AP1

Murmani Akhaladze

TarnNished

The goal of this project was to explore vector and matrix norms, compute distances between random data samples, and visualize the geometric shapes of unit balls for different norms.

Two vector norms were used:

$$\|x\|_1 = \sum_i |x_i|, \quad \|x\|_2 = \sqrt{\sum_i x_i^2}$$

Their corresponding induced matrix norms were chosen as:

$$\|A\|_1 = \max_j \sum_i |a_{ij}|, \quad \|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

where $\lambda_{\max}$ is the largest eigenvalue of $A^T A$ (the spectral norm).

Two random 4-dimensional vectors x_1 and x_2 were generated using `NumPy`. They were reshaped into 2×2 matrices A_1 and A_2. The following distances were computed:

$$d_v^{(1)} = \|x_1 - x_2\|_1, \quad d_v^{(2)} = \|x_1 - x_2\|_2$$

$$d_m^{(1)} = \|A_1 - A_2\|_1, \quad d_m^{(2)} = \|A_1 - A_2\|_2$$

The script printed all distances and plotted visualizations for L1 and L2 unit balls.

For the L2 norm, the unit ball appears as a **circle**, since it includes all points with Euclidean distance ≤ 1 from the origin. For the L1 norm, the unit ball forms a **diamond shape**, as it satisfies $|x_1| + |x_2| \leq 1$. These plots illustrate how different norms define distance differently in geometric space.

This project demonstrated how norm definitions affect both numerical distances and geometric interpretations. The L2 norm emphasizes Euclidean distance, while the L1 norm highlights coordinate-wise differences. Matrix norms extended these ideas to higher dimensions, showing how numerical linear algebra measures magnitude and similarity across both vectors and matrices.

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

def vector_norm(x, p=2):
    return np.linalg.norm(x, ord=p)

def matrix_norm(A, p=2):
    return np.linalg.norm(A, ord=p)

np.random.seed(42)

x1 = np.random.rand(4)
x2 = np.random.rand(4)

A1 = x1.reshape(2, 2)
A2 = x2.reshape(2, 2)

print("x1 =", x1)
print("x2 =", x2)
print("\nA1 =\n", A1)
print("A2 =\n", A2)

d_L1 = vector_norm(x1 - x2, p=1)
d_L2 = vector_norm(x1 - x2, p=2)
d_matrix_L1 = matrix_norm(A1 - A2, p=1)
d_matrix_L2 = matrix_norm(A1 - A2, p=2)

print("\n--- Distances ---")
print(f"Vector distance (L1): {d_L1:.4f}")
print(f"Vector distance (L2): {d_L2:.4f}")
print(f"Matrix distance (L1 induced): {d_matrix_L1:.4f}")
print(f"Matrix distance (L2 induced): {d_matrix_L2:.4f}")

theta = np.linspace(0, 2 * np.pi, 400)
circle_x = np.cos(theta)
circle_y = np.sin(theta)

l1_x = np.linspace(-1, 1, 400)
l1_y1 = 1 - np.abs(l1_x)
l1_y2 = -1 + np.abs(l1_x)

plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
plt.plot(circle_x, circle_y)
plt.title("L2 (Euclidean) Unit Ball")
plt.axis("equal")
plt.grid(True)

plt.subplot(1, 2, 2)
plt.plot(l1_x, l1_y1, 'r')
plt.plot(l1_x, l1_y2, 'r')
plt.title("L1 (Manhattan) Unit Ball")
plt.axis("equal")
plt.grid(True)

plt.suptitle("Sections of Unit Balls for L1 and L2 Norms", fontsize=13)
plt.show()
```

```
C:\Users\makha\PycharmProjects\numerical\.venv\Scripts\python.exe C:\Users\makha\PycharmProjects\
x1 = [0.37454012 0.95071431 0.73199394 0.59865848]
x2 = [0.15601864 0.15599452 0.05808361 0.86617615]

A1 =
 [[0.37454012 0.95071431]
 [0.73199394 0.59865848]]
A2 =
 [[0.15601864 0.15599452]
 [0.05808361 0.86617615]]

--- Distances ---
Vector distance (L1): 1.9547
Vector distance (L2): 1.0977
Matrix distance (L1 induced): 1.0622
Matrix distance (L2 induced): 0.8387

Process finished with exit code 0
```