Decimocuarta Semana Internacional de la Estadística y la Probabilidad
14-18 de junio de 2021

Reversible Jump MCMC (Green)

Juan Antonio Cruz Juárez [a], Hortensia J. Reyes Cervantes[b]
[a,b] *Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico Matemáticas, Puebla, Puebla, México*
[a]juan.cruzjuar@correo.buap.mx, [b]*hreyes@fcfm.buap.mx*.

Resumen
En el presente trabajo se revisa los aspectos importantes que se deben considerar cuando trabajamos con el algoritmo de Reversible Jump propuesto por Green (1995). El algoritmo es de gran utilidad cuando trabajamos modelos con inferencia bayesiana en donde la distribución a posterior es compleja, además de que al momento de utilizar métodos Monte Carlo vía cadenas de Markov (MCMC) el espacio de estado correspondiente a las simulaciones cambia de dimensión.

Palabra claves: Inferencia bayesiana, MCMC, reversible jump.

Introducción:
Cuando trabajamos sobre modelo estadísticos, podemos abordar las características del modelo (media, varianza, etc.) mediante inferencia clásica o bayesiana. Para ambos enfoques debemos tener en cuenta la información con la que se cuenta sobre el modelo, sin embargo, cada uno tiene sus ventajas y desventajas. En esta ocasión abordamos el enfoque bayesiano, que tiene soporte sobre el teorema de Bayes.
Dada la observación y con distribución $P(\theta, y)$, donde θ es el parámetro(s) del modelo, entonces debemos hallar la distribución de probabilidad conjunta entre θ y y. La función de distribución de probabilidad conjunta es el producto de dos densidades; una es llamada distribución **a priori** ($P(\theta)$) y la otra distribución de muestreo (distribución de los datos $P(y|\theta)$),

$$P(\theta, y) = P(\theta)P(y|\theta). \quad (1.1)$$

Condicionando el valor conocido de la observación y y utilizando la regla de Bayes tenemos la distribución de interés llamada **a posterior**

$$\pi(\theta|y) = \frac{P(\theta, y)}{P(y)} = \frac{P(\theta)P(y|\theta)}{P(y)}, \quad (1.2)$$

donde $P(y) = \sum_\theta P(\theta)P(y|\theta)$, o $P(y) = \int P(\theta)P(y|\theta)$ ya sea en el caso discreto o continuo respectivamente. Como puede observar el factor de (1.2) no depende de θ, así se considera constante y se tiene la ecuación equivalente:

$$\pi(\theta|y) \propto P(\theta)P(y|\theta), \quad (1.3)$$

donde $P(y|\theta)$ es la función de verosimilitud del modelo. Cuando se tiene una distribución a posterior compleja se utilizan algoritmos conocidos como métodos Monte Carlo vía cadenas de Markov (MCMC) para obtener una muestra simulada y extraer conclusiones e inferencia.

2. Algoritmos MCMC
Los elementos que forman parte de los algoritmos son las cadenas de Markov y los métodos Monte Carlo. En general los métodos Monte Carlo se dividen en dos categorías (Adrian Barbu 2020):

1. Monte Carlo secuencial, es útil en espacios de dimensión pequeña.
2. MCMC, que simulan una cadena de Markov cuya distribución estacionaria converge a la distribución objetivo.

2.1 Métodos Monte Carlo.

Considere un conjunto de observaciones $\{x_1, x_2, \dots, x_n\} \sim f(x)$ que representa una muestra del modelo probabilístico verdadero $f(x)$, aunque sea desconocido se puede aproximar por una muestra empírica de las observaciones. Para construir un algoritmo MCMC se necesita:

1. Simular un sistema con su respectiva distribución de probabilidad $\pi(x)$.
2. Estimar cantidades mediante la integración Monte Carlo.

$$c = E_\pi\big(f(x)\big) = \int \pi(x) f(x) dx. \qquad (2.1)$$

3. Optimizar la distribución a posterior para hallar los momentos (máximos o mínimos).

$$x^* = argmax\, \pi(x) \qquad (2.2)$$

4. Conocer los parámetros de un conjunto de entrenamiento para optimizar algunas funciones de perdida, como la estimación de máxima verosimilitud de un conjunto simulado $\{x_i, i = 1, 2, \dots, M\}$.

$$\Theta^* = argmax \sum_{i=1}^{M} \log P(x_i, \Theta). \qquad (2.3)$$

5. Visualizar el panorama de una función objetivo y cuantificar así la dificultad de una de las tareas anteriores y la eficiencia de varios algoritmos.

2.2 MCMC

Los MCMC son una técnica de propósito general, que nos ayuda a generar muestras a partir de una probabilidad en un espacio de dimensión grande, donde se generan números aleatorios extraídos de una distribución uniforme sobre [a, b]. La cadena de Markov simulada está diseñada para tener una función de distribución de probabilidad, $\pi(x)$, junto con su probabilidad estacionaria.

Una cadena de Markov $\{X_t\}$, sobre un espacio de estados general E, es una sucesión de variables aleatorias $\{X_t, t = 1,2, \dots, n\}$ que cambia de estado a estado en tiempo discreto (continuo) a partir del kernel de transición P. Este concepto tiene propiedades sobre una sigma algebra de Borel $\mathcal{B}$ (ver [6]). Las condiciones bajo las cuales la i-ésima iteración del kernel de transición converge a la distribución límite π cuando $t \to \infty$. Se necesitan las condiciones siguientes (ver [1]):

1. Invariante. La distribución invariante (o estacionaria) es aquella que satisface:

$$\pi(B) = \int P(y, B)\pi(dy), \quad \forall B \epsilon\, \mathcal{B}(E).$$

2. La cadena de Markov es irreducible y aperiódica.

La distribución invariante utiliza la condición de reversibilidad, aunque esta condición es suficiente, pero no es necesaria para asegurar la convergencia de la cadena a esta distribución. Una cadena de Markov es reversible si satisface la condición de equilibrio, $\pi(dx)P(x, dx) = \pi(dy)P(y, dx)$. Por último, la cadena simulada debe ser recurrente para asegurar la convergencia a la distribución límite.

Para obtener inferencia Bayesiana se suele utilizan los métodos Monte Carlo vía cadenas de Markov, podemos utilizar los valores simulados de esta cadena para poder obtener características del modelo. Observe que el uso de una cadena de un algoritmo MCMC con distribución límite π es similar al uso de una muestra independiente e idénticamente distribuida (iid) en el sentido de que el teorema ergódico garantiza la convergencia de la media empírica. Dos de los algoritmos más simples para esto, son el algoritmo Metropolis-Hastings y el muestreador de Gibbs, aunque estos métodos no son útiles cuando consideramos el espacio de estado como

una unión de subespacios de diferente dimensión.

2.2.1 Algoritmo Metropolis-Hastings

El algoritmo Metropolis-Hastings consiste en lo siguiente:

Suponga que la distribución de probabilidad de interés es dada por $\pi(x)$, $x^{(t)} \epsilon\ \Omega$ el estado actual de la cadena, y la distribución de probabilidad propuesta $q(x, x')$, entonces:

1. Proponemos un nuevo estado x´ mediante la distribución de propuesta $q(x^{(t)}, x')$.
2. Calculamos la probabilidad de aceptación:
$$\alpha(x^{(t)}, x') = min\left\{1, \frac{\pi(x')q(x',\ x^{(t)})}{\pi(x)q(x^{(t)}, x')}\right\}.$$
3. Con probabilidad $\alpha(x^{(t)}, x')$ aceptamos el movimiento y hacemos $x^{(t+1)} = x'$, en otro caso $x^{(t+1)} = x^{(t)}$.

2.2.1 Algoritmo muestreador de Gibbs.

El método del muestreador de Gibbs consiste en lo siguiente: Suponga que la distribución de probabilidad de interés es dada por $\pi(x)$, también el estado actual de la cadena ahora es de la forma $\boldsymbol{x}^{(t)} = (x, x_{2,\dots}, x_k) \in \Omega$ y sea $\boldsymbol{x}^{(t+1)} \in \Omega$ un nuevo estado de la cadena, para generar este estado se realiza lo siguiente

1. Seleccionamos una posición $i = \{1,2,\cdots, d\}$ aleatoriamente, tomando L valores $y_1, y_2, \dots, y_L$.
2. Calculamos la probabilidad condicional del vector $\boldsymbol{u} = (u_1, u_2, \dots, u_l)$, donde $u_k = \pi(x_i = v_k|\boldsymbol{x}_{-i})$, donde $\boldsymbol{x}_{-i}$ toma el vector completo excepto el valor $\boldsymbol{x}_i$.
3. Muestrear $j \sim u$ y el conjunto $x_{-i}^{(t+1)} = x_{-i}^{(t)},\ X_i^{(t+1)} = y_j$.

El orden de selección de las variables del paso 1, puede ser aleatorio o seguir un esquema definido.

3. Reversible Jump MCMC.

Otro algoritmo de gran utilidad que utiliza los métodos MCMC es conocido como Reversible Jump (saltos reversibles), que inicialmente fue propuesto por Green ([2]). Lo novedoso de esta propuesta es que la cadena que se genera por medio de esté algoritmo no necesita tener un espacio de estado fijo, es decir, que la dimensión del espacio de estado puede estar cambiando. De manera formal podemos decir que: si $\boldsymbol{\Omega} = \bigcup_{i=1}^{\infty} \Omega_i$ es el espacio de estados descrito como la unión de subespacios de diferente dimensión ($\dim(\Omega_i) = d_i$), y si π es la distribución de probabilidad definida sobre Ω, entonces la cadena simulada puede cambiar de $\Omega_i\ a\ \Omega_j$, y además se debe cumplir la ecuación de balanceo con respecto a π. El movimiento de reversible jump $q(x, x')$ que va de $\mathbf{x} \in \Omega_i$ a $x' \in \Omega_i$ se obtiene primero muestreando j a partir de una distribución de probabilidad $q(j|i, \boldsymbol{x})$ mediante un vector auxiliar $\boldsymbol{u} \in \mathbb{R}^m$ (donde m es alguna dimensión especificada) con función determinista $x' = f_1(\boldsymbol{x}, \boldsymbol{u})$. Un movimiento de reversa $q(x', x)$ se puede definir de manera similar, primero se realiza un muestreo para i con probabilidad $q(i|j, \boldsymbol{x}')$ y un vector auxiliar $\boldsymbol{u}' \in \mathbb{R}^{m'}$ con fd $x = f_2(x', u')$. Así debemos hallar una biyección $f: \Omega_i \times \mathbb{R}^m \to \Omega_j \times \mathbb{R}^{m'}$ tal que $f(\boldsymbol{x}, \boldsymbol{u}) = (\boldsymbol{x}', \boldsymbol{u}')$.

Con lo anterior, definimos una condición sobre igualdad de dimensiones, $d_i + m = d_j + m'$ que debe satisfacerse, así como $\frac{dx'du'}{dxdu} = \frac{\partial f(x,u)}{\partial(x,u)}$. Para satisfacer la ecuación de equilibrio, el movimiento propuesto $q(x, x')$ es aceptado con probabilidad (Green 1995):

$$\alpha(x, x') = \min\left\{1, \frac{\pi(\boldsymbol{x}')q(\boldsymbol{u}'|\ \boldsymbol{x}')q(i|j, \boldsymbol{x}')}{\pi(\boldsymbol{x})q(\boldsymbol{u}|\boldsymbol{x})q(j|i, \boldsymbol{x})} \left|\det\frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial(\boldsymbol{x}, \boldsymbol{u})}\right|\right\}.$$

4. Conclusión.

Como pudimos observar, las características principales del método de reversible jump no difieren en gran proporción a los algoritmos tradicionales, ya que, dado que es un MCMC, utiliza teoría respecto a cadenas de Markov y los métodos Monte Carlo. La parte quizás

complicada es hacer una buena definición de la función biyectiva que haga la conexión entre los subespacios de diferente dimensión, y que además cumpla la ecuación de equilibrio.

Referencias

Barbu Adrian; Song-Chun Zhu (2020). *Monte Carlo Methods*. Springer.

C. P. Robert and G. Casella (1999). *Monte Carlo statistical methods*. Springer, 1999.

Hastings WK; *Monte carlo sampling methods using markov chains and their applications*. page 97-109, 1970.

Peter J. Green (1995). *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination.* Biometrika.

Rosenbluth MN Teller AH Teller E Metropolis N (1953). Rosenbluth AW. *Equation of state calculations by fast computing machines.* page 1087-1092.

Sergio Hernández-Marín (2007). *Bayesian Analysis of Lidar Signals Using Reversible Jump Markov Chain Monte Carlo Algorithms*. Herior-Watt University.