete=c(a,C)
  opcion=sample(1:length(probab),size=1,prob=abs(probab)+0.00001)
  if(opcion%%4==0){
    estado=opcion/4
  }else{estado=floor(opcion/4)+1}
  if(opciones1[[opcion]][1]==0){
    estad1=c(a,a)
    tempos1=c(T1,T1+HT0)
    media1=cbind(estad1,tempos1)
    if(opciones1[[opcion]][2]==0){
      estad2=c(a,a)
      tempos2=c(T1+HT0,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      return(total)
    }
    if(opciones1[[opcion]][2]==1){
      if(-Q[a,a]>-Q[b,b]){time2=T1+HT0+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b
          ,b]-Q[a,a])}
      else if(-Q[b,b]>-Q[a,a]){time2=T2-rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[
          a,a]-Q[b,b])}
      else{time2=T1+HT0+HT0*runif(0,1)}
      estad2=c(complete[estado],a,a)
      tempos2=c(T1+HT0,time2,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      print('No␣aplica␣caso␣(0,1)␣para␣a=b')
      return(total)
    }
```

```
    if(opciones1[[opcion]][2]==2){
      T1=T1+HT0
      T2=T1+HT0
      a=a
      b=b
      media2=Bisection(a,b,T1,T2,Q1)
      total=rbind(media1,media2)
      return(total)
    }
  }else if(opciones1[[opcion]][1]==1){
    if(-Q[a,a]>-Q[b,b]){time1=T1+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b,b]-Q[
        a,a])}
    else if(-Q[b,b]>-Q[a,a]){time1=T1+HT0-rtrunc(1,spec="exp",a=0,b=HT0,rate=
        Q[a,a]-Q[b,b])}
    else{time1=T1+HT0*runif(0,1)}
    estad1=c(a,complete[estado],complete[estado])
    tempos1=c(T1,time1,T1+HT0)
    media1=cbind(estad1,tempos1)


    if(opciones1[[opcion]][2]==0){
      estad2=c(complete[estado],a)
      tempos2=c(T1+HT0,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      print('No␣aplica␣caso␣(1,0)␣para␣a=b')
      return(total)
    }
    if(opciones1[[opcion]][2]==1){
      if(-Q[a,a]>-Q[b,b]){time2=T1+HT0+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b
          ,b]-Q[a,a])}
      else if(-Q[b,b]>-Q[a,a]){time2=T2-rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[
          a,a]-Q[b,b])}
      else{time2=T1+HT0+HT0*runif(0,1)}
      estad2=c(complete[estado],a,a)
      T2=T1+2*HT0
      tempos2=c(T1+HT0,time2,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      return(total)
    }
    if(opciones1[[opcion]][2]==2){
      T1=T1+HT0
      T2=T1+HT0
      a=complete[estado]
      media2=Bisection(a,b,T1,T2,Q1)
      media2=rbind(media1,media2)
      return(media2)
    }
  }else if(opciones1[[opcion]][1]==2){
    T1=T1
```

```
    b=complete[estado]
    media1=Bisection(a,b,T1,T1+HT0,Q1)
    if(opciones1[[opcion]][2]==0){
      estad2=c(a,a)
      T2=T1+2*HT0
      tempos2=c(T1+HT0,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      return(total)
    }
    if(opciones1[[opcion]][2]==1){
      T2=T1+2*HT0
      if(-Q[a,a]>-Q[b,b]){time2=T1+HT0+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b
          ,b]-Q[a,a])}
      else if(-Q[b,b]>-Q[a,a]){time2=T2-rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[
          a,a]-Q[b,b])}
      else{time2=T1+HT0+HT0*runif(0,1)}
      estad2=c(complete[estado],a,a)
      T2=T1+2*HT0
      tempos2=c(T1+HT0,time2,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      return(total)
    }
    if(opciones1[[opcion]][2]==2){
      T1=T1+HT0
      T2=T1+HT0
      a=complete[estado]
      media2=Bisection(a,b,T1,T2,Q1)
      media2=rbind(media1,media2)
      b=tail(media2[,1],1)
      return(media2)
    }
  }
}else{

  estados=1:n
  C=estados[-c(a,b)]
  probab=NULL
  opciones1=NULL
  opciones2=NULL
  probb=c(exp(Q1[a,a]*HT0)*Rab(a,b,Q1,HT0),
          exp(Q1[a,a]*HT0)*(Po[a,b]-Rab(a,b,Q1,HT0)),
          Rab(a,b,Q1,HT0)*(Po[a,a]-exp(Q1[a,a]*HT0)),
          (Po[a,a]-exp(Q1[a,a]*HT0))*(Po[a,b]-Rab(a,b,Q1,HT0)))
  probab=c(probab,probb)
  opciones1=list(c(0,1),c(0,2),c(2,1),c(2,2))

  probb2=c(exp(Q1[b,b]*HT0)*Rab(a,b,Q1,HT0),
           Rab(a,b,Q1,HT0)*(Po[b,b]-exp(Q1[b,b]*HT0)),
           exp(Q1[b,b]*HT0)*(Po[a,b]-Rab(a,b,Q1,HT0)),
```

```
        (Po[b,b]-exp(Q1[b,b]*HT0))*(Po[a,b]-Rab(a,b,Q1,HT0)))
probab=c(probab,probb2)
opciones2=list(c(1,0),c(1,2),c(2,0),c(2,2))
opciones1=cbind(opciones1,opciones2)

for(i in C ){
  probbc=c(Rab(a,i,Q1,HT0)*Rab(i,a,Q1,HT0),
           Rab(a,i,Q1,HT0)*(Po[i,b]-Rab(i,b,Q1,HT0)),
           Rab(i,b,Q1,HT0)*(Po[a,i]-Rab(a,i,Q1,HT0)),
           (Po[a,i]-Rab(a,i,Q1,HT0))*(Po[i,a]-Rab(i,a,Q1,HT0)))
  probab=c(probab,probbc)
  opcionesc=list(c(1,1),c(1,2),c(2,1),c(2,2))
  opciones1=cbind(opciones1,opcionesc)
}
complete=c(a,b,C)
opcion=sample(1:length(probab),size=1,prob=abs(probab))

if(opcion%%4==0){
  estado=opcion/4
}else{estado=floor(opcion/4)+1}


if(opciones1[[opcion]][1]==0){
  estad1=c(a,a)
  tempos1=c(T1,T1+HT0)
  media1=cbind(estad1,tempos1)
  if(opciones1[[opcion]][2]==0){
    estad2=c(a,a)
    tempos2=c(T1+HT0,T2)
    media2=cbind(estad2,tempos2)
    total=rbind(media1,media2)
    print('No␣aplica␣caso␣(0,0)␣con␣a!=b')
    return(total)
  }
  if(opciones1[[opcion]][2]==1){
    if(-Q[a,a]>-Q[b,b]){time2=T1+HT0+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b
        ,b]-Q[a,a])}
    else if(-Q[b,b]>-Q[a,a]){time2=T2-rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[
        a,a]-Q[b,b])}
    else{time2=T1+HT0+HT0*runif(0,1)}
    estad2=c(complete[estado],b,b)
    tempos2=c(T1+HT0,time2,T2)
    media2=cbind(estad2,tempos2)
    total=rbind(media1,media2)
    return(total)
  }
  if(opciones1[[opcion]][2]==2){
    T1=T1+HT0
    a=a
    b=b
    media2=Bisection(a,b,T1,T2,Q1)
```

```
      total=rbind(media1,media2)
      return(total)
    }
  }else if(opciones1[[opcion]][1]==1){
    if(-Q[a,a]>-Q[b,b]){time1=T1+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b,b]-Q[
        a,a])}
    else if(-Q[b,b]>-Q[a,a]){time1=T1+HT0-rtrunc(1,spec="exp",a=0,b=HT0,rate=
        Q[a,a]-Q[b,b])}
    else{time1=T1+HT0*runif(0,1)}
    estad1=c(a,complete[estado],complete[estado])
    tempos1=c(T1,time1,T1+HT0)
    media1=cbind(estad1,tempos1)
    T2=T1+2*HT0
    if(opciones1[[opcion]][2]==0){
      estad2=c(b,b)
      tempos2=c(T1+HT0,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      return(total)
    }
    if(opciones1[[opcion]][2]==1){
      if(-Q[a,a]>-Q[b,b]){time2=T1+HT0+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b
          ,b]-Q[a,a])}
      else if(-Q[b,b]>-Q[a,a]){time2=T2-rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[
          a,a]-Q[b,b])}
      else{time2=T1+HT0+HT0*runif(0,1)}
      estad2=c(complete[estado],b,b)
      tempos2=c(T1+HT0,time2,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      return(total)
    }
    if(opciones1[[opcion]][2]==2){
      T1=T1+HT0
      a=complete[estado]
      media2=Bisection(a,b,T1,T2,Q1)
      total=rbind(media1,media2)
      return(total)

    }
  }else if(opciones1[[opcion]][1]==2){
    T1=T1
    T2=T1+HT0
    media1=Bisection(a,complete[estado],T1,T2,Q1)
    T2=T1+2*HT0

    if(opciones1[[opcion]][2]==0){
      estad2=c(b,b)
      tempos2=c(T1+HT0,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
```

```
      return(total)
    }
    if(opciones1[[opcion]][2]==1){
      if(-Q[a,a]>-Q[b,b]){time2=T1+HT0+rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[b
          ,b]-Q[a,a])}
      else if(-Q[b,b]>-Q[a,a]){time2=T2-rtrunc(1,spec="exp",a=0,b=HT0,rate=Q[
          a,a]-Q[b,b])}
      else{time2=T1+HT0+HT0*runif(0,1)}
      estad2=c(complete[estado],b,b)
      tempos2=c(T1+HT0,time2,T2)
      media2=cbind(estad2,tempos2)
      total=rbind(media1,media2)
      return(total)
    }
    if(opciones1[[opcion]][2]==2){
      T1=T1+HT0
      T2=T1+HT0
      a=complete[estado]
      b=b
      media2=Bisection(a,b,T1,T2,Q1)
      total=rbind(media1,media2)
      return(total)
    }
  }
 }
}
```

A.10. Estimación de matriz de intensidades

```
tray=trayectoria
estados=1:4
estimar_matriz<-function(tray,estados){
  n=length(tray[,1])
  P=matrix(rep(0,length(estados)^2),length(estados),length(estados))
  Tx=rep(0,length(estados))
  V=rep(0,length(estados))
  for(i in 1:n-1){
    P[tray[i,2],tray[i+1,2]]=P[tray[i,2],tray[i+1,2]]+1
    Tx[tray[i,2]]=Tx[tray[i,2]]+(tray[i+1,1]-tray[i,1])
    V[tray[i,2]]=V[tray[i,2]]+1
  }
  diag=V/Tx
  for(i in estados){
    P[i,]=P[i,]/Tx[i]
    P[i,i]=-diag[i]
  }
  return(P)
}
clase<-prueba1$class
Ts<-sapply(1:4,function(x) sum(tiempos[which(clase==x)+1],na.rm=TRUE))
```

```
visitas<-clase[1]
for(i in 2:length(clase)){
  if(clase[i]==clase[i-1]){visitas<-visitas}
  else{visitas<-c(visitas,clase[i])}
}
V<-sapply(1:4, function(x) sum(as.numeric(visitas==x)))

visitasij<-matrix(rep(0,16),4,4)
for(i in 2:length(clase)){
  if(clase[i]==clase[i-1]){visitas<-visitas}
  else{visitas<-c(visitas,clase[i])}
}

Pij<-function(i,j){
  pij<-0
  en_i=which(clase==i)
  for(k in en_i){
    if(clase[k+1]==j & k!=10000){pij<-pij+1}
  }
  return(pij)
}

P<-rbind(sapply(1:4,Pij,i=1),sapply(1:4,Pij,i=2),sapply(1:4,Pij,i=3),sapply
    (1:4,Pij,i=4))
Q=matrix(rep(0,16),4,4)
for(i in 1:4){
  Q[i,]<-(V[i]/Ts[i])*P[i,]/V[i]
  Q[i,i]=-V[i]/Ts[i]
}

q1<-NULL
q2<-NULL
q3<-NULL
q4<-NULL
for(kk in 1:50){
  edos<-c()
  tiemps<-c()
  for (i in 400:500) {
    puente<-Bisection(prueba1$class[i],prueba1$class[i+1],tiemposacum[i-1],
        tiemposacum[i],Q)
    tiemps<-c(tiemps,puente[,2])
    edos<-c(edos,puente[,1])
  }
  trayectoria<-unique(cbind(tiemps,edos))
  trayectoria<-as.data.frame(trayectoria)
  trayectoria=trayectoria[mixedorder(trayectoria$tiemps),]
  Q<-estimar_matriz(trayectoria,1:4)
  q1<-c(q1,-Q[1,1])
  q2<-c(q2,-Q[2,2])
  q3<-c(q3,-Q[3,3])
```

```
  q4<-c(q4,-Q[4,4])


}
```

A.11. Método de Aceleración de Aitken para Población Multinomial

```
MultinomialEMaitken <- function(y_obs,MLEobs,theta0,tolerancia){

  thetasem <- NULL
  thetasacc<- NULL


  i = 1
  y3 <- 125*0.25*theta0/(0.5+0.25*theta0)
  thetasem[i] <- (y_obs[2] + y3)/(y_obs[1]+y_obs[2]+y3)
  Ic=(y_obs[2]+(125*theta0/(2+theta0)))/(theta0^2)+y_obs[1]/((1-theta0)^2)
  Im=theta0^(-2)*(125)*(theta0/(2+theta0))*(2/(2+theta0))
  thetasacc[i]=theta0+(Ic-Im)^(-1)*Ic*(thetasem[i]-theta0)

  i=2
  y3 <- 125*0.25*thetasacc[i-1]/(0.5+0.25*thetasacc[i-1])
  thetasem[i] <- (y_obs[2] + y3)/(y_obs[1]+y_obs[2]+y3)
  Ic=(y_obs[2]+(125*thetasacc[i-1]/(2+thetasacc[i-1])))/(thetasacc[i-1]^2)+y_
     obs[1]/((1-thetasacc[i-1])^2)
  Im=thetasacc[i-1]^(-2)*(125)*(thetasacc[i-1]/(2+thetasacc[i-1]))*(2/(2+
     thetasacc[i-1]))
  thetasacc[i]=thetasacc[i-1]+(Ic-Im)^(-1)*Ic*(thetasem[i]-thetasacc[i-1])



  while (abs(thetasacc[i]-MLEobs) > tolerancia) {
    i = i+1
    y3 <- 125*0.25*thetasacc[i-1]/(0.5+0.25*thetasacc[i-1])
    thetasem[i] <- (y_obs[2] + y3)/(y_obs[1]+y_obs[2]+y3)
    Ic=(y_obs[2]+(125*thetasacc[i-1]/(2+thetasacc[i-1])))/(thetasacc[i-1]^2)+y_
       obs[1]/((1-thetasacc[i-1])^2)
    Im=thetasacc[i-1]^(-2)*(125)*(thetasacc[i-1]/(2+thetasacc[i-1]))*(2/(2+
       thetasacc[i-1]))
    thetasacc[i]=thetasacc[i-1]+(Ic-Im)^(-1)*Ic*(thetasem[i]-thetasacc[i-1])

  }
  return(list(thetasacc = thetasacc,thetasem=thetasem))
}
```

Bibliografía

Amari, S. (1995). The EM algorithm and information geometry in neural network learning. *Neural Computation.*

Asmussen, S. (1989). Risk theory in a markovian environment. Scandinavian Actuarial Journal 1989, 2, pp. 69-100.

Asmussen, S.; Hobolth, A. (2012) Markov bridges, bisection and variance reduction. Monte Carlo and Quasi-Monte Carlo Methods 2010, l. Plaskota and H. Wozniakowski, Eds., vol. 23 of Springer Proceedings in Mathematics and Statistics, Springer, pp. 3-22.

Baum, L.E.; Petrie„ T. (1966) Statistical inference for probabilistic functions of finite Markov Chains. *Annals of Mathematical Statistics*

Bilmes, J.A. (1998) A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian mixture and Hidden Markov Models.

Blight, B. (1970). Estimation from a Censored Sample for the Exponential Family. Biometrika, 57(2), 389-395.

Boyles, R.A. (1983). On the convergence of the EM Algorithm. Journal of the Royal Statistical Society B 45, 47-50.

Celeux, G.; Diebolt, J. (1992). A stochastic approximationtype EM algorithm for the mixture problem. Stochastics and Stochastic Reports 41, 119-134.

Buck, S.(1960). A Method of Estimation of Missing Values in Multivariate Data Suitable for use with and Electronic Computer. Journal of the Royal Statistical Society. Series B (Methodological), 22(2), 302-306.

Delyon, B; M. Lavielle; E. Moulines. (1999). Convergence of a Stochastic Approximation Versioin of the EM Algorithm. The Annals of Statistics, Vol. 27, no. 1, pp. 94-128.

Diebolt, J; Ip, E.H.S. (1996). Stochastic EM: method and applcation. Markov Chain Monte Carlo in Practice, W.R. Gilks, S. Richardson, D.J. Spiegelhalter (Eds.). London: Chapman and Hall, pp.259-273.

Dempster,A.P.;Laird, N.M.;Rubin,D.B.(1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the royal Statistical Society. Series B (Methodological), Vol. 39, No. 1 1., pp. 1-38

Efron, B.(1967). The two sample probem with censored data. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 4, pp. 831-852.

Hartley, H.(1958). Maximum Likelihood Estimation from Incomplete Data. Biometrics, 14(2), 174-194.

McLachlan, G.J.; T. Krishnan (2007). The EM Algorithm and Extensions. John Wiley and Sons.

Louis, T.A.(1982). Finding the observed information matrix when using the EM algorithm. Journal of the Royal Statistical Society, Series B, Vol. 44, pp 226-233.

Meng, X.L.; Rubin, D.B.(1989). Obtaining asymptotic variance-covariance matrices for missing-data problems using EM. American Statistical Association Proceedings of the Statistical Comuting Section. Alexandria, VA: American Statistical Association, pp. 140-144.

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

Meng, X.L.; Rubin, D.B.(1994). On the global and componentwise rates of convergence of the EM algorithm. Linear Algebra and its Applications 199, 413-425.

Nielsen, S.F. (2000). the Stochastic EM Algorithm: Estimation and Asymptotic Results. Bernoulli, Vol. 6, No. 3, pp. 457-489

Orchard, T.; Woodbury, M.A. (1972). A missing information principle: Theory and applications. Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1. Berkeley, CA: University of California Press, pp. 697-715.

Ostrowski, A.M.(1966). Solution of Equations and Systems of Equations. Second Edition. New York: Academic Press.

Reinhard, J.-M.(1984). On a class of semi-markov risk models obtained as classical risk models in a markovian environment. ASTIN Bulletin 14, 1 pp. 23-43.

Snedecor, G.W., Cochran, W.G. (1967). Statistical Methods. Iowa State University Press, pp 290.

Sundberg, R. (1974). Maximum likelihood theory for incomplete data from an exponential family. Scandinavian Journal of Statistics: Theory and Applications 1, 49-58.

Sundberg, R.(1976). An iterative method for solution of the likelihood equations for incomplete data from exponential families. Communications in Statistics-Simulation and Communication 5, 55-64.

Thisted, R.A. (1988). Elements of Statistical Computing: Numerical Computation. Chapman and Hall.

Turnbull, B.W. (1976). The empirical distribution function with arbitrary grouped, censored and truncated data. J. Royal Statistics Society. Series B, 38, pp. 290-295.

Searle, S.R.; Casella, G.; McCulloch, C.E. (2009). Variance Components. John Wiley and Sons.

Wei, G.C.G.; Tanner, M.A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. Journal of the American statistical Association 85, 699-704.

Wu, C.F.J.(1983). On the convergence properties of the EM algorithm. Annals of Statistics 11, 95-103.

Zangwill, W.I. (1969). Nonlinear Programming: A Unified Approach. Englewood Cliffs, NJ: Prentice-Hall.