”P1” Movimiento Browniano

NESTOR RODRIGUEZ

Febrero 2022

1. Introducción

El primer reto es estudiar de forma sistemática y automatizada el tiempo de ejecución de una caminata (en milisegundos) en términos del largo de la caminata (en pasos) y la dimensión. Para medir el tiempo de una réplica, ejecútala múltiples veces y normaliza con la cantidad de repeticiones para obtener un promedio del tiempo de una réplica individual. El segundo reto es realizar una comparación entre una implementación paralela y otra versión que no aproveche paralelismo en términos del tiempo de ejecución, aplicando alguna prueba estadística adecuada para determinar si la diferencia es significativa.

2. Objetivo

El objetivo de esta práctica es examinar de manera sistemática los efectos de la dimensión en la distancia Euclideana máxima del origen del movimiento Browniano para dimensiones 1, 2, 3, 4 y 5, variando el número de pasos de la caminata (100, 1000 o 10000 pasos), con 30 repeticiones del experimento para cada combinación (1 con 100, 1 con 1000, 1 con 10000, 2 con 100, 2 con 1000, etc.) y graficar todos los resultados en una sola figura con diagramas de caja-bigote.

3. Desarrollo

La definicion de Movimiento Browniano: hace referencia al movimiento de partículas microscopicas que experimentan un movimiento aleatorio debido a fluctuaciones termicas, fenómeno observado por primera vez en 1827 por R. Brown y descrito formalmente en 1905 por A. Einstein. Así que con esto se realizo para generar el código con objetivo de la práctica, esto se encuentran en: mi repositorio en GitHub. Inicie tomando como base el Tomando como base la página código proporcionado por la Dra. Elisa Schaeffer [1], se hace una modificacion para iterar entre la cantidad de pasos que se realizan en cada dimension.

3.1. Ejecución en Python

A continuacion se realiza el siguiente ejercicio en código Python para la realización de la ejecución en los siguientes comandos.

```
from random import random, randint
from math import fabs, sqrt
import matplotlib.pyplot as plt
import numpy as np
from time import time

runs = 30 #replicas
caminatas = [100, 1000, 10000] #pasos
results = [] #almacena las dimensiones
```

```
for i in range(3): #itera la cantidad de pasos
    dur = caminatas[i]
    for dim in range(1, 6): #de una a cinco dimensiones
        mayores = []
        for rep in range(runs):#corre el experimento 30 veces en cada dimension
            before = time()*1000
            pos = [0] * dim
            mayor = 0
            for paso in range(dur):
                eje = randint(0, dim - 1)
                if pos[eje] > -100 and pos[eje] < 100:
                    if random() < 0.5:
                        pos[eje] += 1
                    else:
                        pos[eje] -= 1
                else:
                    if pos[eje] == -100:
                        pos[eje] += 1
                    if pos[eje] == 100:
                        pos[eje] -= 1
                mayor = max(mayor, sqrt(sum([p**2 for p in pos])))
            mayores.append(mayor)
            after = time()*1000
        results.append(mayores)
tiempo = after - before
print(tiempo)

#separar los resultados en tres grupos de caminatas
walks_1 = results[0:4]
walks_2 = results[4:8]
walks_3 = results[8:12]

#empezar a graficar

ticks = ['1', '2', '3', '4', '5']

#funcion definir los colores de cajas y bigotes
def set_box_color(bp, color):
    plt.setp(bp['boxes'], color='green')
    plt.setp(bp['whiskers'], color='gray')
    plt.setp(bp['caps'], color='red')
    plt.setp(bp['medians'], color='purple')

def box_plot(data, edge_color, fill_color):
    bp = ax.boxplot(data, patch_artist=True)

    for element in ['boxes', 'whiskers', 'fliers', 'means', 'medians', 'caps']:
        plt.setp(bp[element], color=edge_color)

    for patch in bp['boxes']:
        patch.set(facecolor=fill_color)

    for patch, color in zip(box['boxes'], colors):
```

```
        patch.set_facecolor(color)


plt.figure()

bpl = plt.boxplot(walks_1, positions=np.array(range(len(walks_1)))*6.0-1.0, sym='-1', widths=
bpc = plt.boxplot(walks_2, positions=np.array(range(len(walks_2)))*6.0, sym='-1', widths=1.2)
bpr = plt.boxplot(walks_3, positions=np.array(range(len(walks_3)))*6.0+1.0, sym='-1', widths=
set_box_color(bpl, '#67001f')
set_box_color(bpc, '#1a1a1a')
set_box_color(bpr, '#d6604d')


plt.xticks(range(0, len(ticks)*5, 5), ticks)
plt.ylim(0, len(ticks)*40)
plt.xlim(-3, len(ticks)*5)
plt.title('Distancia Manhattan')
plt.tight_layout()
plt.savefig('DistanciaMan.png')
plt.show()
```

4. Resultados

En esta seccion se muestra la grafica Manhattan en corelacion de dimensiones 1, 2, 3, 4 y 5, variando el número de pasos de la caminata (100, 1000 o 10000 pasos), con 30 repeticiones del experimento para cada combinación (1 con 100, 1 con 1000, 1 con 10000, 2 con 100, 2 con 1000, etc.).

5. Mediciones y Gráfica

Se observa la siguiente tabla en la distribucion de la medicion en los vectores en Python. A partir de las secciones anteriores encontramos que las principales características de este fenómeno son: Las partículas pequeñas tienen mayor velocidad. Las partículas se mueven más rápido en fluidos con poca viscosidad. La energía promedio de las partículas es proporcional a la temperatura. Aumenta la cantidad de pasos, lo cual se debe a que tiene las posibilidades de determinar en ciertos puntos diferentes del espacio.

Cuadro 1: Mediciones para la distancia Euclideana máxima del origen del movimiento Browniano para dimensiones

Medición	Pasos
1, 2, 3,4 y 5	100
1, 2, 3,4 y 5	1000
1, 2, 3,4 y 5	10000

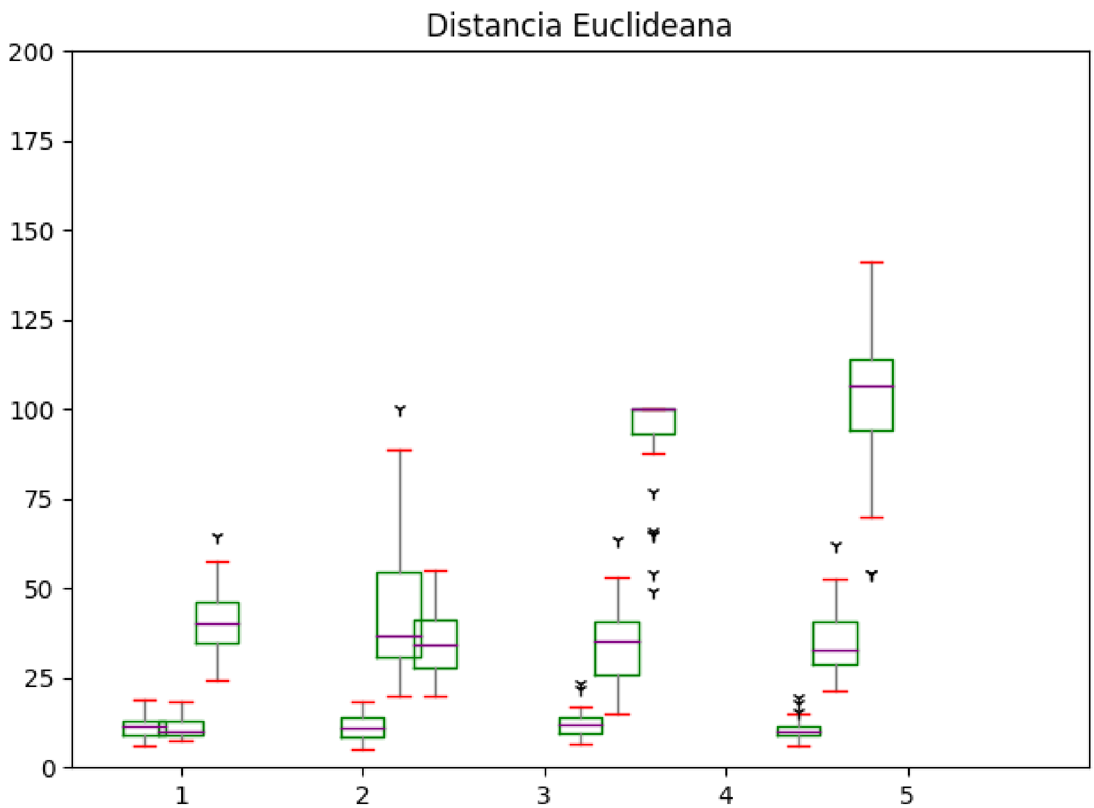

Figura 1: Diagrama caja-bigote en los grupos de 100, 1,000 y 10,000 pasos en dimensiones de 1 a 5 con 30 repeticiones.

6. Conclusion

Se realizo la tarea 1 con éxito para obtener la caja-bigote para las repeticiones del experimento. Al observar los diagramas de la figura 1 se caracteriza en que el rango del máximo que es amplio al aumentar la cantidad de pasos.

Referencias

[1] Dra. Elisa Schaeffer. Brownianmotion. *Repositorio, GitHub*, 2019.