TIEMPOS DE ALCANCE

Nexus-Probability

CURSO 4 (PROCESOS ESTOCÁSTICOS II)

PARTE 4 / LECCIÓN 2

1. Tiempos de Alcance y el Problema de la Ruina del Jugador

Sea T_a el primer instante en que el proceso de movimiento browniano alcanza a. Cuando $a > 0$ se calculará $P\{T_a \leq t\}$ considerando $P\{X(t) \geq a\}$ y condicionando sobre si $T_a \leq t$ o no. Esto da

$$P\{X(t) \geq a\} = P\{X(t) \geq a \mid T_a \leq t\}\, P\{T_a \leq t\} + P\{X(t) \geq a \mid T_a > t\}\, P\{T_a > t\} \quad (1)$$

Ahora, si $T_a \leq t$, entonces el proceso alcanza a en algún punto en $[0, t]$ y, por simetría, es igual de probable que esté por encima o por debajo de a en el tiempo t. Es decir,

$$P\{X(t) \geq a \mid T_a \leq t\} = \frac{1}{2}.$$

Como el segundo término del lado derecho de la Ecuación (1) es claramente igual a 0 (ya que, por continuidad, el valor del proceso no puede ser mayor que a sin haber alcanzado a previamente), se tiene que

$$\begin{aligned} P\{T_a \leq t\} &= 2P\{X(t) \geq a\} \\ &= 2\frac{1}{\sqrt{2\pi t}} \int_a^\infty e^{-x^2/(2t)}\, dx \\ &= 2\frac{1}{\sqrt{2\pi}} \int_{a/\sqrt{t}}^\infty e^{-y^2/2}\, dy, \quad a > 0 \quad (2) \end{aligned}$$

Para $a < 0$, la distribución de T_a es, por simetría, la misma que la de T_{-a}. Por lo tanto, a partir de la Ecuación (2) se obtiene

$$P\{T_a \leq t\} = 2\frac{1}{\sqrt{2\pi}} \int_{|a|/\sqrt{t}}^\infty e^{-y^2/2}\, dy \quad (3)$$

Otra variable aleatoria de interés es el valor máximo que el proceso alcanza en $[0, t]$. Su distribución se obtiene de la siguiente manera: para $a > 0$

$$P\left(\max_{0\leq s\leq t} X(s) \geq a\right) = P\{T_a \leq t\} \quad \text{(por continuidad)}$$
$$= 2P\{X(t) \geq a\} \quad \text{(de (2))}$$
$$= 2\frac{1}{\sqrt{2\pi}} \int_{a/\sqrt{t}}^{\infty} e^{-y^2/2}\, dy.$$

Consideremos ahora la probabilidad de que el movimiento browniano alcance A antes de $-B$, donde $A > 0$ y $B > 0$. Para calcular esto utilizaremos la interpretación del movimiento browniano como límite de la caminata aleatoria simétrica. Para comenzar, recordemos, a partir de los resultados del problema de la ruina del jugador, que la probabilidad de que la caminata aleatoria simétrica suba hasta A antes de bajar hasta B cuando cada paso es igualmente probable de ser hacia arriba o hacia abajo una distancia Δx es igual a

$$\frac{B\Delta x}{(A+B)\Delta x} = \frac{B}{A+B}.$$

Por lo tanto, al dejar $\Delta x \to 0$, se tiene que

$$P\{\text{subir a } A \text{ antes de bajar a } B\} = \frac{B}{A+B}.$$

2. Ejercicios

Ejercicio 1 *Simula una trayectoria de movimiento browniano estándar en $[0, 2]$ y calcula el **primer tiempo** T_a en que la trayectoria alcanza el nivel $a = 1.5$.*

```
import numpy as np
import matplotlib.pyplot as plt

# Par metros comunes
T = 2.0
N = 1000
dt = T / N
t = np.linspace(0, T, N + 1)
np.random.seed(42)

# ==============================
# Ejercicio 1: Primer tiempo de alcance Ta para a = 1.5
# ==============================

a = 1.5
dW = np.sqrt(dt) * np.random.randn(N)
```

```python
W = np.concatenate(([0], np.cumsum(dW)))

# Calcular primer tiempo en que W(t) >= a
hitting_time_index = np.where(W >= a)[0]
if hitting_time_index.size > 0:
    Ta = t[hitting_time_index[0]]
else:
    Ta = np.nan  # No alcanz  a

plt.figure(figsize=(10, 5))
plt.plot(t, W, label='Trayectoria Browniana')
plt.axhline(a, color='r', linestyle='--', label=f'Nivel a = {a}')
if not np.isnan(Ta):
    plt.axvline(Ta, color='g', linestyle='--', label=f'Tiempo de alcance Ta = {Ta:.3f}')
plt.title('Ejercicio 1: Tiempo de Alcance de a = 1.5')
plt.xlabel('Tiempo')
plt.ylabel('X(t)')
plt.grid()
plt.legend()
plt.show()

print(f"Primer tiempo de alcance Ta: {Ta:.4f}" if not np.isnan(Ta) else "No alcanz  el nivel a.")
```

Ejercicio 2 *Simula 1000 trayectorias de movimiento browniano estándar y estima de manera empírica la probabilidad de que* $T_a \leq 1$ *para* $a = 1$.

```python
a = 1
M = 1000
hitting_counts = 0

for _ in range(M):
    dW = np.sqrt(dt) * np.random.randn(N)
    W = np.concatenate(([0], np.cumsum(dW)))
    hitting_time_index = np.where(W >= a)[0]
    if hitting_time_index.size > 0 and t[hitting_time_index[0]] <= 1:
        hitting_counts += 1

prob_empirical = hitting_counts / M
print(f"Probabilidad emp rica de que Ta     1 para a = 1: {prob_empirical:.4f}")
```

Ejercicio 3 *Para 1000 trayectorias, calcula la distribución empírica del* ***valor máximo*** *alcanzado en el intervalo* $[0,1]$. *Compara con la fórmula teórica del máximo.*

```
T_max = 1.0
N_max = int(N * T_max / T)
max_values = np.zeros(M)

for i in range(M):
    dW = np.sqrt(dt) * np.random.randn(N_max)
    W = np.concatenate(([0], np.cumsum(dW)))
    max_values[i] = np.max(W)

plt.figure(figsize=(10, 5))
plt.hist(max_values, bins=30, density=True, alpha=0.7, color='orange', edgecolor='black')
plt.title('Ejercicio 3: Distribuci n del M ximo en [0,1]')
plt.xlabel('M ximo de X(t)')
plt.ylabel('Densidad')
plt.grid()
plt.show()
```