

El algoritmo Gibbs sampler

Jorge Francisco de la Vega Góngora
Profesor del Departamento de Estadística del ITAM

*“All probabilities are conditional on something, and to be useful
they must condition on the right thing.”*

Frank Harrell

El *Gibbs Sampler* (GS), o muestreador de Gibbs como le llamaríamos en español, es uno de los avances en el ámbito estadístico más notables del siglo XX que ha facilitado un crecimiento exponencial en las aplicaciones prácticas, especialmente aquellas que tienen un enfoque Bayesiano. Las ideas asociadas al GS permiten descomponer problemas muy complejos en conjuntos de problemas más pequeños y fáciles de resolver. Este algoritmo es una de las piezas claves que permitió conectar una serie de ideas que resultaron en la consolidación de la Estadística Bayesiana y su aplicación intensiva en la ciencia actual.

Introducción

Los métodos de Monte Carlo (MC) estudian la generación de una muestra de observaciones $\theta_1, \dots, \theta_m$ provenientes de una función de distribución $F(\theta)$ específica, que puede ser univariada o multivariada. Usualmente tales muestras se utilizan para simular el comportamiento aleatorio de fenómenos o sistemas complejos y tratar de comprender su funcionamiento bajo un ambiente controlado y poder realizar inferencias.

En ese contexto se han propuesto una gran variedad de algoritmos y métodos. La primera generación de métodos MC, fueron desarrollados por Nicholas Metropolis, Stanislaw Ulam y John Von Neumann, entre otros, durante los años 40's y 50's. Se basan en generar muestras *independientes* de la distribución de interés, a partir de números uniformes independientes, o a través de métodos indirectos, como el método de aceptación y rechazo, que se asemeja al lanzamiento de dardos en una región y ‘aceptando’ los puntos que caen en la región de interés. Aún cuando estos métodos pueden ser eficientes y fáciles de implementar en el caso univariado, resultan complejos en dimensiones mayores.

Otro conjunto de métodos MC fueron desarrollados por el mismo Metropolis, en conjunto con el par de matrimonios de científicos Rosenbluth y Teller, para obtener muestras de la distribución de interés, pero en donde las observaciones ya no son independientes. Estos métodos se basan en generar observaciones de una cadena de Markov ergódica cuya distribución límite es la distribución de interés. El algoritmo de Metropolis fue perfeccionado en 1970 por W. Keith Hastings, quien lo generalizó y le dio la forma final al algoritmo de Metropolis-Hastings que se reconoce hasta hoy y que se conoce como *Markov Chain Monte Carlo* (MCMC). Más detalle sobre esta parte de la historia se puede consultar, por ejemplo en [6].

Aun cuando se ha probado que el GS es un caso particular del algoritmo de Metropolis-Hastings, aquel tuvo un origen diferente y no fue hasta varios años después que se ha mostrado la conexión entre los dos.

El procesamiento de imágenes y su análisis es un tema relevante para los militares, la industria de la automatización y el diagnóstico médico. Las imágenes borrosas requieren el procesamiento de señales, eliminación de ruido y afinación para hacerlas reconocibles. A principios de la década de los 80's, Ulf Grenader en Brown diseñaba modelos matemáticos para imágenes médicas explorando el efecto que un pixel puede tener en unos cuantos de sus vecinos. Los cálculos involucraban fácilmente más de un millón de variables desconocidas. Stuart Geman atendía el seminario de Grenader y junto con su hermano Donald Geman trataban de restaurar una fotografía borrosa de una señal en el camino.

Stuart y Donald inventaron una variante MC que fue particularmente adecuada para problemas de restauración de imágenes. Decidieron nombrar el método por analogía del problema que estaban resolviendo al de la configuración de una caja de chocolates. Un regalo popular para el día de las madres en esa época era la “*Whitman's sampler*” que contenía dentro de la caja un diagrama que identificaba el relleno oculto de cada chocolate. Como el diagrama era muy parecido a una matriz de variables ocultas, decidieron llamar al método como el **Gibbs Sampler**, ya que en su problema utilizaban la distribución de Gibbs, en honor al físico americano del siglo XIX, Josiah Williard Gibbs.

El propósito de esta nota es explicar, de manera sencilla e intuitiva, y sin entrar en formalidad matemática estricta, las ideas fundamentales del algoritmo y mostrar con ejemplos algunas de sus aplicaciones prácticas.

Marco Teórico

Conceptos y algoritmo

En simples palabras, la idea del GS es que las distribuciones condicionales determinan las distribuciones marginales. Si $\pi(\boldsymbol{\theta})$ es una densidad conjunta donde $\boldsymbol{\theta}$ es un vector particionado en $p > 1$ subvectores $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_p)$ el problema a resolver es obtener la densidad marginal para un subconjunto de las variables en el vector $\boldsymbol{\theta}$, digamos $\boldsymbol{\theta}_i$:

$$\pi_i(\boldsymbol{\theta}_i) = \int \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}_{-i},$$

donde $\boldsymbol{\theta}_{-i} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, \dots, \boldsymbol{\theta}_p)$. La integral anterior puede ser muy difícil de obtener (analítica o numéricamente).

El Gibbs Sampler permite generar una muestra $\boldsymbol{\theta}_i^{(1)}, \dots, \boldsymbol{\theta}_i^{(m)}$ de la distribución $\pi_i(\boldsymbol{\theta}_i)$ sin requerir explícitamente la distribución π_1 , a partir de las distribuciones condicionales:

$$\pi(\boldsymbol{\theta}_j | \boldsymbol{\theta}_{-j}), \quad j = 1, \dots, p$$

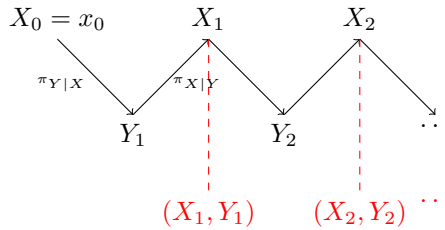
Aterrizando ideas

iterando sobre los p subvectores de la partición del vector θ . En particular, permite obtener una muestra de la distribución completa $\pi(\theta)$ a partir de las distribuciones marginales.

Para aterrizar ideas, consideremos el caso bidimensional, $(X, Y) \sim \pi(x, y)$. El algoritmo propone generar una sucesión de parejas (X_j, Y_j) del siguiente modo. Para $j = 1, 2, \dots, n$:

1. Se inicializa $X_0 = x_0$.
2. Genera $Y_j \sim \pi_{Y|X}(\cdot | X_{j-1})$.
3. Genera $X_j \sim \pi_{X|Y}(\cdot | Y_j)$
4. Hacer $j := j + 1$.

Una representación gráfica del flujo del algoritmo se muestra a continuación:



La sucesión de valores $X_0, Y_0, X_1, Y_1, \dots, X_k, Y_k$ se conoce como **sucesión de Gibbs**. Hay dos maneras de tomar muestras para π_X : ya sea tomando los valores de la sucesión de Gibbs a partir de una k grande, lo que daría una muestra de observaciones de π_X no independientes, o bien, si se generan m sucesiones de Gibbs independientes de longitud k , y tomando las “últimas” observaciones de esas m sucesiones, suponiendo nuevamente que k es suficientemente grande.

Para el caso con más de dos variables, se particiona el vector θ en k componentes $\theta = (\theta_1, \theta_2, \dots, \theta_k)$, y si podemos simular observaciones de $\pi_i(\theta_i | \theta_{-i})$, entonces podemos aplicar el siguiente algoritmo. Para $j = 1, \dots, n$:

1. Inicializar $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_k^{(0)})$ en un punto relativamente arbitrario.

2. Obtener $\theta^{(j)}$ a partir de $\theta^{(j-1)}$ generando

$$\begin{aligned}\theta_1^{(j)} &\sim \pi(\theta_1 | \theta_2^{(j-1)}, \dots, \theta_k^{(j-1)}) \\ \theta_2^{(j)} &\sim \pi(\theta_2 | \theta_1^{(j)}, \theta_3^{(j-1)}, \dots, \theta_k^{(j-1)}) \\ \theta_3^{(j)} &\sim \pi(\theta_3 | \theta_1^{(j)}, \theta_2^{(j)}, \dots, \theta_k^{(j-1)}) \\ &\vdots \\ \theta_{k-1}^{(j)} &\sim \pi(\theta_{k-1} | \theta_1^{(j)}, \theta_2^{(j)}, \dots, \theta_k^{(j-1)}) \\ \theta_k^{(j)} &\sim \pi(\theta_k | \theta_1^{(j)}, \theta_2^{(j)}, \dots, \theta_{k-1}^{(j)})\end{aligned}$$

3. Hacer $j := j + 1$ y repetir a partir del paso 2 hasta alcanzar n .

Funcionamiento del algoritmo

En su artículo, Stuart y Donald Geman [8] demuestran la convergencia del GS. Sin embargo, su demostración es oscura y compleja, utilizando lenguaje no estadístico y haciendo uso de conceptos de construcción de imágenes digitales, redes neuronales y sistemas expertos.

La naturaleza Markoviana del GS es mostrada en el siguiente ejemplo de Casella y George [4], que seguiremos aquí. No pretende ser la prueba del algoritmo sino ilustrar su fundamento a través de un caso particular.

Consideramos $\theta = (X, Y)$ dos variables Bernoulli con distribución conjunta dada por

$$P(X = i | Y = j) = p_{ij} \text{ con } p_{ij} \geq 0, \quad p_{11} + p_{12} + p_{21} + p_{22} = 1.$$

La distribución marginal de X , por ejemplo, es Bernoulli con parámetro $p_{21} + p_{22}$.

Por otra parte, las distribuciones condicionales se pueden expresar en dos matrices, $A_{x|y}$ y $A_{y|x}$ con los siguientes elementos:

$$A_{x|y} = \begin{bmatrix} \frac{p_{11}}{p_{11}+p_{21}} & \frac{p_{21}}{p_{11}+p_{21}} \\ \frac{p_{12}}{p_{12}+p_{22}} & \frac{p_{22}}{p_{12}+p_{22}} \end{bmatrix} \quad \text{y} \quad A_{y|x} = \begin{bmatrix} \frac{p_{11}}{p_{11}+p_{12}} & \frac{p_{12}}{p_{11}+p_{12}} \\ \frac{p_{21}}{p_{21}+p_{22}} & \frac{p_{22}}{p_{21}+p_{22}} \end{bmatrix}$$

Por ejemplo, $P(Y = 1 | X = 0)$, el elemento (1,2) de $A_{y|x}$:

$$P(Y = 1 | X = 0) = \frac{P(Y = 1, X = 0)}{P(X = 0)} = \frac{p_{12}}{p_{11} + p_{12}}.$$

Para generar una sucesión de Gibbs, $Y_0, X_0, Y_1, X_1, \dots, Y_k, X_k, \dots$ las matrices $A_{y|x}$ y $A_{x|y}$ se pueden considerar como las matrices de transición dando las probabilidades condicionales $P(Y = i | X = j)$ y viceversa.

Si queremos obtener la distribución marginal de X , nos concentramos sólo en las observaciones X 's de la sucesión de Gibbs. Podemos pensar entonces en la cadena de Markov que lleva de los estados iniciales de X a estados posteriores de X , pasando por los estados de Y , con matriz de transición:

$$A_{x|x} = A_{y|x}A_{x|y}.$$

Entonces podemos ver que la matriz de transición en k pasos, que nos da las probabilidades $P(X_k = i | X_0 = j)$ es $A_{x|x}^k$. La distribución marginal de X_k se puede escribir como $\pi_k = (\pi_k(0), \pi_k(1))$ y entonces podemos ver que se cumple la siguiente relación de recurrencia:

$$\pi_k = \pi_0 A_{x|x}^k = (\pi_0 A_{x|x}^{k-1}) A_{x|x} = \pi_{k-1} A_{x|x}$$

Si las entradas de $A_{x|x}$ son positivas y dada cualquier distribución inicial π_0 , π_k converge a una única distribución π que satisface la condición de punto fijo ¹: $\pi = \pi A_{x|x}$. Entonces, si la sucesión de Gibbs converge, la π que satisface esta recurrencia es la distribución marginal de X , $\pi = \pi_x$.

Ejemplos y Aplicaciones

GS en modelos jerárquicos

Comenzaremos con una de las aplicaciones más comunes del GS. Dennis Lindley junto con uno de sus estudiantes, Adrian F. M. Smith, descompusieron varios problemas complejos en modelos jerárquicos, en donde resulta muy fácil aplicar el GS ya que se cuenta con las condicionales de manera explícita.

Consideraremos el modelo jerárquico Binomial-Beta:

$$\begin{aligned} X|\theta &\sim \mathbf{Bin}(n, \theta) \\ \theta &\sim \mathcal{Be}(a, b) \end{aligned}$$

En este modelo densidad conjunta de X y θ se puede escribir como

$$\begin{aligned} f(X, \theta) = f(X|\theta)f(\theta) = f(\theta|X)f(X) &= \binom{n}{x} \theta^x (1-\theta)^{n-x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1} (1-\theta)^{b-1} \\ f(\theta|X) = \frac{f(X, \theta)}{f(X)} &\propto \binom{n}{x} \theta^{x+a-1} (1-\theta)^{n-x+b-1} \end{aligned}$$

Entonces $\theta|X \sim \mathcal{Be}(x+a, n-x+b)$ (beta y binomial son familias conjugada). Así que para generar de la distribución conjunta (X, θ) se puede generar de $f(X|\theta)$ y de $f(\theta|X)$. También podemos tener la distribución marginal de X . Queremos obtener muestras de la distribución conjunta de los datos y el parámetro. En el siguiente cálculo numérico se considera $X|\theta$ una binomial con $n = 15$, y para obtener θ utilizamos $a = 3$ y $b = 7$. El siguiente código de R obtiene una muestra de tamaño 5,000 para los pares (X, θ) .

¹Esto se puede revisar en cualquier libro básico de procesos estocásticos, por ejemplo, [2].

```
set.seed(12345)
nsim <- 5000
n <- 15 #Parámetro dado
a <- 3; b <- 7 #Parámetros de Beta
Theta <- rbeta(1,a,b) #Valores iniciales
X <- rbinom(1,n,Theta[1])
for(i in 2:nsim){
  X[i] <- rbinom(1,n,Theta[i])
  Theta[i] <- rbeta(1,a+X[i],n-X[i]+b)
}
```

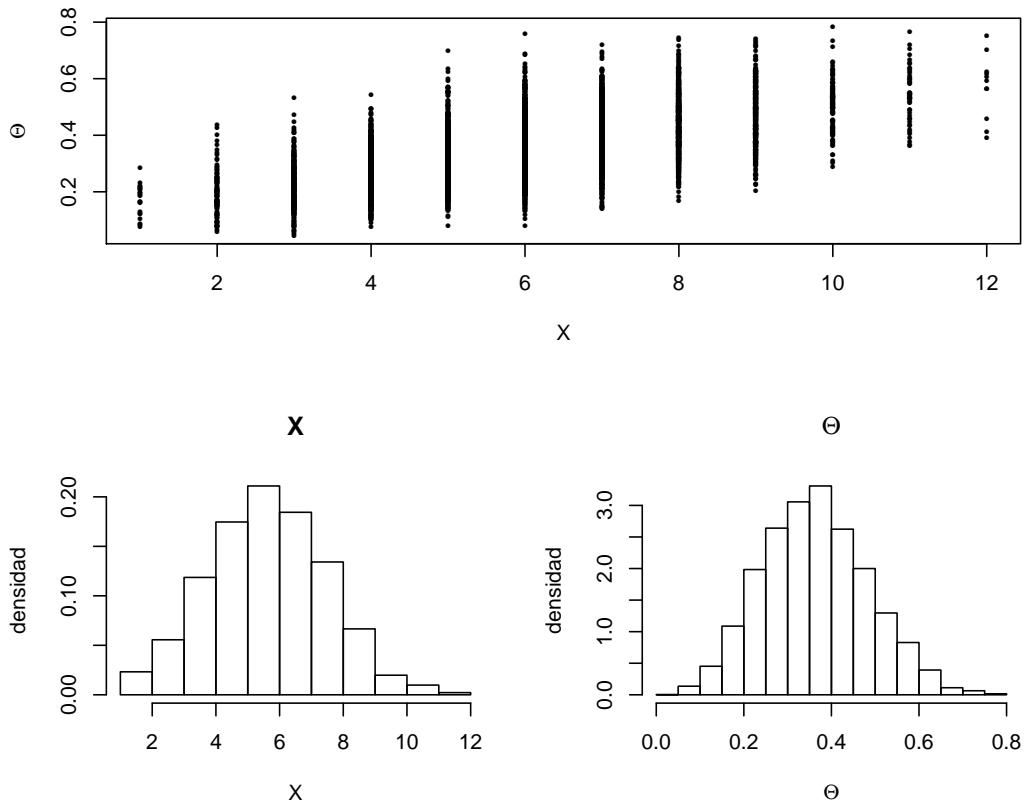


Figura 1: Simulación de la distribución conjunta de (X, θ) y sus distribuciones marginales

Modelo de Ising

El siguiente ejemplo muestra una aplicación del GS que utiliza una distribución de Gibbs como en el artículo de Geman y Geman [8]. El modelo de Ising fue propuesto como problema de tesis por Whilhem Lenz to Ernst Ising, su alumno, que lo resolvió en 1925. Es un modelo matemático de ferromagnetismo en mecánica estadística, pero que también ha sido usado para el procesamiento de imágenes digitales. El modelo de Ising se estudia, entre otras razones, por sus propiedades de transición de fase.

En una gráfica retícula de $n \times n$ como la siguiente de 4×4 :

σ_{11}	σ_{12}	σ_{13}	σ_{14}
σ_{21}	σ_{22}	σ_{23}	σ_{24}
σ_{31}	σ_{32}	σ_{33}	σ_{34}
σ_{41}	σ_{42}	σ_{43}	σ_{44}

A cada vértice v se le asigna un color, por ejemplo blanco con valor $+1$ o rojo con valor -1 . Una configuración $\sigma = (\sigma_v)$ es una asignación de colores en los vértices de la retícula. Hay n^2 vértices y 2^{n^2} posibles configuraciones. Cada configuración tiene una energía asociada, dada por la función $E(\sigma) = -\sum_{v \sim w} \sigma_v \sigma_w$, donde la relación ' \sim ' es de vecindad (cada vértice tiene cuatro vecinos: arriba, abajo, izquierda y derecha, excepto en la frontera de la retícula). La configuración dada en el ejemplo de arriba tiene energía $E(\sigma) = 12$.

La distribución de Gibbs es una distribución de probabilidad sobre el conjunto de configuraciones con un parámetro T :

$$\pi(\sigma) = \frac{e^{-E(\sigma)/T}}{\sum_{\tau} e^{-E(\tau)/T}}$$

El parámetro T tiene interpretación física de temperatura. En este modelo, si la temperatura es infinita, la distribución de Gibbs es uniforme sobre el conjunto de configuraciones. En dos dimensiones, el sistema entra en un cambio radical de conducta a una temperatura crítica de $T = 2.269$. Los detalles se pueden ver en [10].

El GS se puede usar fácilmente para simular este modelo. Dada una configuración σ , un vértice v se escoge uniformemente al azar. El color en ese sitio se actualiza de la distribución condicional de ese vértice dados los otros vértices de la configuración σ .

Si denotamos σ_{-k} a la configuración sin el vértice k , y σ^+ a la configuración con un $+1$ en el vértice k y σ^- a la configuración con un -1 en ese vértice, entonces podemos calcular la probabilidad condicional:

$$P(\sigma_k = +1 | \sigma_{-k}) = \frac{P(\sigma^+)}{P(\sigma_{-k})} = \frac{P(\sigma^+)}{P(\sigma^+) + P(\sigma^-)} \quad (1)$$

$$= \frac{\exp\{-E(\sigma^+)/T\}}{\exp\{-E(\sigma^+)/T\} + \exp\{-E(\sigma^-)/T\}} \quad (2)$$

$$= \frac{1}{1 + \exp\{(E(\sigma^+) - E(\sigma^-))/T\}} \quad (3)$$

$$= \frac{1}{1 + \exp\{-2 \sum_{i \sim k} \sigma_i\}} \quad (4)$$

La última igualdad se obtiene notando que en $E(\sigma^+)$ y $E(\sigma^-)$ se pueden separar los sumandos que no son vecinos del vértice k y se cancelan entre sí, sobreviviendo sólo los vértices vecinos de k .

Implementamos el GS tomando un vértice al azar y actualizando cada vértice de acuerdo a las distribuciones condicionales obtenidas.

```
g <- 150 #Elementos de la retícula
valT <- c(Inf, 2.267574, 1, -1) #valores de temperatura
final <- list(NULL)

for(v in valT){
  nsim <- 100000 #número de iteraciones
  M <- matrix(sample(c(-1, 1), (g+2)^2, rep=T), nrow=g+2) #retícula
  M[c(1, g+2), ] <- 0 #asigna 0 a los extremos de la retícula valor 0
  M[, c(1, g+2)] <- 0

  for(m in 1:nsim){
    i <- sample(2:(g+1), 1) #muestrea renglón y columna
    j <- sample(2:(g+1), 1)
    E <- M[i, j+1] + M[i, j-1] + M[i-1, j] + M[i+1, j] #calcula la energía del vértice
    p <- 1/(1+exp(-2*E/v)) #probabilidad del vértice
    if(runif(1)<p) M[i, j] <- 1 else M[i, j] <- -1 #actualiza el color del vértice
  }
  final[[match(v, valT)]] <- M[2:(g+1), 2:(g+1)]
}
```

La simulación del modelo de Ising se realizó generando 100,000 iteraciones en un grid de 150×150 para cuatro valores de la temperatura. En este contexto, el espacio de estados de las configuraciones tiene 2^{22500} elementos. La simulación prácticamente no toma tiempo de ejecución en una laptop y se puede observar cómo impacta la temperatura en la concentración de energía.

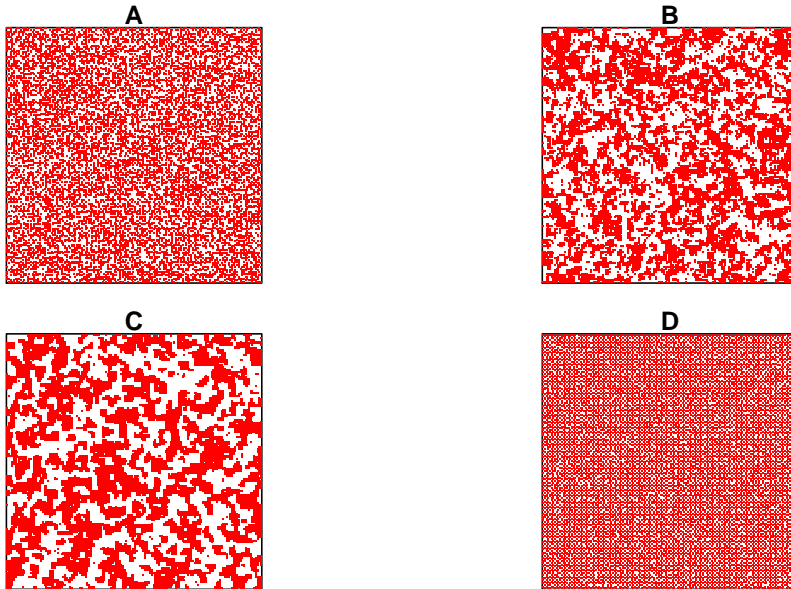


Figura 2: Simulación del modelo de Ising para un modelo con grid 150^2 . A: $T = \infty$, B: 2.267574, C: 1, D: -1

Análisis de Punto de Cambio

En este último ejemplo consideramos un proceso Poisson con punto de cambio, esto es, un punto en el tiempo donde la tasa de eventos cambia:

$$X_t \sim \begin{cases} \mathcal{P}(\mu t) & 0 < t \leq k \\ \mathcal{P}(\lambda t) & t > k \end{cases}$$

Dada una muestra de n observaciones del proceso anterior, el problema consiste en estimar μ , λ y k . Este es un proceso muy común y que ha sido ampliamente estudiado. Hay varias opciones de especificación de modelos Bayesianos para resolverlo. Una aplicación concreta del problema anterior se relaciona con los datos publicados en el artículo de Jarret [9]. Los datos corresponden a las fechas de 191 explosiones en minas de carbón que resultaron en más de 10 fatalidades desde marzo 15, 1851 hasta marzo 22, 1962. Los datos se encuentran en `coal`, en el paquete `boot` de R.

Los datos muestran un cambio en el número promedio de desastres por año alrededor de 1900. Los números anuales de accidentes se muestran a continuación:

2.65704155889863948976832052116954270403668880256364975808182748398572507

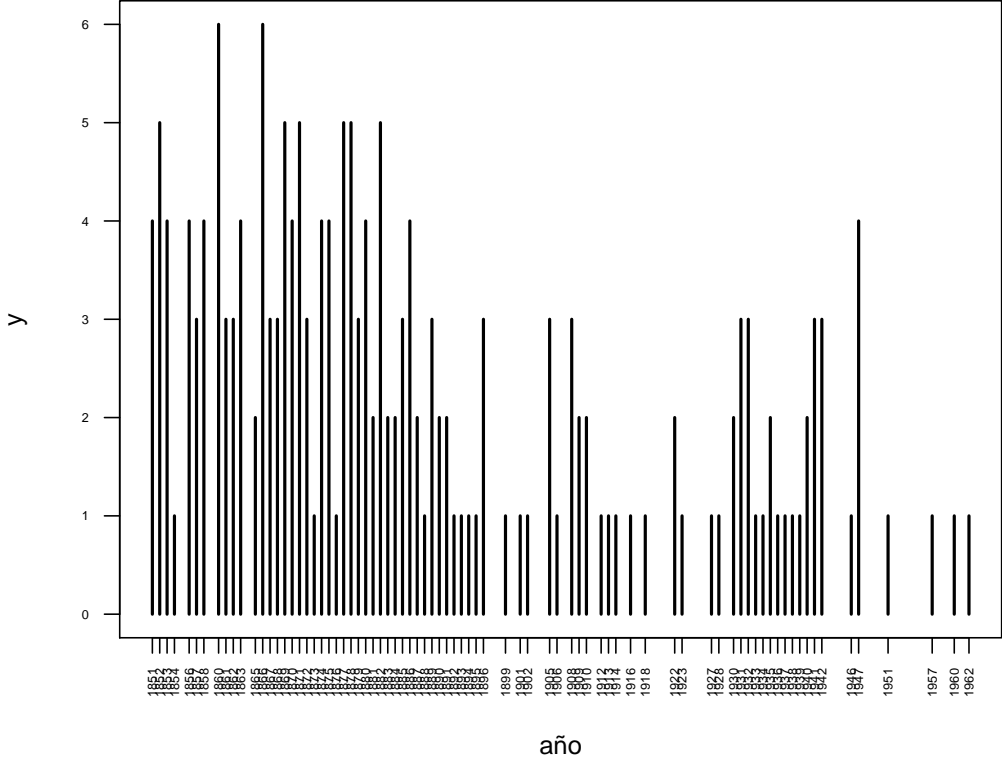


Figura 3: Frecuencia de accidentes fatales en minas de carbón entre 15/03/1851 y 22/03/1962.

```

y
[1] 4 5 4 1 0 4 3 4 0 6 3 3 4 0 2 6 3 3 5 4 5 3 1 4 4 1 5 5 3 4 2 5 2 2 3
[36] 4 2 1 3 2 2 1 1 1 1 3 0 0 1 0 1 1 0 0 3 1 0 3 2 2 0 1 1 1 0 1 0 1 0 0
[71] 0 2 1 0 0 0 1 1 0 2 3 3 1 1 2 1 1 1 1 2 3 3 0 0 0 1 4 0 0 0 1 0 0 0 0
[106] 0 1 0 0 1 0 1
    
```

Modelamos estos datos de la siguiente manera. Definimos Y_i como el número de desastres en el año i (1851=1). Entonces, si k es el año punto de cambio,

$$Y_i \sim \mathcal{P}(\mu), \quad i = 1, \dots, k,$$

$$Y_i \sim \mathcal{P}(\lambda), \quad i = k + 1, \dots, n$$

Aterrizando ideas

Hay $n = 112$ observaciones terminando en el año 1962. El parámetro es $\theta = (k, \mu, \lambda)$. Se requiere estimar como distribución objetivo la posterior $\pi(k, \mu, \lambda|y)$, y en particular, la distribución posterior de k , $\pi(k|y, \mu, \lambda)$.

Se puede construir un modelo Bayesiano con las siguientes distribuciones iniciales *independientes*:

$$\begin{aligned}k &\sim U\{1, 2, \dots, n\}, \\ \mu &\sim \mathcal{G}(a_1, b_1), \\ \lambda &\sim \mathcal{G}(a_2, b_2)\end{aligned}$$

donde a_1 , a_2 , b_1 y b_2 son hiperparámetros que se pueden fijar o bien, se pueden considerar a su vez aleatorios. Sean $S_k = \sum_{i=1}^k Y_i$ el total de eventos hasta el año k y $S'_k = S_n - S_k$ el total de eventos entre el año $k + 1$ y el año n . Para aplicar GS, se necesita especificar las distribuciones condicionales posteriores. Las densidades condicionales para k , μ y λ están dadas por:

$$\begin{aligned}\mu|y, k &\sim \mathcal{G}(a_1 + S_k, k + b_1) \\ \lambda|y, k &\sim \mathcal{G}(a_2 + S'_k, n - k + b_2) \\ k|y, \mu, \lambda &\sim \frac{L(Y|k, \mu, \lambda)}{\sum_{j=1}^n L(Y|j, \mu, \lambda)}\end{aligned}$$

donde L es la función de verosimilitud

$$L(Y|k, \mu, \lambda) = e^{k(\lambda - \mu)} \left(\frac{\mu}{\lambda}\right)^{S_k}.$$

A continuación se muestra el código en R para llevar a cabo la simulación.

```
#Simulación de los datos para la distribución del punto de cambio.
n <- length(y) #longitud de los datos
m <- 2000      #longitud de la cadena
mu <- lambda <- k <- numeric(m)
L <- numeric(n) #inicializo vector de valores de verosimilitud
k[1] <- sample(1:n,1) #valor inicial del punto de cambio (seleccionado al azar)
mu[1] <- 1
lambda[1] <- 1
b1 <- b2 <- 1 #valores de los hiperparámetros
a1 <- a2 <- 2

for (i in 2:m){
  #genera mu
  mu[i] <- rgamma(1, shape = a1 + sum(y[1:k[i-1]]), rate = k[i-1] + b1)
  #genera lambda
  lambda[i] <- rgamma(1, shape = a2 + sum(y)-sum(y[1:k[i-1]]),
                    rate = n - k[i-1] + b2)
  for(j in 1:n){
    L[j] <- exp((lambda[i]-mu[i])*j) * (mu[i]/lambda[i])^sum(y[1:j])
  }
}
L <- L/sum(L)
```

```
#genera k de la distribución discreta L en 1:n
k[i] <- sample(1:n,prob=L,size=1)
}
```

La Figura 4 muestra las gráficas de las trazas de las cadenas para los tres parámetros muestreados, y la Figura 5 muestra histogramas de las densidades marginales obtenidas.

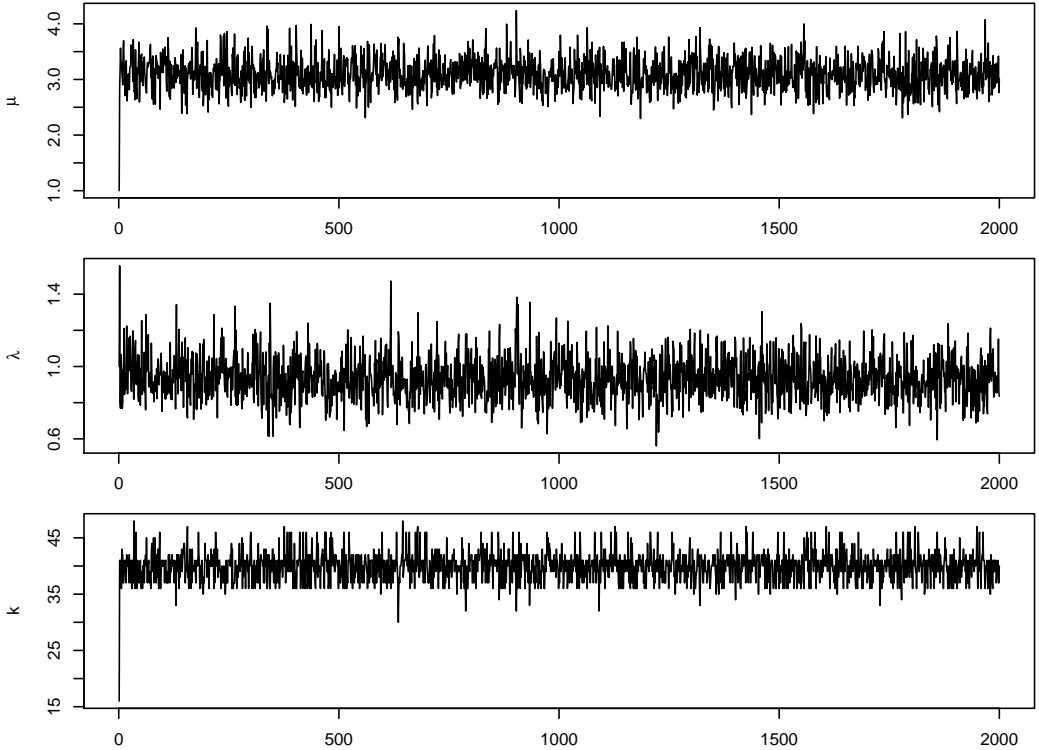


Figura 4: Trazas de las cadenas generadas por GS para los tres parámetros considerados: μ , λ y k

Entonces el punto de cambio está alrededor de $\hat{k} = 40$, que corresponde al año $1851 + 40 = 1891$. De 1851 a 1890 la media Poisson es alrededor de $\hat{\mu} \approx 3.1$ y del año 1891 hacia adelante la media es $\hat{\lambda} \approx 0.93$.

```
# histogramas de la salida del Gibbs sampler:
b <- 500 #burn-in
par(mfrow=c(1,3))
label1 <- paste("mu=",round(mean(mu[b:m]),1))
label2 <- paste("lambda=", round(mean(lambda[b:m]),1))
hist(mu[b:m], main="", xlab=label1, prob=TRUE) #mu posterior
hist(lambda[b:m], main="", xlab=label2, prob=TRUE) #lambda posterior
hist(k[b:m], breaks = min(k[b:m]):max(k[b:m]), prob=TRUE, main="", xlab = "Punto de cambio k")
```

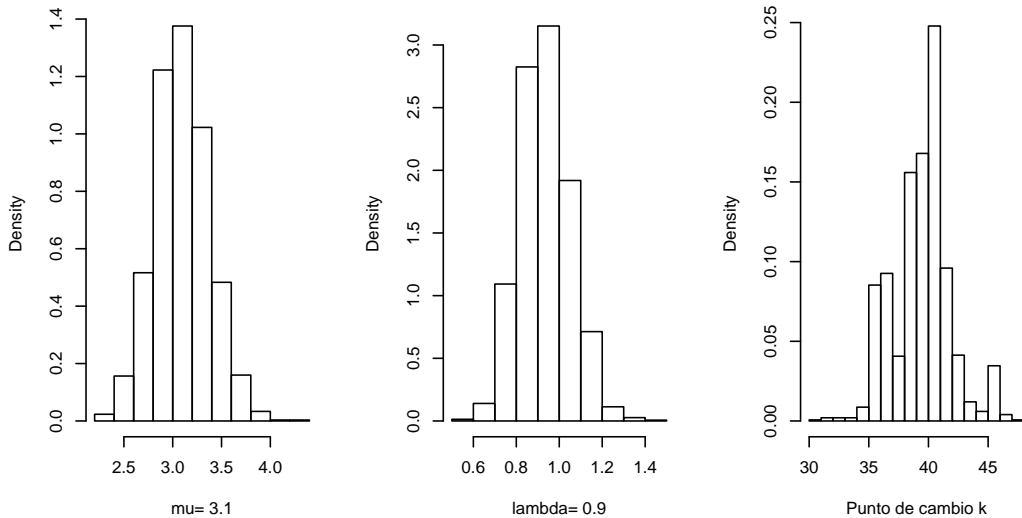


Figura 5: Histogramas de las distribuciones marginales para los tres parámetros estimados

Conclusiones

El Gibbs sampler es una herramienta poderosa de simulación que ha tenido un impacto significativo en la investigación científica en general, permitiendo que una gran variedad de planteamientos bayesianos puedan resolverse computacionalmente y junto con el algoritmo de Metropolis-Hastings, cambiaron por completo el paradigma estadístico para resolver problemas.

Hay varios aspectos relevantes que deben ser estudiados con más detalle. Por ejemplo, ¿por cuánto tiempo es necesario simular la cadena para considerar que las observaciones ya están lo suficientemente cerca a la distribución objetivo? De acuerdo a Charles Geyer [11], fuerte contribuidor a la teoría de MCMC, no se requiere esperar, o en otras palabras, lo que se conoce como el período de *burn-in* es innecesario.

En esta nota se habló poco de las herramientas computacionales y todos los ejemplos se hicieron en R. Sin embargo, existen muchos otros programas que se han realizado ex-profeso para el cálculo del GS, entre ellas se encuentran **BUGS** (*Bayesian Analysis Using the Gibbs Sampler*), desarrollado por David Spiegelhalter en Cambridge desde 1989 y liberado en 1991. De aquí se deriva **OpenBUGS** que tiene una interface a R. También se cuenta con **JAGS** (*Just Another Gibbs Sampler*) y con **Stan**, que tiene soporte para varios lenguajes de programación.

Esta y otras herramientas computacionales importantes son cubiertas en cursos académicos como Estadística Computacional, Simulación Estocástica, Métodos computacionales Bayesianos, entre otras posibilidades. Estas herramientas deberían formar parte del herramienta cotidiano de todos los estadísticos y los ahora llamados *científicos de datos*.

Referencias

- [1] Robert, Christian y Casella, George. *Monte Carlo statistical Methods*. Springer, USA, 2004.
- [2] Dobrow, Robert. *Introduction to Stochastic Processes with R*. Wiley, USA, 2016.
- [3] Bertsch Mcgrayne, Sharon. *The Theory That Would Not Die*. Yale University Press, USA, 2011.
- [4] Casella, George, George, Edward I. “Explaining the Gibbs sampler”, *The American Statistician* **46** (1992):167-174.
- [5] Eckhardt, Roger. “Stan Ulam, John von Neumann and the Monte Carlo Method”, *Los Alamos Science* **15** Special Issue (1987):131-137.
- [6] Alvarado, Christian, Diluvi, Gian Carlo y Espinosa, Demian “El algoritmo de Metropolis-Hastings”, *Laberintos e Infinitos* **41** (2016):21-29.
- [7] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. “Equation of State Calculations by Fast Computing Machines”, *J. Chem. Phys.* **21** (1953):1087-1092.
- [8] Geman, S. and Geman, D. “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images”, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **6** (1984):721-741.
- [9] Jarret, R. G. “A note on the intervals between coal-mining disasters.”, *Biometrika* **66** (1979):191-193.
- [10] “Square-lattice Ising model.” *Wikipedia*. Consultado el 16 de marzo de 2018.
https://en.wikipedia.org/wiki/Square-lattice_Ising_model
- [11] “Burn-in is unnecessary.” *Charles Geyer*. Consultado el 10 de enero de 2018.
<http://users.stat.umn.edu/~geyer/mcmc/burn.html>