Simulación Monte Carlo y Cadenas de Markov

Módulo 4 : Técnicas computacionales avanzadas para modelar fenómenos sociales
Concentración en Economía Aplicada y Ciencia de Datos
ITESM

31 de octubre de 2022

Método de Monte Carlo

Idea: Método para calcular el área bajo una curva. Es una solución estadística al problema de integración.

Supongamos que existe $M > 0$ tal que $0 \leq f(\theta) \leq M$ para todo $\theta \in [a, b]$ y que queremos calcular la integral

$$I = \int_a^b f(\theta) d\theta \tag{1}$$

el valor de la integral es el área bajo la curva $\phi = f(\theta)$ para $\theta \in [a, b]$. Dicha gráfica queda inscrita en el rectángulo $R = [a, b] \times [0, M]$.

Método de Monte Carlo

Sea

$$p(\theta, \phi) = \frac{1}{M(b-a)} I_R(\theta, \phi) \tag{2}$$

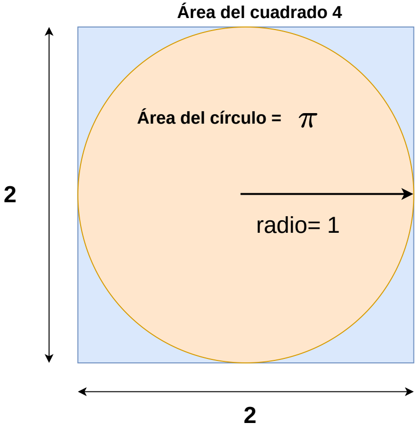
Entonces $p(\theta, \phi)$ corresponde a la función de densidad de una distribución uniforme sobre el rectángulo R. La integral I puede entonces estimarse simulando una muestra $(\theta_1, \phi_1), \ldots, (\theta_N, \phi_N)$ de $p(\theta, \phi)$ y contando cuántos de estos valores caen bajo la curva $\phi = f(\theta)$.

El estimador $\hat{I}$ obtenido es un estimador insesgado de I.

La varianza del estimador es

$$Var(\hat{I}) = \frac{I}{N}\{M(b-a) - I\}$$

Cálculo de π por el método de Monte Carlo (Wilkinson and Allen, 1998)



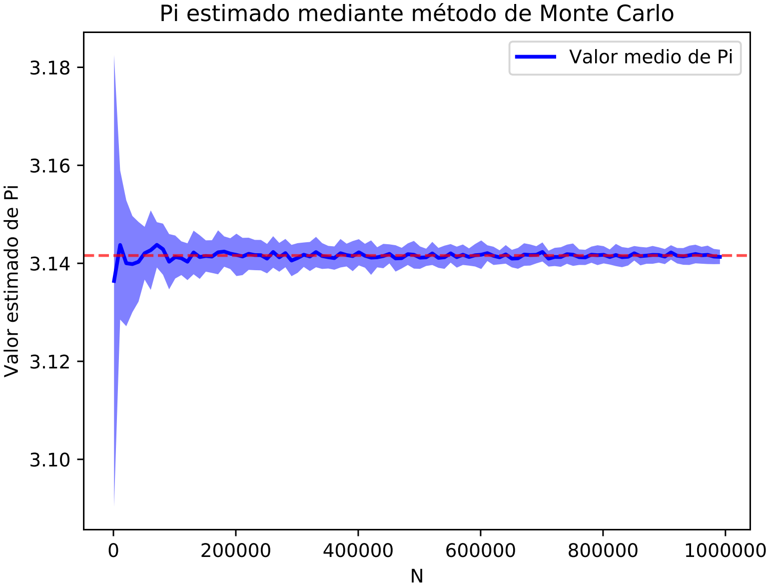
$$\frac{\text{Área del círculo}}{\text{Área del cuadrado}} = \frac{\pi(1)^2}{4} = \frac{\pi}{4}$$

que puede ser descrita por la integral

$$\int_0^1 \sqrt{1-x^2}dx = \frac{\pi}{4}$$

Para calcular π se generan parejas aleatorias de números (x_r, y_r), distribuidos uniformemente entre 0 y 1, y contamos cuántos de estos puntos caen dentro del círculo, esto es, si se cumple la igualdad

$$y_r^2 + x_r^2 \leq 1$$


Pi estimado mediante método de Monte Carlo
Valor medio de Pi
3.18
3.16
3.14
3.12
3.10
Valor estimado de Pi
0
200000
400000
600000
800000
1000000
N

Inferencia bayesiana

- En un contexto de inferencia bayesiana, donde se tiene un modelo muestral $Y \sim p(y|\theta)$ y una distribución inicial $p(\theta)$, la distribución final se calcula de la siguiente forma:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(\theta')p(y|\theta')d\theta'}$$

- Aún descartando el denominador (el cual se puede interpretar como un factor de escalamiento que hace que la función integre a 1), el cálculo analítico de $p(\theta|y)$, en la mayoria de las ocasiones, es complicado.

- Hay distintas técnicas para calcular $p(\theta|y)$ (aproximación de Laplace, métodos de cuadratura, etc), pero por la capacidad de cómputo disponible las técnicas de simulación de Monte Carlo vía cadenas de Markov (MCMC) han sido mayormente utilizadas.
- La idea de MCMC es **construir una cadena de Markov que sea fácil de simular (a través de un proceso de muestreo) y cuya distribución de equilibrio corresponda a la distribución final de interés**.

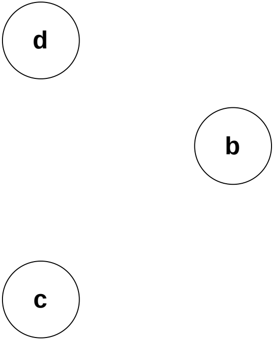
Cadenas de Markov

Definition

Cadena de Markov. Proceso estocástico discreto con estados finitos que cumple la propiedad de Markov.

$$P(X_{t+1} = s | X_1 = s_1, \ldots, X_t = s_t) = P(X_{t+1} = s | X_t = s_t)$$

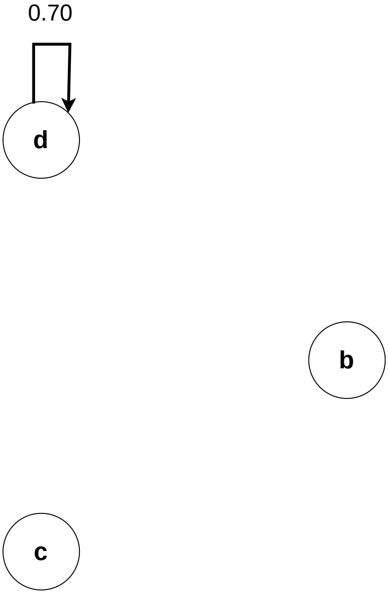
Cadenas de Markov



Estados:

d: Dormitorio
b: Bar
c: Comedor

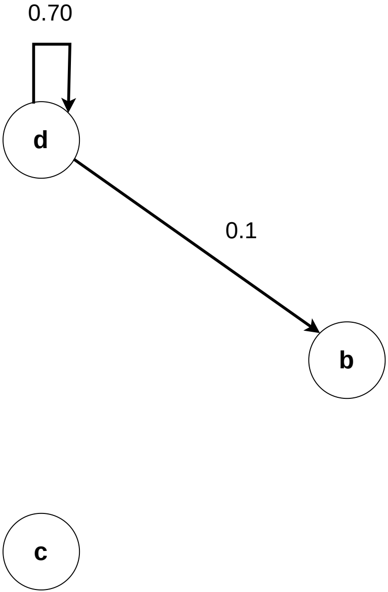
Ejemplo tomado de



La arista representa la posibilidad que un pasajero que está en el dormitorio en el tiempo t continúe en el dormitorio en el tiempo $t+1$.

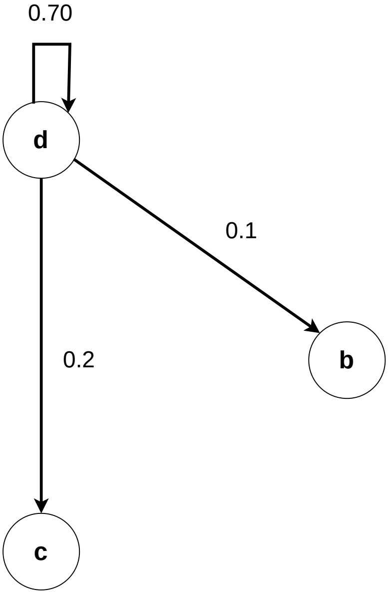
Se etiqueta con la probabilidad de que esto suceda (**Probabilidad de transición**)

$$P(X_{t+1} = d \mid X_t = d) = 0{,}7$$



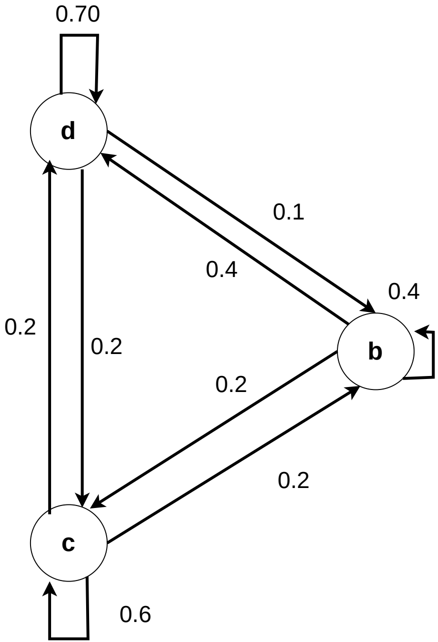
Arista para la transición del dormitorio al bar

$$P(X_{t+1} = b \mid X_t = d) = 0{,}1$$

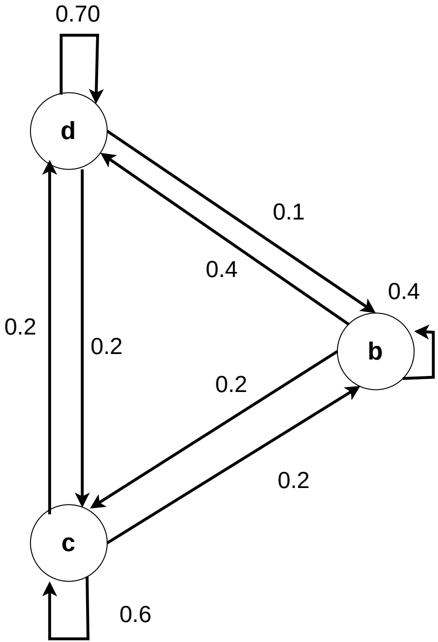


Arista para la transición del dormitorio al comedor

$$P(X_{t+1} = c \mid X_t = d) = 0{,}2$$



Agregamos todas las transiciones posibles y sus probabilidades.

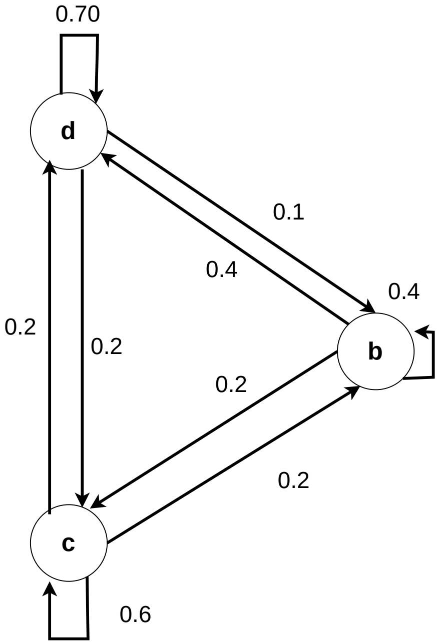


Las probabilidades de transición pueden representarse de manera matricial:

$$T = \begin{bmatrix} 0{,}7 & 0{,}4 & 0{,}2 \\ 0{,}1 & 0{,}4 & 0{,}2 \\ 0{,}2 & 0{,}2 & 0{,}6 \end{bmatrix} \tag{3}$$

con entradas

$$p_{i,j} \equiv P(X_{t+1} = i \mid X_t = j)$$



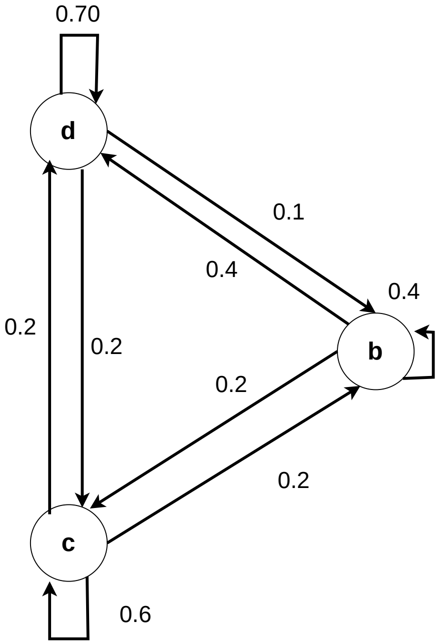
Matriz de probabilidad de estado:

$$T = \begin{bmatrix} 0{,}7 & 0{,}4 & 0{,}2 \\ 0{,}1 & 0{,}4 & 0{,}2 \\ 0{,}2 & 0{,}2 & 0{,}6 \end{bmatrix}$$

La dinámica para la distribución de probabilidad puede expresarse:

$$\begin{bmatrix} P(X_{t+1} = d) \\ P(X_{t+1} = b) \\ P(X_{t+1} = c) \end{bmatrix} = T \begin{bmatrix} P(X_t = d) \\ P(X_t = b) \\ P(X_t = c) \end{bmatrix} \tag{4}$$

$$P(X_{t+1}) = TP(X_t) \tag{5}$$



Si sabemos que el pasajero se encuentra en el dormitorio al tiempo $t = 0$:

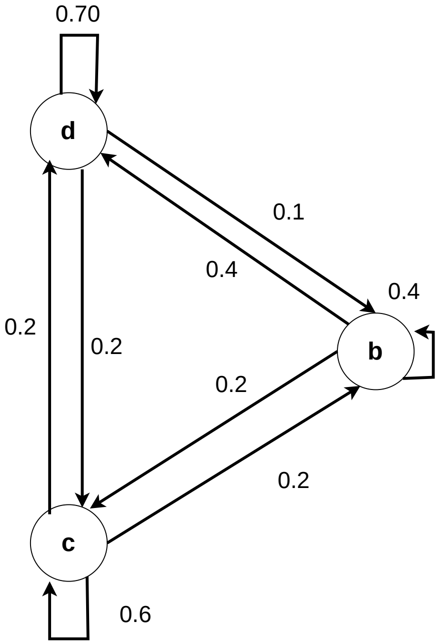
$$P(X_0) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = P_0$$

Al tiempo siguiente la distribución es:

$$P(X_1) = TP_0 = P_1$$

$$P(X_1) = \begin{bmatrix} 0{,}7 & 0{,}4 & 0{,}2 \\ 0{,}1 & 0{,}4 & 0{,}2 \\ 0{,}2 & 0{,}2 & 0{,}6 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$P(X_1) = \begin{bmatrix} 0{,}7 \\ 0{,}1 \\ 0{,}2 \end{bmatrix}$$



Para tiempos futuros

$$P(X_t) = TP(X_{t-1}) \equiv P_t$$

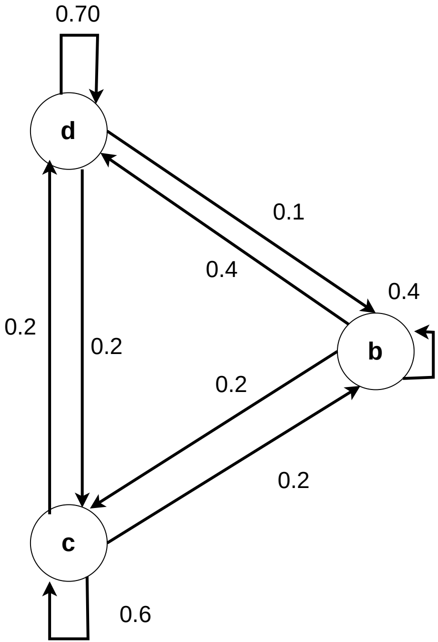
$$P1 = TP_0$$

$$P2 = TP_1$$

$$\begin{bmatrix} 0{,}7 & 0{,}4 & 0{,}2 \\ 0{,}1 & 0{,}4 & 0{,}2 \\ 0{,}2 & 0{,}2 & 0{,}6 \end{bmatrix} \begin{bmatrix} 0{,}7 \\ 0{,}2 \\ 0{,}1 \end{bmatrix}$$

$$= \begin{bmatrix} 0{,}57 \\ 0{,}15 \\ 0{,}28 \end{bmatrix}$$

$$TTP_0 = T^2 P_0$$



Para tiempos futuros

$$P_t = T^t P_0$$

Teorema Perron-Frobenius

Un proceso de Markov converge a un equilibrio estadístico único si cumple las siguientes condiciones

- **Conjunto finito de estados**: $S = \{1, 2, \ldots, K\}$
- **Reglas de transición fijas**: La matriz de transición no cambia en el tiempo.
- **Ergodicidad (accesibilidad entre estados)**: El sistema puede alcanzar cualquier estado desde cualquier estado a través de una serie de transiciones.
- **No ciclicidad**: El sistema no produce un ciclo a través de una secuencia de estados.

Algoritmo de Metropolis (Theodoridis, 2020)

- La distribución propuesta cambia con el tiempo siguiendo la evolución de una cadena de Markov.
- La cadena se construye de manera que su matriz de transición tenga la distribución deseada $p(x)$ la cual es invariante.
- La distribución propuesta depende del valor del estado previo,x_{n-1}, esto es, $q(\cdot|x_{n-1})$.
- Es decir, generar una nueva muestra (un nuevo estado) depende del valor del estado previo.

Algorithm 1: Algoritmo Metropolis (Theodoridis, 2020)

Sea la distribución deseada $p(\cdot)$
Escoge una distribución de propuesta $q(\cdot|\cdot)$
Escoge el valor del estado inicial x_0

for $n = 1, 2, \ldots, N$ **do**
    Toma un valor $x \sim q(\cdot|x_{n-1})$
    Calcula el valor de aceptación
    `/* Si la probabilidad de` $p(x)$ `es más grande que la de` $p(x_{n-1})$`, entonces se acepta la nueva muestra. En caso contrario, esta es aceptada-rechazada en base a su valor relativo */`

$$\alpha(x|x_{n-1}) = \min\left\{1, \frac{p(x)}{p(x_{n-1})}\right\}$$

    Escoge $u \sim U(0, 1)$

    if $u \leq \alpha(x|x_{n-1})$ **then**
        $x_n = x$
    else
        $x_n = x_{n-1}$
    end if
end for

Ejemplo con una distribución normal con varianza conocida

Sea $\theta \sim \text{normal}(\mu, \tau^2)$ y $\{y_1, \ldots, y_n | \theta\} \sim$ i.i.d. $\text{normal}(\theta, \sigma^2)$, la distribución posterior de θ es $\text{normal}(\mu_n, \tau_n^2)$ donde

$$\mu_n = \bar{y} \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau^2}} + \mu \frac{\frac{n}{\tau^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau^2}}$$

$$\tau_n^2 = \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\tau^2}}$$

Supongamos que $\sigma^2 = 1, \tau^2 = 10, \mu = 5, n = 5$ y
$\text{y} = \{9{,}37, 10{,}18, 9{,}16, 11{,}60, 10{,}33\}$.
Con estos datos, $\mu_n = 10{,}03$ y $\tau_n^2 = 0{,}2$, de manera que
$p(\theta|\text{y}) = \text{dnorm}(10{,}03, 0{,}44)$

- Supongamos que no podemos obtener la distribución final, así que usamos el algoritmo Metropolis.
- La proporción de aceptación de un valor propuesto θ^* con respecto al valor actual $\theta^{(s)}$ es

$$r = \frac{p(\theta^*|\mathrm{y})}{p(\theta^{(s)}|\mathrm{y})} = \left(\frac{\prod_{i=1}^{n} \mathrm{dnorm}(y_i, \theta^*, \sigma)}{\prod_{i=1}^{n} \mathrm{dnorm}(y_i, \theta^{(s)}, \sigma)} \right) \times \left(\frac{\mathrm{dnorm}(\theta^*, \mu, \tau)}{\mathrm{dnorm}(\theta^{(s)}, \mu, \tau)} \right)$$

- En algunos casos calcular r directamente puede ser inestable, por lo que se sugiere calcular el logaritmo de r

$$\log r = \sum_{i=1}^{n} \left[\log \mathrm{dnorm}(y_i, \theta^*, \sigma) - \log \mathrm{dnorm}(y_i, \theta^{(s)}, \sigma) \right] + \log \mathrm{dnorm}(\theta^*, \mu, \tau) - \log \mathrm{dnorm}(\theta^{(s)}, \mu, \tau)$$

- Manteniendo las cosas en la escala logarítmica, el valor propuesto se acepta si $\log u < \log r$, donde u es una muestra de una distribución uniforme en $(0, 1)$.

References I

Theodoridis, S. (2020). *Machine Learning. A Bayesian and Optimization Perspective*. Academic Press, second edition edition.

Wilkinson, B. and Allen, M. (1998). *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. Prentice-Hall, Inc., USA.