Cómputo Científico para Probabilidad, Estadística y Ciencia de Datos

Ezau Faridh Torres Torres

TAREA 8: MCMC: MH con Kerneles Híbridos y Gibbs Sampler

Fecha de entrega: 06/Nov/2024.

NOTA: Los ejercicios se encuentran repartidos en los archivos:

- ejercicio1_tarea8.py
- ejercicio2_tarea8.py
- ejercicio3_tarea8.py

---

1. Aplique el algoritmo de Metropolis-Hastings considerando como función objetivo la distribución normal bivariada:

$$f_{X_1,X_2}(\bar{x}) = \frac{1}{2\pi} |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\bar{x}-\mu)'\Sigma^{-1}(\bar{x}-\mu)\right\} \tag{1}$$

donde

$$\mu = \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \quad \Sigma = \begin{pmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{pmatrix}. \tag{2}$$

Así, se tienen las siguientes distribuciones condicionales:

$$X_1|X_2 = x_2 \sim \mathcal{N}\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(x_2-\mu_2), \sigma_1^2(1-\rho^2)\right), \tag{3}$$

$$X_2|X_1 = x_1 \sim \mathcal{N}\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x_1-\mu_1), \sigma_2^2(1-\rho^2)\right). \tag{4}$$

Considere las siguientes propuestas:

$$q_1\left((x_1', x_2')|(x_1,x_2)\right) = f_{X_1|X_2}(x_1'|x_2)\cdot \mathbb{1}(x_2'=x_2), \tag{5}$$

$$q_2\left((x_1', x_2')|(x_1,x_2)\right) = f_{X_2|X_1}(x_2'|x_1)\cdot \mathbb{1}(x_1'=x_1). \tag{6}$$

A partir del algoritmo MH usando Kerneles híbridos simule valores de la distribución normal bivariada, fijando $\sigma_1 = \sigma_2 = 1$, considere los casos $\rho = 0.85$ y $\rho = 0.99$.

Respuesta:

En el archivo ejercicio1_tarea8.py, se comienza implementando la función *contornos()*, la cual recibe una cadena de Markov simulada y grafica su trayectoria. Además, recibe los parámetros de una distribución normal bivariada y genera su gráfica de contornos. Es útil para comparar la concentración de la cadena en la región de interés.

A continuación, se implementa la función *METROPOLIS_HASTINGS_HYBRID_KERNELS()*, la cual aplica el algoritmo Metropolis-Hastings con kérneles híbridos para K propuestas simulando una cadena de Markov en $\mathbb{R}^n$ y toma los siguientes argumentos:

- La función objetivo f (en este caso es la normal bivariada (1)).
- Una lista de funciones que generan propuestas $[q_{1_{gen}}, \ldots, q_{K_{gen}}]$, es decir, en este caso las expresadas en (5) y (6).
- Lista de funciones que calculan la densidad de probabilidad de las propuestas $[q_{1_{pdf}}, \ldots, q_{K_{pdf}}]$ (es decir, (5) y (6)). Es importante que se respete el mismo orden de la lista de funciones generadoras.
- Lista de probabilidades de seleccionar cada kernel de propuesta $[p_1, \ldots, p_K]$. Es importante que se respete el mismo orden de la lista de funciones generadoras y que $\sum_k p_k = 1$. Si no se da, se toman probabilidades iguales para cada kernel, i.e., $p_1 = \cdots p_K = \frac{1}{K}$.
- El valor inicial x_0 en el soporte de la distribución objetivo.
- El número de iteraciones del algoritmo (casi siempre se usa $N = 10,000$).

La función regresa la cadena simulada en $\mathbb{R}^n$ e imprime la tasa de aceptación lograda y el conteo de aceptaciones por kernel. A continuación, se define la función *objetivo()* la cual implementa la expresión (1) y las propuestas:

- *prop1_gen()* y *prop1_pdf()*
- *prop2_gen()* y *prop2_pdf()*

las cuales implementan las propuestas (5) y (6), así como su densidad de probabilidad respectiva. En estas funciones es importante notar que se respeta cuando se deja fija alguna de las variables ($x_2' = x_2$ en la propuesta 1 y $x_1' = x_1$ en la propuesta 2).

Finalmente, se ejecuta el código principal, en el cual se definen los parámetros para la normal bivariada objetivo: el problema pide tomar $\sigma_1 = \sigma_2 = 1$ y para elegir μ_1 y μ_2, se eligieron de forma uniforme en $[0, 10]$, resultando en $\mu_1 = 7.216$ y $\mu_2 = 0.077$ aunque se puede elegir donde sea necesario.

El punto x_0 inicial se tomó en $[10, 2.5]$ y aunque teóricamente puede tomarse donde sea debido a que el soporte de la normal bivariada es todo $\mathbb{R}^2$, si se aleja mucho del punto $[\mu_1, \mu_2]$, puede causar indeterminaciones. Para el algoritmo Metropolis-Hastings, se usaron $100,000$ muestras y a ambas propuestas se les dio probabilidad de elección igual a $\frac{1}{2}$.

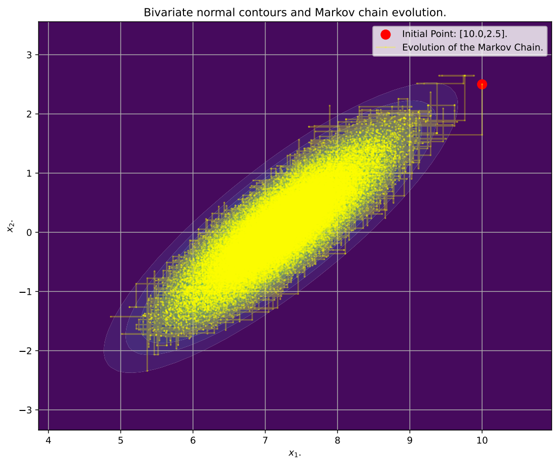


Figure 1: $\rho = 0.85$.

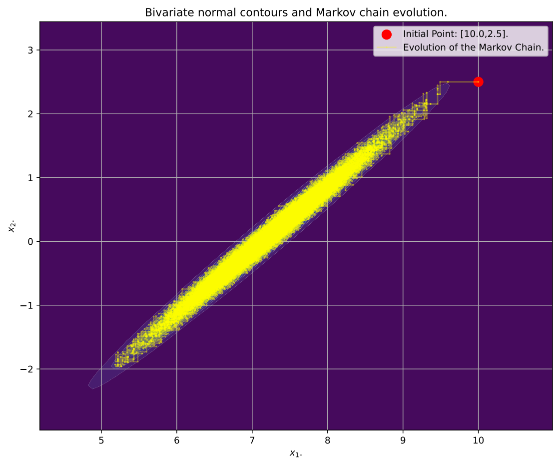


Figure 2: $\rho = 0.99$.

Finalmente, para cada valor de $\rho \in \{0.85, 0.99\}$, se generaron las listas de funciones generadoras y sus respectivas densidades para aplicar el algoritmo Metropolis Hastings con Kérneles Híbridos generando los resultados mostrados en la figura anterior.

Sabemos que en una distribución normal bivariada, el coeficiente de correlación ρ controla la dependencia lineal entre las dos variables aleatorias, afectando la forma de los contornos de la distribución conjunta. Es por esto que podemos notar que a medida que ρ se acerca a 1, la dispersión en la dirección perpendicular a la correlación disminuye y la distribución se concentra más a lo largo de una línea, lo que explica el efecto de "concentración" que se observa en los datos simulados para $\rho = 0.99$ en comparación con $\rho = 0.85$.

2. Considere los tiempos de falla $t_1, \dots, t_n$ con distribución $Weibull(\alpha, \lambda)$:

$$f(t_i|\alpha, \lambda) = \alpha\lambda t_i^{\alpha-1} e^{-t_i^{\alpha}\lambda} \tag{7}$$

Se asumen como a priori $\alpha \sim exp(c)$ y $\lambda|\alpha \sim Gama(\alpha, b)$, por lo tanto, $f(\alpha, \lambda) = f(\lambda|\alpha)f(\alpha)$. Así, para la distribución posterior se tiene:

$$f(\alpha, \lambda|\bar{t}) \propto f(\bar{t}|\alpha, \lambda)f(\alpha, \lambda) \tag{8}$$

A partir del algoritmo MH usando Kerneles híbridos simule valores de la distribución posterior $f(\alpha, \lambda|\bar{t})$, considerando las siguientes propuestas:

- Propuesta 1:

$$\lambda_p|\alpha, \bar{t} \sim Gama\left(\alpha + n, b + \sum_{i=1}^{n} t_i^{\alpha}\right) \text{ y dejando } \alpha \text{ fijo.} \tag{9}$$

- Propuesta 2:

$$\alpha_p|\lambda, \bar{t} \sim Gama\left(n + 1, -\log b - \log r_1 + c\right) \text{ con } r_1 = \prod_{i=1}^{n} t_i \text{ y dejando } \lambda \text{ fijo.} \tag{10}$$

- Propuesta 3:

$$\alpha_p \sim exp(c) \text{ y } \lambda_p|\alpha_p \sim Gama(\alpha_p, b) \tag{11}$$

- Propuesta 4 (RWMH):

$$\alpha_p = \alpha + \epsilon, \text{ con } \epsilon \sim \mathcal{N}(0, \sigma) \text{ y dejando } \lambda \text{ fijo.} \tag{12}$$

Simular datos usando $\alpha = \lambda = 1$ con $n = 30$. Para la a priori usar $c = 1$ y $b = 1$.

Respuesta:

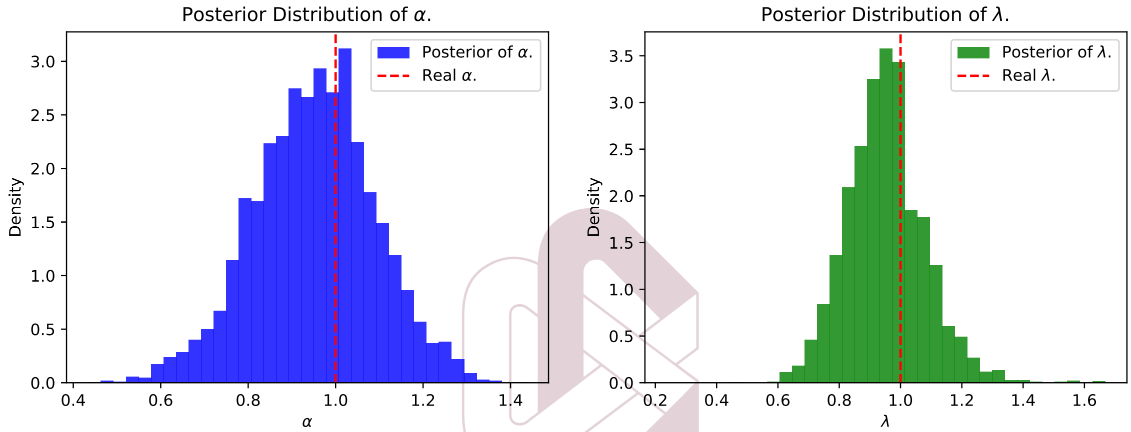
En el archivo ejercicio2_tarea8.py, se comienza implementando la función *plot_chains()*, la cual recibe una cadena de Markov simulada y grafica el histograma de cada variable, además, traza los valores reales de α y β, que en ejercicio se piden ambos igual a 1. Gracias a lo general que es la función *METROPOLIS_HASTINGS_HYBRID_KERNELS()* descrita en el ejercicio anterior, se vuelve a usar sin hacer ninguna modificación.

A continuación, se define la función *posterior()* la cual implementa la expresión (8) y las propuestas:

- *prop1_gen()* y *prop1_pdf()*
- *prop2_gen()* y *prop2_pdf()*
- *prop3_gen()* y *prop3_pdf()*
- *prop4_gen()* y *prop4_pdf()*

las cuales implementan las propuestas (9), (10), (11) y (12) así como su densidad de probabilidad respectiva. En estas funciones es importante notar que se respeta cuando se deja fija alguna de las variables (α en la propuesta 1 y λ en la propuesta 2 y 4).

Finalmente, se ejecuta el código principal, en el cual se definen los parámetros dados por el ejercicio, como $\alpha = \lambda = b = c = 1$ y $n = 30$. Además, para la propuesta 4, se usa $\sigma = 0.1$ gracias a que se obtuvieron buenos resultados con este valor. El punto x_0 inicial se tomó aleatorio en $[0, 1]$ por simplicidad y se usaron $10,000$ iteraciones del algoritmo. Además, se le dio igual probabilidad de elección a cada una de las propuestas: $\frac{1}{4}$. Así, se generaron las observaciones t de la distribución Weibull para luego generar las listas de funciones generadoras y sus respectivas densidades para aplicar el algoritmo Metropolis Hastings con Kérneles Híbridos generando los resultados mostrados a continuación:



Podemos notar que las estimaciones para α y λ fueron bastante buenas, ya que las distribuciones tienen sus medias muy cerca a los valores reales $\alpha = \lambda = 1$. Este resultado se repitió en varias ocasiones para distintos valores de los parámetros y número de iteraciones, en particular, parecía invariante a la elección de la desviación estándar σ de la propuesta 4.

3. Considere el ejemplo referente al número de fallas de bombas de agua en una central nuclear, donde p_i representa el número de fallas en el tiempo de operación t_i, con $i = 1, \ldots, n$.

   Se considera el modelo $p_i \sim Poisson(\lambda_i t_i)$, (las λ_i son independientes entre si), con distribuciones a priori $\lambda_i|\beta \sim Gama(\alpha, \beta)$ y $\beta \sim Gama(\gamma, \delta)$, por lo tanto:

   $$f(\lambda_1, \ldots, \lambda_n, \beta) = f(\lambda_1|\beta)f(\lambda_2|\beta)\cdots f(\lambda_n|\beta)f(\beta) \tag{13}$$

   Para la distribución posterior se tiene:

   $$f(\lambda_1, \ldots, \lambda_n, \beta|\bar{p}) \propto L(\bar{p}, \bar{\lambda}, \beta)f(\lambda_1, \ldots, \lambda_n, \beta) \tag{14}$$

   Simule valores de la distribución posterior $f(\lambda_1, \ldots, \lambda_n, \beta|\bar{p})$ usando un kernel híbrido, considerando las propuestas:

   $$\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t} \sim Gama(p_i + \alpha, \beta + t_i) \tag{15}$$

   $$\beta|\bar{\lambda}, \bar{t} \sim Gama\left(n\alpha + \gamma, \delta + \sum_{i=1}^{n} \lambda_i\right) \tag{16}$$

   Verifique que estas son propuestas Gibbs.

   Use los datos del Cuadro 1 con los parámetros a priori $\alpha = 1.8$, $\gamma = 0.01$ y $\delta = 1$.

Bomba (i)	1	2	3	4	5	6	7	8	9	10
T. de uso (t_i)	94.32	15.72	62.88	125.76	5.24	31.44	1.05	1.05	2.1	10.48
# de fallas (p_i)	5	1	5	14	3	20	1	1	4	22

Table 1: Datos de bombas de agua en centrales nucleares (Robert y Casella, p. 385) para el ejemplo 8.3.

Respuesta:

De forma muy similar a los ejercicios anteriores, en el archivo ejercicio3_tarea8.py, se comienza implementando la función *plot_chains()*, la cual recibe una cadena de Markov simulada y grafica el histograma de cada variable, en este caso, se imprimen todas las λ_i juntas y el correspondiente a β aparte. Una vez más, gracias a lo general que es la función *METROPOLIS_HASTINGS_HYBRID_KERNELS()* descrita en el ejercicio 1, se vuelve a usar sin hacer ninguna modificación.

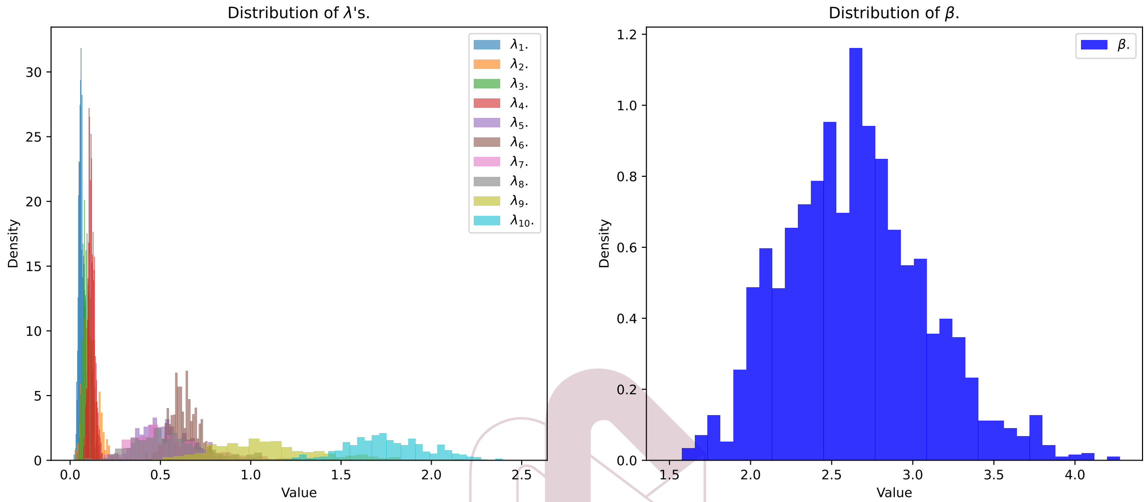
A continuación, se define la función *posterior()* la cual implementa la expresión (14) y las propuestas:

- *prop_lambda_i_gen()* y *prop_lambda_i_pdf()*
- *prop_beta_gen()* y *prop_beta_pdf()*

las cuales implementan las propuestas (15) y (16) así como su densidad de probabilidad respectiva. Notemos que se tiene una función para λ_i, al final se hace una lista para contemplar a todos los $i \in \{1, \ldots, 10\}$

Finalmente, se definen los tiempos de operación t y el número de fallas correspondientes p dados en la tabla y se ejecuta el código principal, en el cual se definen los parámetros dados por el ejercicio, como $\alpha = 1.8$, $\gamma = 0.01$ y $\delta = 1$. El punto x_0 inicial se tomó aleatorio en $[0, 1]$ por simplicidad y se usaron $15,000$ iteraciones del algoritmo, aunque al generar los histogramas finales, se retiró un burn-in de 2000 puntos. También, se le dio igual probabilidad de elección a cada una de las propuestas: $\frac{1}{11}$.

Así, se generaron las listas de funciones generadoras y sus respectivas densidades (terminando con 11 funciones generadoras, 10 para λ_i y la de β) para aplicar el algoritmo Metropolis Hastings con Kérneles Híbridos, generando los resultados mostrados a continuación:



Tras analizar los resultados generados, la simulación de la distribución posterior $f(\lambda_1, \ldots, \lambda_n, \beta|\bar{p})$ utilizando el kernel híbrido parece haber proporcionado estimaciones razonables para los parámetros λ_i y β.

Las gráficas de densidad para los valores de λ_i muestran variabilidad entre las bombas. Esto refleja que las tasas de falla son distintas para cada bomba, posiblemente debido a las diferencias en el tiempo de operación (t_i) y el número de fallas observadas (p_i). Los valores de λ_i para bombas con más fallas y mayor tiempo de operación tienden a ser mayores, lo cual es consistente con el modelo de Poisson.

La distribución posterior de β tiene una densidad concentrada, lo que indica que las observaciones proporcionan información suficiente para estimar este parámetro. Esto sugiere que el valor de β es estable.

Demostración:

- Propuesta para $\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t}$:

Sabemos que $p_i \sim Poisson(\lambda_i t_i)$ y que la distribución a priori de $\lambda_i|\beta$ es $Gama(\alpha, \beta)$. Entonces, la distribución posterior completa de λ_i (condicional en β y los datos observados) se obtiene como sigue:

$$f(\lambda_i|p_i, t_i, \beta) \propto f(p_i|\lambda_i, t_i) f(\lambda_i|\beta) \tag{17}$$

Sustituyendo las distribuciones, se obtiene:

$$f(\lambda_i|p_i, t_i, \beta) \propto \frac{(\lambda_i t_i)^{p_i} e^{-\lambda_i t_i}}{p_i!} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \tag{18}$$

Agrupando términos de λ_i, tenemos:

$$f(\lambda_i|p_i, t_i, \beta) \propto \lambda_i^{p_i+\alpha-1} e^{-(\beta+t_i)\lambda_i} \tag{19}$$

Esta es la forma de una distribución $Gama(p_i+\alpha, \beta+t_i)$, lo cual es precisamente la propuesta indicada en el ejercicio:

$$\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t} \sim Gama(p_i+\alpha, \beta+t_i) \tag{20}$$

Esto demuestra que la propuesta para λ_i es la distribución condicional completa, por lo que es una propuesta Gibbs.

- Propuesta para $\beta|\bar{\lambda}, \bar{t}$:

La distribución a priori de β es $Gama(\gamma, \delta)$, y dado que $\lambda_i|\beta \sim Gama(\alpha, \beta)$ para cada i, la posterior completa de β se obtiene multiplicando estas distribuciones. La verosimilitud conjunta de λ_i dado β es:

$$f(\lambda_1, \ldots, \lambda_n|\beta) = \prod_{i=1}^{n} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \tag{21}$$

Multiplicando por el prior $f(\beta) \propto \beta^{\gamma-1} e^{-\delta\beta}$, la distribución posterior de β es:

$$f(\beta|\bar{\lambda}, \bar{t}) \propto \beta^{n\alpha+\gamma-1} e^{-(\delta+\sum_{i=1}^{n}\lambda_i)\beta} \tag{22}$$

Esto es una $Gama(n\alpha+\gamma, \delta+\sum_{i=1}^{n}\lambda_i)$, que corresponde exactamente a la propuesta:

$$\beta|\bar{\lambda}, \bar{t} \sim Gama(n\alpha+\gamma, \delta+\sum_{i=1}^{n}\lambda_i) \tag{23}$$

Así, también hemos demostrado que esta propuesta para β es su distribución condicional completa, confirmando que es una propuesta Gibbs.