Práctica 2: Método Montecarlo para cadenas de Markov

Ejercicio 1. Método de la transformación inversa Sea $F(x)$, $x \in \mathbb{R}$, una función de distribución acumulada, es decir $F : \mathbb{R} \to [0,1]$ es monónotona no decreciente, continua a derecha y con límites laterales por izquierda, tal que $\lim_{x\to-\infty} F(x) = 0$ y $\lim_{x\to+\infty} F(x) = 1$. Se define la inversa generalizada de F, $F^{-1} : [0,1] \to \mathbb{R}$,

$$F^{-1}(y) = \min\{x : F(x) \geq y\}, \quad y \in [0,1].$$

1. Si F es invertible entonces la inversa generalizada coincide con la función inversa. Probar que en general vale que
$$F^{-1}\big(F(x)\big) \leq x \quad \text{y} \quad F\big(F^{-1}(y)\big) \geq y.$$
2. Sea U una variable con distribución uniforme en $[0,1]$. Probar que $X = F^{-1}(U)$ está distribuida como X, es decir $P(X \leq x) = F(x)$.

Ejercicio 2. Sorteo por aceptación-rechazo Hay casos en que no podemos encontrar una fórmula explícita para F^{-1}, por ejemplo si la distribución es normal, y necesitamos aplicar otro método. Supongamos que F está asociada a una densidad $f(x)$, es decir, es una distribución absolutamente continua, y sabemos generar otra variable aleatoria absolutamente continua Y con función de distribución G y densidad $g(x)$, tal que $\sup_x\{f(x)/g(x)\} \leq M < \infty$. Consideremos el siguiente algoritmo.

1. Generar $Y \sim G$.
2. Generar una variable $U \sim U[0,1]$ independiente de Y.
3. Si
$$U \leq \frac{f(Y)}{Mg(Y)},$$
tomar $X = Y$ (aceptar); en caso contrario volver a 1. (rechazar).

Probar que la variable X así obtenida tiene función de distribución F.

Ejercicio 3.

1. Generar una variable X exponencial con parámetro $\lambda > 0$.
2. Generar una variable $Z \sim N(\mu, \sigma^2)$.

Ejercicio 4. Algoritmo de subida de colina (hill climb) Un algoritmo de subida de colina es un método para encontrar los máximos de una función f definida sobre los vértices de un grafo: si la posición actual es x, y un vecino y de x verifica $f(y) > f(x)$, el algoritmo indica mover a y. Si f tiene máximos locales, el algoritmo puede quedar atrapado en uno de ellos antes de encontrar el máximo global. Una posible solución a este problema es hacer movimientos aleatorios, que permitan acceder a niveles más bajos de la función.
Sea (V, E) un grafo regular (todos los nodos tienen el mismo grado), de manera que la matriz de transición asociada al paseo aleatorio a vecinos más cercanos sobre el grafo es simétrica, y sea $f : V \to \mathbb{R}$ la función que se desea optimizar. Dado $\lambda \geq 1$, definimos

$$\pi_\lambda(i) = \frac{\lambda^{f(i)}}{Z(\lambda)},$$

donde $Z(\lambda) = \sum_{i\in V} \lambda^{f(i)}$. Observemos que $\pi_\lambda(i)$ es creciente en $f(i)$, es decir favorece a los vértices donde f alcanza mayores valores.

1. Utilizar la dinámica de Metropolis-Hastings para aproximar π_λ, usando como cadena original el paseo aleatorio a vecinos más cercanos.

2. Sea $V^* := \{i \in V : f(i) = \text{máx}_j f(j)\}$. Probar que

$$\lim_{\lambda\to\infty} \pi_\lambda(i) = \frac{\mathbf{1}_{\{i\in V^*\}}}{|V^*|},$$

es decir la distribución invariante de la cadena de Metropolis-Hastings converge a la distribución uniforme sobre los máximos de f.

Ejercicio 5. Modelo hardcore Consideremos un grafo (V, E) y el espacio de estados $\Omega \subseteq \{0,1\}^V$ tal que si $\eta \in \Omega$, $\eta(i)\eta(j) = 0$ cuando $i \sim j$. Esto modela una distribución de partículas en los vértices del grafo tal que hay a lo sumo una partícula en cada vértice, y dos vértices adyacentes no pueden estar simultáneamente ocupados. Cuando $\eta(i) = 1$ diremos que el vértice se encuentra ocupado, y si $\eta(i) = 0$ diremos que está vacante.
Sea π la distribución uniforme sobre Ω.

1. Utilizar la dinámica de Metropolis-Hastings para aproximar π, construida a partir de la siguiente cadena: se elige un vértice al azar, y con probabilidad $1/2$ se coloca una partícula en el vértice (si el vértice ya estaba ocupado, no hay cambios en la configuración).

2. Utilizar la dinámica de Glauber para aproximar π.

Comparar ambas dinámicas.