Cómputo Científico para Probabilidad, Estadística y Ciencia de Datos

Ezau Faridh Torres Torres

TAREA 6: MCMC: Metropolis-Hastings

Fecha de entrega: 16/Oct/2024.

NOTA: Los ejercicios se encuentran repartidos en los archivos:

- ejercicio1_tarea6.py
- ejercicio2_tarea6.py
- ejercicio4_tarea6.py

---

1. Simular $n = 5$ y $n = 50$ v.a's Bernoulli $Be\left(\frac{1}{3}\right)$; sea r el número de éxitos en cada caso.

Respuesta:

En el archivo ejercicio1_tarea6.py se implementa la función *SIMULAR_BERNOULLI()* con la cual, se simulan $n = 5$ y $n = 50$ variables aleatorias Bernoulli $Be(\frac{1}{3})$ y se calcula el número de éxitos en cada caso. Se utiliza la función *np.random.binomial()* de la librería *numpy* para simular las variables aleatorias binomiales ya que para el caso de una sola prueba es equivalente a una distribución Bernoulli. Se obtuvieron los siguientes resultados (se usó una semilla para reproducibilidad):

- Éxitos para $n = 5 : 2$.
- Éxitos para $n = 50 : 15$.

---

2. Implementar el algoritmo Metropolis-Hastings para simular de la posterior.

$$f(p|\bar{x}) \propto p^r(1-p)^{n-r}\cos(\pi p)\cdot \mathbb{1}_{[0,\frac{1}{2}]}(p). \tag{1}$$

con los dos casos de n y r de arriba. Para ello poner la propuesta $(p'|p) = p' \sim Beta(r+1, n-r+1)$ y la distribución inicial de la cadena $\mu \sim \mathcal{U}\left(0, \frac{1}{2}\right)$.

Respuesta:

En el archivo ejercicio2_tarea6.py se implementa la función *METROPOLIS_HASTINGS()*, la cual simula una cadena de Markov utilizando el algoritmo de Metropolis-Hastings y toma los siguientes argumentos:

- La función objetivo $f()$, (en este caso es la posterior (1)).
- La distribución propuesta $q()$ (en este caso es la distribución $Beta(r+1, n-r+1)$).
- El valor inicial x_0 (en este caso es $\mu \sim \mathcal{U}\left(0, \frac{1}{2}\right)$.)
- El número de iteraciones (por default se toma $N = 10000$).

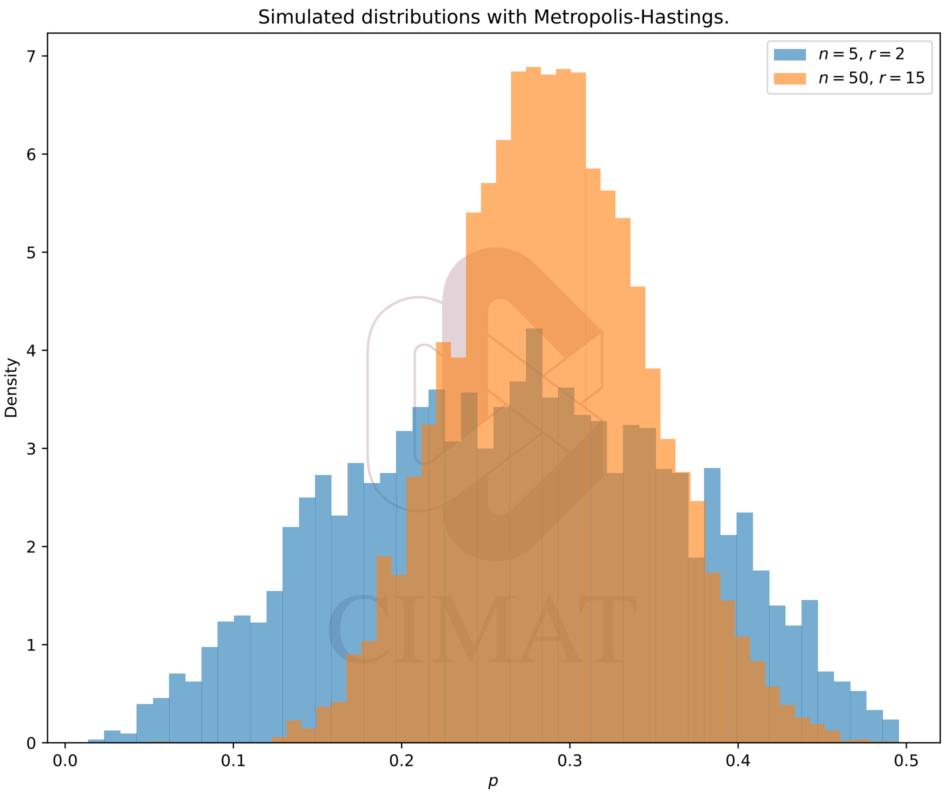
Además, regresa la cadena de Markov simulada. Usa el criterio de aceptación: si y_t es la propuesta dada por $q(\cdot|x_t)$ en x_t, entonces se acepta y_t con probabilidad $\rho(x_t, y_t)$ con

$$\rho(x,y) = \min\left\{1, \frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}\right\} \tag{2}$$

y se rechada con probabilidad $1-\rho(x_t, y_t)$. A continuación, se define la función posterior (1) llamada *posterior()*, y la función *propuesta()* dada por $Beta(r+1, n-r+1)$ la cual, como vimos en clase, no depende del estado actual x_t.

Finalmente, se hizo la simulación usando la función la distribución posterior (1) para $n=5$ y $n=50$ usando el algoritmo de Metropolis-Hastings haciendo uso de la función *SIMULAR_BERNOULLI()* del ejercicio 1. Se comparan las distribuciones obtenidas con un histograma y se obtuvo el siguiente resultado para una muestra de tamaño $N=10000$:

- Éxitos para $n=5:2$.
- Éxitos para $n=50:15$.



Podemos notar que en ambas simulaciones para n y r, se obtuvieron buenas estimaciones de $p=\frac{1}{3}$ ya que, $\frac{n}{r}$ no resulta tan alejado de $\frac{1}{3}$ para cada caso (se usó una semilla para reproducibilidad).

NOTA: Los comentarios extras se dan al final del ejercicio 3.

---

3. Argumentar porque la cadena es $f-$irreducible y porque es ergódica. Implementar el algoritmo con los datos descritos y discutir los resultados.

Respuesta:

Sabemos que la $f-$irreducibilidad de una cadena de Markov significa que es posible alcanzar cualquier subconjunto del espacio de estados a partir de cualquier estado con probabilidad positiva, donde f es una medida no trivial en el espacio de estados. En el contexto del algoritmo Metropolis-Hastings con nuestra propuesta

$$p^{'}|p \sim Beta(r+1, n-r+1), \tag{3}$$

la cadena es $f-$irreducible por las siguientes razones:

- **Soporte de la propuesta:** La distribución $Beta(r+1, n-r+1)$ tiene soporte en $[0,1]$ (aunque la posterior tiene soporte restringido a $[0, \frac{1}{2}]$ y claramente $[0, \frac{1}{2}] \subset [0,1]$). Esto significa que, desde cualquier valor $p \in (0, \frac{1}{2})$, la propuesta tiene probabilidad positiva de moverse a cualquier otro punto en este intervalo.
- **Aceptación de la propuesta:** La densidad objetivo $f(p|\bar{x})$ es continua y positiva en el intervalo $(0, \frac{1}{2})$ (aunque puede ser cercana a cero en los extremos). Esto asegura que hay una probabilidad positiva de aceptar cualquier propuesta $p^{'}|p$ si proviene de la distribución $Beta(r+1, n-r+1)$, siempre que esté dentro de $(0, \frac{1}{2})$.

Por lo tanto, desde cualquier estado p , es posible alcanzar cualquier otro estado en $(0, \frac{1}{2})$ con probabilidad positiva, lo que asegura que la cadena es $f-$irreducible $\blacksquare$

Ahora, la ergodicidad de la cadena se refiere a que, con el tiempo, la cadena "visita" todos los estados posibles de manera que las frecuencias de visita convergen a las probabilidades estacionarias de la distribución objetivo $f(p|\bar{x})$. En clase revisamos que, el que la cadena sea $f-$irreducible entonces era ergódica, sin embargo, se va a detallar más:

La cadena es ergódica por las siguientes razones:

- **Irreducibilidad:** Ya se revisó que la cadena es $f-$irreducible, lo que es una condición necesaria para la ergodicidad.
- **Aperiodicidad:** La propuesta Beta no es determinista, lo que implica que no hay ciclos predefinidos en la cadena. Esto significa que la cadena no quedará atrapada en patrones cíclicos, y por tanto es aperiodica (en clase vimos que dada la aperiodicidad, la cadena convergía a la distribución objetivo).
- **Acotamiento del soporte:** La posterior (1) tiene soporte en $[0, \frac{1}{2}]$, lo que es un conjunto acotado. En este intervalo, la distribución Beta y la densidad objetivo permiten que la cadena se mueva de manera adecuada por todo el espacio, y la distribución $f(p|\bar{x})$ actúa como la distribución estacionaria $\blacksquare$

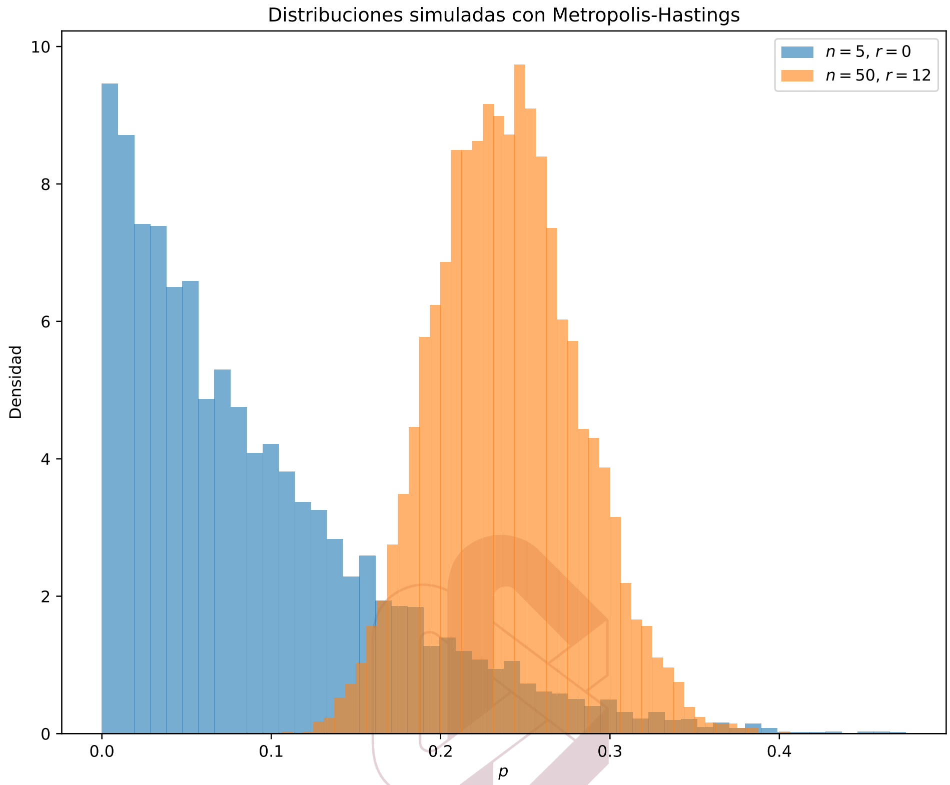
Resultados:

Ya se implementó el algoritmo con los datos descritos en el ejercicio anterior en el archivo ejercicio2_tarea6.py y se tienen los siguientes resultados/comentarios.

Como se mencionó antes, se puede notar que en ambas simulaciones para n y r, se obtuvieron buenas estimaciones de $p = \frac{1}{3}$ ya que, $\frac{n}{r}$ no resulta tan alejado de $\frac{1}{3}$ para cada caso. Variando la semilla, se notó en algunas ocasiones un mal desempeño al momento de estimar $p = \frac{1}{3}$, esto dependiento de la simulación de r para $n = 5$ o $n = 50$, ya que para ser un mejor estimador, r debe ser $\frac{n}{3}$ aproximadamente. Un ejemplo de esto se puede ver en la siguiente figura:

Además, cuando n es pequeño (como $n = 5$), la distribución simulada tiende a estar más dispersa. Esto se debe a que la cantidad de datos disponibles (5) no es suficiente para ajustar fuertemente los parámetros de la posterior. Cuando n es mayor, la distribución se ajusta más cerca de los valores que maximizan la posterior, lo que refleja una mayor precisión en la estimación del parámetro p debido a la mayor cantidad de datos.

Finalmente, dado que la cadena es ergódica, tras un número suficientemente grande de iteraciones, las muestras convergen a la distribución posterior $f(p|\bar{x})$. Esto se observa en la forma de los histogramas que se alinean con las formas teóricas esperadas de las distribuciones.



4. Implementar el algoritmo Metropolis-Hastings con la posterior de arriba tomando una propuesta diferente.

Respuesta:

Dentro de la bibliografía más importante relacionada con Metropolis-Hastings, se encontró que un ejemplo común de propuesta es usar una distribución normal simétrica alrededor del valor actual p, como

$$p^{'}|p \sim \mathcal{N}(p, \sigma^2), \tag{4}$$

truncada en el intervalo $[0, \frac{1}{2}]$. Esto garantiza que las propuestas siempre estén dentro del dominio de la posterior. En el archivo ejercicio4_tarea6.py se usa la misma estructura del programa implementado en el ejercicio 2, con la diferencia de que se modificó la propuesta $(p^{'}|p) = p^{'} \sim Beta(r+1, n-r+1)$ por (4) con $\sigma^2 = 0.1$ para controlar la dispersión de las propuestas.

Se usó la misma distribución inicial de la cadena $\mu \sim \mathcal{U}\left(0, \frac{1}{2}\right)$ y se obtuvieron los siguientes resultados:

- Éxitos para $n = 5 : 2$.
- Éxitos para $n = 50 : 15$.

Además, experimentando con varias configuraciones, como variando el parámetro σ^2 y el número de muestras n, se pudo notar que, al igual que antes cuando n crece, se tiene que la distribución se ajusta más cerca de los valores que maximizan la posterior, lo que refleja una mayor precisión en la estimación del parámetro p debido a la mayor cantidad de datos y que funciona mejor con un valor de σ^2 pequeño. Se puede notar que la distribución propuesta (4), también cumple con las condiciones que hacen que la cadena sea $f-$irreducible y ergódica por las mismas razones mencionadas en el ejercicio 3.