Estadística Bayesiana

Métodos de Simulación Estocástica

Lizbeth Naranjo Albarrán

Facultad de Ciencias, UNAM

March 25, 2022

Índice

Capítulo 1

Métodos de Simulación Estocástica en Estadística Bayesiana

En este capítulo estudiaremos algunos métodos de simulación estocástica, necesarios para la estimación Bayesiana, usaremos principalmente métodos de Monte Carlo vía cadenas de Markov, y daremos damos una breve descripción de WinBUGS, JAGS y STAN, y finalmente trataremos de usar métodos INLA.

Hay varios libros que pueden ayudar en estos temas, algunos de los que se usarán son Chen et al. (2000), Gamerman and Lopes (2006), Gilks et al. (1996) y Robert and Casella (2000), entre otros.

1.1 Integración Monte Carlo

Como se mencionó anteriormente, en el enfoque Bayesiano de la Estadística, la incertidumbre presente en un modelo dado, $p(x|\theta)$, se representa a través de una distribución de probabilidad $p(\theta)$ sobre el espacio parametral Θ (generalmente multidimensional) que define al modelo. El teorema de Bayes,

$$p(\theta|x) = \frac{p(\theta)p(x|\theta)}{p(x)},$$

permite entonces incorporar la información contenida en un conjunto de datos $x = (x_1, \ldots, x_n)$, produciendo una descripción conjunta de la incertidumbre sobre los valores de los parámetros del modelo a través de la distribución final $p(\theta|x)$. Desafortunadamente, la implementación de los métodos Bayesianos ocasionalmente requieren de un esfuerzo computacional muy alto. La mayor parte de este esfuerzo se concentra en el cálculo de ciertas características de la distribución final del parámetro de interés. Además, en la práctica es común que la dimensión de θ sea muy grande. Por otro lado, excepto en aplicaciones muy sencillas tanto $p(x|\theta)$ como

$p(\theta)$ pueden llegar a tener formas muy complicadas. En la mayoría de los problemas las integrales requeridas no pueden resolverse analíticamente, por lo que es necesario contar con métodos numéricos eficientes que permitan calcular o aproximar integrales en varias dimensiones. Algunos de estos métodos de integración son las técnicas de Monte Carlo vía cadenas de Markov. Para mayores detalles consultar Gilks, Richardson y Spiegelhalter (1996).

La integración de Monte Carlo evalúa $E[g(\theta)]$ obteniendo muestras denotadas por $\{\theta^{(t)},\ t = 1, \ldots, n\}$ de la distribución $p(\theta|x)$ y entonces aproximando

$$E[g(\theta)] \approx \frac{1}{n} \sum_{t=1}^{n} g(\theta^{(t)}).$$

De esta manera, la media poblacional de $g(\theta)$ se estima por medio de una media muestral. Cuando las muestras $\{\theta^{(t)}\}$ son independientes, la ley de los grandes números asegura que la aproximación puede hacerse tan precisa como se desee incrementando el tamaño de la muestra, n. Note que aquí n está bajo el control del analista: no es el tamaño fijo para una muestra de datos.

En general, seleccionar muestras $\{\theta^{(t)}\}$ independientes de $p(\theta|x)$ no es factible, ya que $p(\theta|x)$ puede ser no estándar. Sin embargo, las $\{\theta^{(t)}\}$ no necesariamente necesitan ser independientes. Las $\{\theta^{(t)}\}$ pueden ser generadas por cualquier proceso que selecciona muestras por todo el soporte de $p(\theta|x)$ en las proporciones correctas. Una forma de hacer esto es a través de una cadena de Markov teniendo a $p(\theta|x)$ como su distribuión estacionaria. Esto es entonces un método de Monte Carlo vía cadenas de Markov.

En resumen, las técnicas de Monte Carlo vía cadenas de Markov permiten generar, de manera iterativa, observaciones de distribuciones multivariadas que difícilmente podrían simularse utilizando métodos directos. La idea básica es muy simple: construir una cadena de Markov que sea fácil de simular y cuya distribución de equilibrio corresponda a la distribución final que nos interesa.

Proposición 1.1. Sea $\theta^{(1)}, \theta^{(2)}, \ldots$ una cadena de Markov homogénea, irreducible y aperiódica, con espacio de estados Θ y distribución de equilibrio $p(\theta|x)$. Entonces, conforme $t \to \infty$,

$$(i) \quad \theta^{(t)} \xrightarrow{\mathcal{D}} \theta, \qquad \textit{donde } \theta \sim p(\theta|x);$$
$$(ii) \quad \frac{1}{t} \sum_{i=1}^{t} g(\theta^{(i)}) \to E(g(\theta)|x).$$

`Ejemplos:` Bayes7_1IntegracionMonteCarlo.pdf
`Código R:` Bayes7_1IntegracionMonteCarlo.R

Ejemplo 1.1 (Normal)**.** Dada una muestra de tamaño n de una distribución $Normal(0,1)$, $x_1, \ldots, x_n$, y considere la función de distribución acumulada $\Phi(t)$ igual a:

$$\Phi(t) \;=\; \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x$$

Por medio del método de Monte Carlo se puede obtener una aproximación como:

$$\widehat{\Phi}(t) \;\approx\; \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[x_i \leq t]$$

con varianza exacta

$$\frac{\Phi(t)[1-\Phi(t)]}{n}$$

¿Cómo se obtiene?

La integración Monte Carlo permite calcular la experanza poblacional por medio de una media muestral:

$$E[g(X)] \approx \frac{1}{n}\sum_{t=1}^{n} g(X^{(t)}).$$

Entonces, sea $X \sim Normal(0,1)$, y defina la función g como $g(X) = \mathbb{I}[X \leq t]$, entonces, la media poblacional de $g(X)$ es:

$$\begin{aligned} E[g(X)] \qquad &= \int g(x)f(x)\mathrm{d}x \qquad = \int \mathbb{I}[x \leq t] f(x)\mathrm{d}x \\ &= \int \mathbb{I}[x \leq t] \tfrac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x \;= \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x \quad = \Phi(t) \end{aligned}$$

Entonces se obtiene una muestra de tamaño n de una distribución $Normal(0,1)$, $x_1, \ldots, x_n$, y se calcula $g(x_1) = \mathbb{I}[x_1 \leq t], \ldots, g(x_n) = \mathbb{I}[x_n \leq t]$, y se calcula la media muestral:

$$\frac{1}{n}\sum_{t=1}^{n} g(X^{(t)}) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{I}[x_i \leq t]$$

Entonces, usando la integración Monte Carlo,

$$\Phi(t) \;\approx\; \frac{1}{n}\sum_{t=1}^{n} \mathbb{I}[x_i \leq t]$$

Además cada una de estas funciones $g(x) = \mathbb{I}[x \leq t]$ valen 0 o 1, es decir, que son v.a. Bernoulli con probabilidad de éxito $\Phi(t)$, porque

$$P(\mathbb{I}[X \leq t] = 1) \;=\; P(X \leq t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x \quad = \Phi(t)$$

■

1.1.1 Muestreo por Importancia

El muestreo por Importancia se basa en una representación alternativa de

$$E_f\left[h(x)\right] = \int_{\mathcal{X}} h(x) f_X(x) \mathrm{d}x$$

Dada una función de densidad g, estrictamente positiva cuando $h(x)f_X(x) \neq 0$, se reescribe la esperanza,

$$\begin{aligned} E_f\left[h(x)\right] &= \int_{\mathcal{X}} h(x) f_X(x) \frac{g(x)}{g(x)} \mathrm{d}x \\ &= \int_{\mathcal{X}} h(x) \frac{f_X(x)}{g(x)} g(x) \mathrm{d}x \\ &= E_g\left[h(x) \frac{f_X(x)}{g(x)}\right] \end{aligned}$$

donde $\mathcal{X}$ es el soporte de X, es decir, el conjunto donde X toma sus valores bajo f_X y h, puede ser más pequeño que el soporte de la densidad g, es decir,

$$\sup(hf_X) \subseteq \sup(g)$$

La restricción del soporte se debe cumplir, porque de lo contrario se podría truncar el suporte, y producir resultados insesgados.

El estimador de $E_f\left[h(x)\right]$ usando la integración Monte Carlo:

$$E_f\left[h(x)\right] \approx \frac{1}{n} \sum_{i=1}^{n} h(x_i) \frac{f_X(x_i)}{g(x_i)}$$

donde $x_1, \ldots, x_n$ es una muestra generada a partir de g.

Ejemplo 1.2. Calcular $E[h(X)]$, donde $X \sim Uniforme(0, 10)$ y $h(X) = 10\exp(-2|x-5|)$. Usando: (a) integración Monte Carlo y (b) muestreo por importancia. ■

Ejemplo 1.3. Estimar los momentos $E[X^k]$ donde X se distribuye Doble Exponencial o Laplace:

$$\begin{aligned} f(x) &= \tfrac{1}{2}\exp(-|x|) \\ F(x) &= \exp(x)/2 I[x \leq 0] + (1 - \exp(-x)/2) I[x > 0] \end{aligned}$$

Opción 1: simulando de Doble Exponencial y usando Integración Monte Carlo.
Opción 2: simulando de $N(0, 4)$ y usando muestreo por Importancia. ■

Ejemplo 1.4. Sea $Z \sim Normal(0,1)$, calcular $P[Z > 4.5] = 3.398 \times 10^{-6}$

Esta probabilidad es demasiado pequeña y el método de Monte Carlo estándard se rompe/descompone cuando se tienen que calcular probabilidades de las colas. Se requiere un número grande de simulaciones.

Usando muestreo por Importancia, se considera el soporte restringido a $(4.5, \infty)$. Tomar las funciones g como la función de densidad $Exponencial(1)$, que es la selección 'natural', pero que esté truncada en 4.5,

$$g(X) = \frac{e^{-x}}{P[x \geq 4.5]} = \frac{e^{-x}}{\int_{4.5}^{\infty} e^{-x} \mathrm{d}x} = \frac{e^{-x}}{e^{-4.5}} = e^{-(x-4.5)}$$

El estimador por el muestreo por importancia es

$$\begin{aligned} P[X > 4.5] &= E[\mathbb{I}[X > 4.5]] = \int_{4.5}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \mathrm{d}x \\ &= \int_{4.5}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \frac{1}{e^{-(x-4.5)}} e^{-(x-4.5)} \mathrm{d}x \\ &\approx \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} \frac{1}{e^{-(x_i-4.5)}} \end{aligned}$$

donde $x_1, \ldots, x_n$ son v.a.i.i.d. $Exponencial(1)$. ■

`Código R:` Bayes7_1IntegracionMonteCarlo.R

1.1.2 Método de Aceptación y Rechazo

El método de aceptación y rechazo es un método indirecto para generar variables aleatorias: $f_X(\cdot)$ función objetivo (*target*) y $g_Y(\cdot)$ función de densidad candidato.

Restricciones:

- $f_X(\cdot)$ y $g_Y(\cdot)$ tienen soporter compatibles, $g_Y(x) > 0$ cuando $f_X(x) > 0$.
- Existe una constante M con $\frac{f_X(x)}{g_Y(x)} \leq M \ \forall x$.

Algoritmo de Aceptación y Rechazo para simular X:

(1) Generar $Y \sim g_Y(y)$.

(2) Generar $U \sim U(0,1)$, independiente de Y.

(3) Si $u \leq \frac{1}{M}\frac{f_X(y)}{g_Y(y)}$ entonces $X = y$. En caso contrario, descartar y, y regresando a (1).

¿Cómo funciona el algoritmo de Aceptación-Rechazo?
La función de densidad acumulada de la v.a. aceptada es:

$$\begin{aligned} P\left[Y \leq x | u \leq \frac{1}{M}\frac{f_X(y)}{g_Y(y)}\right] &= \frac{P\left[Y \leq x \ , \ U \leq \frac{1}{M}\frac{f_X(y)}{g_Y(y)}\right]}{P\left[U \leq \frac{1}{M}\frac{f_X(y)}{g_Y(y)}\right]} = \frac{\int_{-\infty}^{x}\int_{0}^{\frac{1}{M}\frac{f_X(y)}{g_Y(y)}} \mathrm{d}u g(y)\mathrm{d}y}{\int_{-\infty}^{\infty}\int_{0}^{\frac{1}{M}\frac{f_X(y)}{g_Y(y)}} \mathrm{d}u g_Y(y)\mathrm{d}y} \\ &= \frac{\int_{-\infty}^{x}\frac{1}{M}\frac{f_X(y)}{g_Y(y)}g_Y(y)\mathrm{d}y}{\int_{-\infty}^{\infty}\frac{1}{M}\frac{f_X(y)}{g_Y(y)}g_Y(y)\mathrm{d}y} = \frac{\int_{-\infty}^{x} f_X(y)\mathrm{d}y}{\int_{-\infty}^{\infty} f_X(y)\mathrm{d}y} \\ &= \int_{-\infty}^{x} f_X(y)\mathrm{d}y = P\left[X \leq x\right] \end{aligned}$$

Ejemplo 1.5. Sea $X \sim Beta(a,b)$ el target, usamos $U(0,1)$ como densidad candidata, y $M = \max\{f(x)\}$. ■

`Código R:` Bayes7_2DistribucionBeta.R

1.2 Monte Carlo vía Cadenas de Markov

En estadística Bayesiana, $P(X|\boldsymbol{\theta})$ y $P(\boldsymbol{\theta})$ pueden llegar a tener formas muy complicadas. En la mayoría de los problemas las integrales requeridas para calcular $P(\boldsymbol{X})$, $P(X^*)$ y $P(X^*|\boldsymbol{X})$ no pueden resolverse de manera analítica, por lo que es necesario usar métodos numéricos eficientes que permitan calcular o aproximar estas integrales. Algunos de estos métodos son las técnicas de Monte Carlo vía cadenas de Markov (*Markov chain Monte Carlo*, MCMC) (Gilks et al., 1996; Chen et al., 2000; Gamerman and Lopes, 2006).

La intergración de Monte Carlo evalúa la media de $g(\boldsymbol{\theta})$, $\mathrm{E}\left[g(\boldsymbol{\theta})\right]$, obteniendo muestras de la distribución $P(\boldsymbol{\theta}|\boldsymbol{X})$, denotadas por $\left\{\boldsymbol{\theta}^{(t)}; t = 1, \ldots, m\right\}$, y aproximando

$$\mathrm{E}\left[g(\boldsymbol{\theta})\right] \approx \frac{1}{m}\sum_{t=1}^{m} g\big(\boldsymbol{\theta}^{(t)}\big),$$

es decir, la media poblacional de $g(\boldsymbol{\theta})$ se estima por medio de la media muestral.

Las muestras $\left\{\boldsymbol{\theta}^{(t)}; t = 1, \ldots, m\right\}$ pueden ser generadas por cualquier proceso que selecciona muestras por todo el soporte de $P(\boldsymbol{\theta}|\boldsymbol{X})$ en las proporciones correctas. En particular, las muestras se pueden generar a través de una cadena de Markov teniendo a $P(\boldsymbol{\theta}|\boldsymbol{X})$ como su distribución estacionaria, y a este proceso se le conoce como método de Monte Carlo vía cadenas de Markov. Los dos algoritmos más comúnes en los métodos MCMC son muestreo de Gibbs y Metropolis-Hastings.

`Código R:` Bayes7_2DistribucionBeta.R

`Código R:` Bayes7_3DistribucionNormal.R

1.2.1 Algoritmo Metropolis-Hastings

El algoritmo de Metropolis-Hastings construye una cadena de Markov definiendo las probabilidades de transición de la siguiente manera.

Sea $Q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ una densidad de transición (arbitraria) y se define

$$\alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta}) \;=\; \min\left\{\frac{P(\boldsymbol{\theta}^*|\boldsymbol{X})Q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{P(\boldsymbol{\theta}|\boldsymbol{X})Q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}, 1\right\}.$$

Dado un valor inicial $\boldsymbol{\theta}^{(0)}$, la t-ésima iteración consta de:

1) Generar una observación $\boldsymbol{\theta}^*$ de $Q\big(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t)}\big)$.

2) Generar una variable $u \sim \mathrm{U}(0,1)$.

3) Si $\alpha\big(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(t)}\big) \geq u$, hacer $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^*$, en caso contrario $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$.

De esta manera se genera una cadena de Markov con distribución de transición

$$\begin{aligned} P\big(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}\big) \;&=\; \alpha\big(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t+1)}\big)Q\big(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}\big)\mathrm{I}\big[\boldsymbol{\theta}^{(t+1)} \neq \boldsymbol{\theta}^{(t)}\big] \\ &+\; \left[1 - \int \alpha\big(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^*\big)Q\big(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t)}\big)\mathrm{d}\boldsymbol{\theta}^*\right]\mathrm{I}\big[\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}\big]. \end{aligned}$$

La probabilidad de aceptación $\alpha\big(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^*\big)$ sólo depende de $P(\boldsymbol{\theta}|\boldsymbol{X})$ a través de un cociente, por lo que la constante de normalización no es necesaria.

Este algoritmo construye una cadena de Markov definiendo las probabilidades de transición de la siguiente manera.

Sea $Q(\theta^*|\theta)$ una densidad de transición (arbitraria) y definamos

$$\alpha(\theta^*, \theta) = \min\left\{\frac{p(\theta^*|x)Q(\theta|\theta^*)}{p(\theta|x)Q(\theta^*|\theta)}, 1\right\}.$$

Algoritmo:

Dado un valor inicial $\theta^{(0)}$, la t-ésima iteración consiste en:

1. generar una observación θ^* de $Q(\theta^*|\theta^{(t)})$;

2. generar una variable $u \sim U(0,1)$;

3. si $u \leq \alpha(\theta^*, \theta^{(t)})$, hacer $\theta^{(t+1)} = \theta^*$; en caso contrario, hacer $\theta^{(t+1)} = \theta^{(t)}$.

Este procedimiento genera una cadena de Markov con distribución de transición

$$\begin{aligned} p(\theta^{(t+1)}|\theta^{(t)}) &= \alpha(\theta^{(t)}, \theta^{(t+1)}) Q(\theta^{(t+1)}|\theta^{(t)}) I_{\{\theta^{(t+1)} \neq \theta^{(t)}\}} \\ &+ \left[1 - \int \alpha(\theta^{(t)}, \theta^*) Q(\theta^*|\theta^{(t)}) d\theta^*\right] I_{\{\theta^{(t+1)} = \theta^{(t)}\}}. \end{aligned}$$

La probabilidad de aceptación $\alpha(\theta^*, \theta)$ sólo depende de $p(\theta|x)$ a través de un cociente, por lo que la constante de normalización no es necesaria.

Comentario. La versión original del algoritmo de Metropolis-Hastings toma $Q(\theta^*|\theta) = Q(\theta|\theta^*)$, en cuyo caso

$$\alpha(\theta^*, \theta) = \min\left\{\frac{p(\theta^*|x)}{p(\theta|x)}, 1\right\}.$$

Dos casos particulares en la práctica son:

- *Caminata aleatoria.* Sea $Q(\theta^*|\theta) = Q_1(\theta^* - \theta)$, donde $Q_1(\cdot)$ es una densidad de probabilidad simétrica centrada en el origen. Entonces

$$\alpha(\theta^*, \theta) = \min\left\{\frac{p(\theta^*|x)}{p(\theta|x)}, 1\right\}.$$

- *Independencia.* Sea $Q(\theta^*|\theta) = Q_0(\theta^*)$, donde $Q_0(\cdot)$ es una densidad de probabilidad sobre Θ. Entonces

$$\alpha(\theta^*, \theta) = \min\left\{\frac{\omega(\theta^*)}{\omega(\theta)}, 1\right\},$$

con $\omega(\theta) = p(\theta|x)/Q_0(\theta)$.

En la práctica es común utilizar, después de una reparametrización apropiada, distribuciones de transición normales ó t de Student ligeramente sobredispersas, por ejemplo

$$Q(\theta^*|\theta) = N_d(\theta^*|\theta, \kappa V(\hat{\theta})) \qquad \text{(caminata aleatoria)}$$

ó

$$Q_0(\theta^*) = N_d(\theta^*|\hat{\theta}, \kappa V(\hat{\theta})) \qquad \text{(independencia)},$$

donde $\hat{\theta}$ y $V(\hat{\theta})$ denotan a la media y a la matriz de varianzas-covarianzas de la aproximación normal asintótica para $p(\theta|x)$, respectivamente, y $\kappa \geq 1$ es una factor de sobredispersión.

Ejemplo 1.6 (Coeficiente de correlación)**.** Suponga que $D = \{\mathbf{y}_i = (y_{1i}, y_{2i})^t, i = 1, \ldots, n\}$ es una muestra aleatoria de una distribución normal bivariada $N_2(0, \Sigma)$, donde

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Suponga una distribución inicial uniforme $U(-1, 1)$ para ρ; la densidad final para ρ está dada por

$$p(\rho|D) \propto (1-\rho^2)^{-n/2} \exp\left\{-\frac{1}{2(1-\rho^2)}\left(S_{11} - 2\rho S_{12} + S_{22}\right)\right\},$$

donde $-1 < \rho < 1$, y $S_{rs} = \sum_{i=1}^{n} y_{ri} y_{si}$ para $r, s = 1, 2$. Generar valores de ρ de la densidad $p(\rho|D)$ no es algo trivial, ya que no es una función log-cóncava. Por tanto, consideramos el algoritmo de Metropolis-Hastings para generar ρ. Como $-1 < \rho < 1$, consideremos la transformación

$$\rho = \frac{-1 + e^{\xi}}{1 + e^{\xi}}, \qquad -\infty < \xi < \infty.$$

Entonces

$$p(\xi|D) = p(\rho(\xi)|D)\frac{2e^{\xi}}{(1+e^{\xi})^2}.$$

En vez de obtener un muestreo directo de ρ, generamos ξ eligiendo una distribución proporcional a $N(\hat{\xi}, \hat{\xi}^2_{\hat{\xi}})$, donde $\hat{\xi}$ es el máximo del logaritmo de $p(\xi|D)$, que puede obtenerse mediante el algoritmo de Newton-Raphson o el algoritmo de Nelder-Mead, y $\hat{\sigma}^2_{\hat{\xi}}$ es menos el inverso de la segunda derivada del $\log p(\xi|D)$ evaluada en $\xi = \hat{\xi}$, es decir,

$$\hat{\sigma}^{-2}_{\xi} = -\left.\frac{d^2 \log p(\xi|D)}{d\xi^2}\right|_{\xi=\hat{\xi}}.$$

El algoritmo para generar ξ opera de la manera siguiente:

Dado un valor inicial $\xi^{(0)}$,

1. generar ξ^* de $N(\hat{\xi}, \hat{\sigma}^2_{\hat{\xi}})$;
2. generar una variable $u \sim U(0, 1)$;
3. si $u \leq \alpha(\xi^*, \xi^{(t)})$, entonces $\xi^{(t+1)} = \xi^*$; en caso contrario, $\xi^{(t+1)} = \xi^{(t)}$ donde

$$\alpha(\xi^*, \xi^{(t)}) = \min\left\{\frac{p(\xi^*|D)\phi\left(\frac{\xi^{(t)} - \hat{\xi}}{\hat{\sigma}_{\hat{\xi}}}\right)}{p(\xi^{(t)}|D)\phi\left(\frac{\xi^* - \hat{\xi}}{\hat{\sigma}_{\hat{\xi}}}\right)}, 1\right\}$$

   y ϕ es la función de densidad de probabilidad normal estándar.

Después de obtener ξ, podemos obtener ρ aplicando la transformación correspondiente. ■

1.2.2 Muestreo de Gibbs

El muestreo de Gibbs permite simular una cadena de Markov $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(m)}$ con distribución de equilibrio $P(\boldsymbol{\theta}|\boldsymbol{X})$. Cada nuevo valor de la cadena se puede obtener a través de un proceso iterativo que sólo requiere generar muestras de distribuciones cuya dimensión es menor que d (la dimensión de $\boldsymbol{\theta}$) y que en la mayoría de los casos tienen una forma más sencilla que la de $P(\boldsymbol{\theta}|\boldsymbol{X})$.

Sea $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$ una partición del vector $\boldsymbol{\theta}$, donde $\theta_i \in \mathbb{R}^{d_i}$ y $\sum_{i=1}^{k} d_i = d$. Se elige un valor inicial $\boldsymbol{\theta}^{(0)} = \left(\theta_1^{(0)}, \ldots, \theta_k^{(0)}\right)$, se simula una cadena de Markov en la que $\boldsymbol{\theta}^{(t+1)}$ se obtiene a partir de $\boldsymbol{\theta}^{(t)}$ de las distribuciones condicionales completas de la siguiente manera:

1) Generar $\theta_1^{(t+1)}$ de $P\left(\theta_1|\theta_2^{(t)}, \theta_3^{(t)}, \ldots, \theta_k^{(t)}, \boldsymbol{X}\right)$.

2) Generar $\theta_2^{(t+1)}$ de $P\left(\theta_2|\theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_k^{(t)}, \boldsymbol{X}\right)$.

$\vdots$

k) Generar $\theta_k^{(t+1)}$ de $P\left(\theta_k|\theta_1^{(t+1)}, \theta_2^{(t+1)}, \ldots, \theta_{k-1}^{(t+1)}, \boldsymbol{X}\right)$.

De esta manera se genera una cadena de Markov con distribución de transición

$$P\left(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}\right) \;=\; \prod_{i=1}^{k} P\left(\theta_i^{(t+1)}|\theta_1^{(t+1)}, \ldots, \theta_{i-1}^{(t+1)}, \theta_{i+1}^{(t)}, \ldots, \theta_k^{(t)}, \boldsymbol{X}\right).$$

En algunos casos la distribución final implica cierta estructura de independencia condicional entre algunos de los elementos de $\boldsymbol{\theta}$, y en ocasiones las distribuciones condicionales completas se simplifican.

Al igual que el algoritmo de Metropolis-Hastings, el algoritmo de Gibbs permite simular una cadena de Markov $\theta^{(1)}, \theta^{(2)}, \ldots$ con distribución de equilibrio $p(\theta|x)$. En este caso, sin embargo, cada nuevo valor de la cadena se puede obtener a través de un proceso iterativo que sólo requiere generar muestras de distribuciones cuya dimensión es menor que d y que en la mayoría de los casos tienen una forma más sencilla que la de $p(\theta|x)$.

Sea $\theta = (\theta_1, \ldots, \theta_k)$ una partición del vector θ, donde $\theta_i \in \Re^{d_i}$ y $\sum_{i=1}^{k} d_i = d$. Las densidades

$$\begin{gathered} p(\theta_1|\theta_2, \ldots, \theta_k, x) \\ \vdots \\ p(\theta_i|\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_k, x) \quad (i = 2, \ldots, k-1) \\ \vdots \\ p(\theta_k|\theta_1, \ldots, \theta_{k-1}, x) \end{gathered}$$

se conocen como *densidades condicionales completas* y en general pueden identificarse fácilmente al inspeccionar la forma de la distribución final $p(\theta|x)$. De hecho, para cada $i = 1, \ldots, k$,

$$p(\theta_i|\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_k, x) \propto p(\theta|x),$$

donde $p(\theta|x) = p(\theta_1, \ldots, \theta_k|x)$ es vista sólo como función de θ_i.

Algoritmo:

Dado un valor inicial $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_k^{(0)})$, el algoritmo de Gibbs simula una cadena de Markov en la que $\theta^{(t+1)}$ se obtiene a partir de $\theta^{(t)}$ de la siguiente manera:

generar una observación $\theta_1^{(t+1)}$ de $p(\theta_1|\theta_2^{(t)}, \theta_3^{(t)}, \ldots, \theta_k^{(t)}, x)$;

generar una observación $\theta_2^{(t+1)}$ de $p(\theta_2|\theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_k^{(t)}, x)$;

$$\vdots$$

generar una observación $\theta_k^{(t+1)}$ de $p(\theta_k|\theta_1^{(t+1)}, \theta_2^{(t+1)}, \ldots, \theta_{k-1}^{(t+1)}, x)$.

La sucesión $\theta^{(1)}, \theta^{(2)}, \ldots$ así obtenida es entonces una realización de una cadena de Markov cuya distribución de transición está dada por

$$p(\theta^{(t+1)}|\theta^{(t)}) = \prod_{i=1}^{k} p(\theta_i^{(t+1)}|\theta_1^{(t+1)}, \ldots, \theta_{i-1}^{(t+1)}, \theta_{i+1}^{(t)}, \ldots, \theta_k^{(t)}, x).$$

Comentario. En ocasiones la distribución final implica cierta estructura de independencia condicional entre algunos de los elementos del vector θ. En estos casos es común que muchas de las densidades condicionales completas se simplifiquen.

Ejemplo 1.7 (Gibbs: modelo jerárquico)**.** Consideremos el modelo jerárquico definido por

$$\begin{aligned} \text{I.} \quad & p(x|\omega) &&= \prod_{i=1}^{m} p(x_i|\omega_i); \\ \text{II.} \quad & p(\omega|\phi) &&= \prod_{i=1}^{m} p(\omega_i|\phi); \\ \text{III.} \quad & p_0(\phi). \end{aligned}$$

Esta estructura define un modelo para x parametrizado por $\theta = (\omega, \phi) = (\omega_1, \ldots, \omega_m, \phi)$ y con distribución inicial $p(\theta) = p_0(\phi)p(\omega|\phi)$, de manera que la distribución final está dada por

$$p(\theta|x) \propto p_0(\phi) \prod_{i=1}^{m} \{p(x_i|\omega_i)p(\omega_i|\phi)\}.$$

Entonces $k = m + 1$ y las densidades condicionales completas toman la forma

$$\begin{aligned} p(\theta_1|\theta_2, \ldots, \theta_{k-1}, \theta_k, x) &= p(\omega_1|\phi, x_1) \\ &\vdots \\ p(\theta_{k-1}|\theta_1, \ldots, \theta_{k-2}, \theta_k, x) &= p(\omega_m|\phi, x_m) \\ p(\theta_k|\theta_1, \ldots, \theta_{k-2}, \theta_{k-1}, x) &= p_0(\phi|\omega), \end{aligned}$$

donde $p_0(\phi|\omega) \propto p_0(\phi)p(\omega|\phi)$. ■

Ejemplo 1.8 (Gibbs: modelo normal bivariado)**.** El propósito de este ejemplo es examinar la estructura de la correlación exacta de una cadena de Markov inducida por el muestreo de Gibbs. Suponga que la distribución posterior $p(\theta|D)$ es una distribución normal bivariada $N_2(\mu, \Sigma)$ con

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \qquad y \qquad \sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix},$$

donde μ_j, σ_j, $j = 1, 2$, y ρ son conocidos. Entonces el muestreo de Gibbs requiere muestrear de

$$\theta_1 \sim N\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(\theta_2 - \mu_2), \sigma_1^2(1 - \rho^2)\right)$$

y

$$\theta_2 \sim N\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_1 - \mu_1), \sigma_2^2(1 - \rho^2)\right).$$

Sea $\{\theta_i = (\theta_{1,i}, \theta_{2,i})^t,\ i \geq 0\}$, que denota una cadena de Markov inducida por el muestreo de Gibbs para la distribución normal bivariada anterior. Si iniciamos a partir de la distribución estacionaria, es decir, $\theta_0 \sim N_2(\mu, \Sigma)$, entonces cada una de las sucesiones $\{\theta_{1,i},\ i \geq 0\}$ y $\{\theta_{2,i},\ i \geq 0\}$ es un proceso AR(1).

Veamos este resultado. Sea $\{z_{1,i}, z_{2,i},\ i \geq 0\}$ una sucesión de variables aleatorias i.i.d. $N(0, 1)$. Entonces la estructura del muestreo de Gibbs implica

$$\begin{aligned} \theta_{1,0} &= \mu_1 + \sigma_1 z_{1,0}, \\ \theta_{2,0} &= \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_{1,0} - \mu_1) + \sigma_2\sqrt{1 - \rho^2}z_{2,0}, \end{aligned}$$

y

$$\begin{aligned} \theta_{1,i+1} &= \mu_1 + \rho\frac{\sigma_1}{\sigma_2}(\theta_{2,i} - \mu_2) + \sigma_1\sqrt{1 - \rho^2}z_{1,i+1}, \\ \theta_{2,i+1} &= \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_{1,i+1} - \mu_1) + \sigma_2\sqrt{1 - \rho^2}z_{2,i+1}, \end{aligned} \tag{1.1}$$

para $i \geq 0$. Ahora consideremos el primer componente $\theta_{1,i+1}$. De la ecuación (1.1), para $i \geq 0$,

$$\begin{aligned}\theta_{1,i+1} &= \mu_1 + \rho\frac{\sigma_1}{\sigma_2}\left[\rho\frac{\sigma_2}{\sigma_1}(\theta_{1,i}-\mu_1) + \sigma_2\sqrt{1-\rho^2}z_{2,i}\right] + \sigma_1\sqrt{1-\rho^2}z_{1,i+1} \\ &= \mu_1 + \rho^2(\theta_{1,i}-\mu_1) + \rho\sigma_1\sqrt{1-\rho^2}z_{2,i} + \sigma_1\sqrt{1-\rho^2}z_{1,i+1}. \qquad (1.2)\end{aligned}$$

Sea $\psi = \rho^2$ y $\sigma_1^{*2} = \sigma_1^2(1-\rho^4)$. Sea $\{z_i^*,\ i \geq 0\}$ que denota una sucesión de variables aleatorias i.i.d. $N(0,1)$. Como $z_{1,i}$ y $z_{2,i+1}$ son independientes e idénticamente distribuidas $N(0,1)$, entonces podemos reescribir (1.2) como

$$\begin{aligned}\theta_{1,0} &= \mu_1 + \sigma_1 z_0^*, \qquad (1.3)\\ \theta_{1,i+1} &= \mu_1 + \psi(\theta_{1,i}-\mu_1) + \sigma_1^* z_{i+1}^* \qquad \forall i \geq 0. \qquad (1.4)\end{aligned}$$

Así, $\{\theta_{1,i},\ i \geq 0\}$ es un proceso AR(1) con rezago 1 y autocorrelación $\psi = \rho^2$. Similarmente, $\{\theta_{2,i},\ i \geq 0\}$ es también un proceso AR(1) con rezago 1 y autocorrelación $\psi = \rho^2$. La única diferencia es que usamos $\sigma_2^* = \sigma_2\sqrt{1-\rho^4}$ en vez de σ_1^* en (1.4), y usamos μ_2 y σ_2 en vez de μ_1 y σ_1 en (1.3). ■

1.2.3 Convergencia

Supongamos que se desea generar una muestra de tamaño N de la distribución $p(\theta|x)$. Si para cada uno de N valores iniciales $\theta_1^{(0)}, \ldots, \theta_N^{(0)}$ corremos alguno de los algoritmos discutidos en esta sección, entonces, de acuerdo con la proposición 1.1(i), después de un cierto número de iteraciones T suficientemente grande los valores $\theta_1^{(T)}, \ldots, \theta_N^T$ pueden considerarse como una muestra de tamaño N de la distribución final de θ. Alternativamente podemos generar una sola cadena y tomar los valores $\theta^{(T+K)}, \theta^{(T+2K)}, \ldots, \theta^{(T+NK)}$ como una muestra de $p(\theta|x)$, donde K se elige de manera que la correlación entre las observaciones sea pequeña.

En general no es fácil determinar en qué momento la(s) cadena(s) ha(n) convergido. Un método empírico comúnmente utilizado, basado en la proposición 1.1(ii), consiste en graficar los promedios ergódicos de algunas funciones de θ contra el número de iteraciones y elegir el valor T a partir del cual las gráficas se estabilizan. En este caso es frecuente omitir los primeros valores de la(s) cadena(s) al calcular los promedios ergódicos. La idea de este *periodo de calentamiento* es permitir que la(s) cadena(s) salga(n) de una primera fase de inestabilidad. En el caso particular del muestreo de Gibbs la velocidad de convergencia depende fuertemente de la correlación entre los componentes del vector θ bajo la distribución final $p(\theta|x)$: entre más alta sea la correlación más lenta será la convergencia.

1.3 Software

Para el análisis de datos de estadística Bayesiana se requieren métodos numéricos, y muchos de ellos ya se encuentran en librearías de R o pueden programarse en R (Albert (2009)). A continuación presentaremos otros software utilizados en estadísitica Bayesiana.

1.3.1 WinBUGS

WinBUGS es un *software* de libre acceso en internet que se puede obtener de la página `http://www.mrc-bsu.cam.ac.uk/bugs/`. Fue diseñado por Spiegelhalter, Thomas y Best y es parte del proyecto BUGS (*Bayesian Inference Using Gibbs Sampling*) que desarrollaron estos investigadores para el análisis Bayesiano de modelos estadísticos a través de métodos de Monte Carlo vía cadenas de Markov.

WinBUGS permite que métodos complejos de simulación sean sencillos para los usuarios de la estadística Bayesiana aplicada en diversas disciplinas (Lunn et al. (2012),Lunn et al. (2000), Ntzoufras (2009)).

El *software* ofrece una interfaz con el usuario basada en cuadros de diálogo y comandos a través de los cuales se analiza el modelo, por lo que el ambiente de WinBUGS se vuelve más amigable. Además, también es posible realizar una interfaz con R[1] o S-Plus, los cuales manejan una sintaxis muy similar a la que usa WinBUGS, aunque WinBUGS no cuenta con tantos comandos como R o S-Plus.

La manera de operar de WinBUGS está basada en el muestreo de Gibbs (ver sección 1.2.2); es decir, dada una función de verosimilitud y una distribución inicial, el propósito es muestrear valores de los parámetros del modelo a partir de la distribución final. De estos valores muestreados, es posible obtener estimadores de los parámetros y hacer todo tipo de inferencias sobre éstos.

El desarrollo y mejoramiento de las técnicas de Monte Carlo basadas en cadenas de Markov ha hecho posible construir la distribución final de modelos Bayesianos muy complejos. Sin embargo, generar una muestra a través del algoritmo de Gibbs para obtener la distribución final de un modelo Bayesiano puede ser una tarea muy complicada, sobre todo para el caso en el que se introducen muchos parámetros en el modelo.

WinBUGS proporciona herramientas simples para llevar a cabo la simulación Monte Carlo; a través de WinBUGS es posible implementar el muestreo de Gibbs para una gran variedad de modelos.

[1] *Software* de libre acceso en la página de internet `http://www.r-project.org.`

El *software* tiene ciertas limitaciones para detectar la convergencia, resumir las muestras y realizar diagnósticos sobre el ajuste de los modelos. Sin embargo, es posible usar los paquetes `boa` o `coda` de R junto con WinBUGS para realizar un mejor análisis de las muestras simuladas.

WinBUGS intenta usar el método de muestreo más apropiado para cada parámetro del modelo estocástico de acuerdo con la tabla 1.1.

Distribución Objetivo	Método de muestreo
Discreta	Inversión de una función de distribución acumulada
Forma cerrada	Muestreo directo usando algoritmos estándar
Log-cóncava	Muestreo de rechazo adaptativo
Rango restringido	Muestreo *slice*
Rango sin restricción	Metropolis-Hastings (sección 1.2.1)

Table 1.1: Métodos de muestreo de WinBUGS.

Ejemplo

El siguiente ejemplo se obtuvo del libro de Peter Congdon (2001), *Bayesian Statistical Modelling*. Hay preocupación por el incremento de la incidencia de cáncer de seno en las mujeres, sin embargo, en el Reino Unido el número de muertes por esta enfermedad es más o menos constante. Ambas tendencias implican una disminución de "casos fatales", es decir, una disminución de la tasa de muerte entre los nuevos casos de la enfermedad.

Suponga que estamos interesados en el intervalo de mayor densidad de esta tasa, en una situación donde tenemos $k = 11$ tipos de cáncer.

Sea $\mathbf{n} = (n_1, n_2, \ldots, n_k)$ el número de muertes en cada uno de los 11 tipos de cáncer (1 es cáncer de seno). El vector aleatorio $\mathbf{n}$ tiene una distribución $Multinomial(n_1, \ldots, n_k|\pi, N)$, donde $N = \sum_{i=1}^{k} n_i$ es el número total de muertes por cáncer.

Para elegir la distribución inicial de π se podría tomar el patrón de muertes por cáncer de mujeres en el Reino Unido en años anteriores, o de algún otro país semejante a él.

En este ejemplo se considera una distribución inicial conjugada no informativa (ver sección **??**) para π, la distribución $Dirichlet(\pi|1, 1, \ldots, 1)$.

El programa realizado en WinBUGS es el siguiente:

Modelo

```
model { # Modelo Multinomial
for(i in 1:k) {
```

```
c[i] <- 1
}
pi[1:k] ~ ddirch(c[]) # Inicial
n[1:k] ~ dmulti(pi[],N) # Verosimilitud
}
```

Datos

```
list(n = c(14080, 12990, 6440, 4350, 3420, 3190, 2600, 2420,
1820, 1760, 23610),
k = 11, # Tipos de Cáncer
N = 76680) # Total de Muertes por Cáncer
```

La distribución multinomial puede expresarse, equivalentemente, como la distribución conjunta de k variables aleatorias independientes con distribución Poisson, de tal manera que $n_i \sim Poisson(n_i|\mu_i)$ con $i = 1, \ldots, k$, sujetas a la condición $\sum_{j=1}^{k} n_j = N$ (ver sección **??**). Las probabilidades de la distribución multinomial se obtienen con:

$$\pi_j = \frac{\mu_j}{\sum_{i=1}^{k} \mu_i}.$$

Para este caso las distribuciones iniciales conjugadas para los parámetros μ_i son gamma, y utilizaremos distribuciones iniciales que son no informativas, $Gamma(\mu_i|1, 1)$.

El programa utilizado en WinBUGS es el siguiente:

Modelo

```
model{ # Modelo Poisson
for(i in 1:k) {
mu[i] ~ dgamma(1,1) # Inicial
n[i] ~ dpois(mu[i]) # Verosimilitud
pi[i] <- mu[i]/mu.sum # Parámetro Multinomial
}
mu.sum <- sum(mu[ ])
}
```

Datos

```
list(n = c(14080, 12990, 6440, 4350, 3420, 3190, 2600, 2420,
1820, 1760, 23610),
k = 11) # Tipos de Cáncer
```

Las estadísticas básicas que se obtienen para la distribución final del vector de parámetros π son las mismas en ambos modelos y se presentan en la tabla 1.2.

	mean	sd	MC error	2.5%	median	97.5%
π_1	0.1836	0.00139	$1.542E-5$	0.1809	0.1836	0.1863
π_2	0.1694	0.00135	$1.285E-5$	0.1667	0.1694	0.1721
π_3	0.0840	$9.939E-4$	$8.783E-6$	0.0820	0.0839	0.0859
π_4	0.0567	$8.385E-4$	$8.441E-6$	0.0551	0.0567	0.0584
π_5	0.0446	$7.422E-4$	$6.667E-6$	0.0431	0.0445	0.0460
π_6	0.0416	$7.225E-4$	$7.654E-6$	0.0402	0.0416	0.0430
π_7	0.0339	$6.555E-4$	$7.594E-6$	0.0326	0.0339	0.0351
π_8	0.0315	$6.333E-4$	$6.818E-6$	0.0303	0.0315	0.0328
π_9	0.0237	$5.534E-4$	$4.943E-6$	0.0226	0.0237	0.0248
π_{10}	0.0229	$5.457E-4$	$5.224E-6$	0.0219	0.0229	0.0240
π_{11}	0.3079	0.00168	$1.604E-5$	0.3047	0.3079	0.3113

Table 1.2: Estadísticas básicas de la distribución final de π.

■

1.3.2 JAGS

`Código R:` Bayes7_4DistNegBinom.R

1.3.3 STAN

1.4 INLA

Gomez-Rubio (2020)

Bibliografía

Albert, J. (2009). *Bayesian Computation with R* (second ed.). New York: Springer.

Chen, M.-H., Q.-M. Shao, and J. G. Ibrahim (2000). *Monte Carlo Methods in Bayesian Computation*. Series in Statistics. New York: Springer.

Congdon, P. (2006). *Bayesian Statistical Modelling* (Second ed.). Chichester: John Wiley & Sons.

Gamerman, D. and H. F. Lopes (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference* (Second ed.). Boca Raton, Florida: Chapman & Hall/CRC.

Gilks, W. R., S. Richardson, and D. J. Spiegelhalter (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.

Gomez-Rubio, V. (2020). *Bayesian Inference with INLA* (1st ed.). Boca Raton, Florida: Chapman and Hall/CRC.

Gutiérrez-Peña, E. (1997). *Métodos Computacionales en la Inferencia Bayesiana*. Monografías. México: IIMAS, UNAM.

Lunn, D., C. Jackson, N. Best, A. Thomas, and D. Spiegelhalter (2012). *The BUGS Book - A Practical Introduction to Bayesian Analysis*. Boca Raton, Florida: CRC Press / Chapman & Hall.

Lunn, D. J., A. Thomas, N. Best, and D. Spiegelhalter (2000). WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing 10*(4), 325–337.

Ntzoufras, I. (2009). *Bayesian Modeling Using WinBUGS*. New Jersey: John Wiley & Sons.

Ríos-Insúa, D., S. Ríos-Insúa, J. Martín-Jiménez, and A. Jiménez-Martín (2009). *Simulación: métodos y aplicaciones*. Alfaomega.

Robert, C. P. and G. Casella (2000). *Introducing Monte Carlo Methods with R*. Use R! Springer.