淡江大學統計學系

數據科學組碩一

第二次作業

613890176 魏祺紘

繳交時間:10/21

使用程式 ：Python

Q1 : Write the procedure and a program for calculating the Jordan Decomposition of the matrix A.

A = $\begin{bmatrix} 25 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 9 \end{bmatrix}$

$\begin{vmatrix} 25 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 9 \end{vmatrix}$

= (25-λ) $\begin{vmatrix} 4-\lambda & 1 \\ 1 & 9-\lambda \end{vmatrix}$-(-2) $\begin{vmatrix} -2 & 1 \\ 1 & 9-\lambda \end{vmatrix}$+4 $\begin{vmatrix} -2 & 4-\lambda \\ 4 & 1 \end{vmatrix}$

$=\lambda^3-38\lambda^2+340\lambda+759$

由 Python 程式得知，特徵值λ_1、λ_2、λ_3

λ_1 = 26.0739 ,λ_2 = 8.495 ,λ_3 = 3.425

V_1 = $\begin{bmatrix} 0.9717 \\ -0.0779 \\ 0.2230 \end{bmatrix}$,V_2 = $\begin{bmatrix} -0.1914 \\ 0.2935 \\ 0.9366 \end{bmatrix}$,V_3 = $\begin{bmatrix} 0.1384 \\ 0.9528 \\ -0.2703 \end{bmatrix}$

Jordan Decomposition

Γ = $\begin{bmatrix} 0.9717 & -0.1914 & 0.1384 \\ -0.0779 & 0.2935 & 0.9528 \\ 0.2230 & 0.9366 & -0.2703 \end{bmatrix}$,Λ = $\begin{bmatrix} 26.0785 & 0 & 0 \\ 0 & 8.4958 & 0 \\ 0 & 0 & 3.4258 \end{bmatrix}$

Γ^T = $\begin{bmatrix} 0.9717 & -0.0779 & 0.2230 \\ -0.1914 & 0.2935 & 0.9366 \\ 0.1384 & 0.9528 & -0.2703 \end{bmatrix}$

$\Gamma\Lambda\Gamma^{T}$

$$= \begin{bmatrix} 0.9717 & -0.1914 & 0.1384 \\ -0.0779 & 0.2935 & 0.9528 \\ 0.2230 & 0.9366 & -0.2703 \end{bmatrix} \begin{bmatrix} 26.0785 & 0 & 0 \\ 0 & 8.4958 & 0 \\ 0 & 0 & 3.4258 \end{bmatrix} \begin{bmatrix} 0.9717 & -0.0779 & 0.2230 \\ -0.1914 & 0.2935 & 0.9366 \\ 0.1384 & 0.9528 & -0.2703 \end{bmatrix}$$

$$= \begin{bmatrix} 25.340 & -1.626 & 0.474 \\ -2.032 & 2.493 & 3.263 \\ 5.816 & 7.957 & -0.925 \end{bmatrix} \begin{bmatrix} 0.9717 & -0.0779 & 0.2230 \\ -0.1914 & 0.2935 & 0.9366 \\ 0.1384 & 0.9528 & -0.2703 \end{bmatrix}$$

$$= \begin{bmatrix} 25 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 9 \end{bmatrix} = A$$

```
Q1: 計算Jordan Decompasition
特徵值:
λ1 = 26.0785
λ2 = 8.4958
λ3 = 3.4258

特徵向量:
v1 = [-0.97169436  0.07792066 -0.22302119]
v2 = [ 0.19143136 -0.29348804 -0.9365996 ]
v3 = [ 0.13843451  0.95278179 -0.27026422]
驗證 A = ΓΛΓ⁻¹:
是否相等: True
```

Q2 : Use the answer of Q1 to check whether matrix A is positive definite. Explain your reason.

By Q1 $\lambda_1 = 26.0739$,$\lambda_2 = 8.495$,$\lambda_3 = 3.425$

∵ 所有的特徵值皆大於 0

∴ 可判定 A 為 Positive definite

```
Q2: 檢查矩陣是否正定
矩陣 A 是否正定: True
特徵值: [ 3.42575179  8.49579584 26.07845236]
```

Q3 : Use the answer of Q1 to calculate $A^{-1/2}$.

$A^{-1/2} = \Gamma \Lambda^{-1/2} \Gamma^{T}$

$$= \begin{bmatrix} 0.9717 & -0.1914 & 0.1384 \\ -0.0779 & 0.2935 & 0.9528 \\ 0.2230 & 0.9366 & -0.2703 \end{bmatrix} \begin{bmatrix} 0.5402 & 0 & 0 \\ 0 & 0.3430 & 0 \\ 0 & 0 & 0.1958 \end{bmatrix} \begin{bmatrix} 0.9717 & -0.0779 & 0.2230 \\ -0.1914 & 0.2935 & 0.9366 \\ 0.1384 & 0.9528 & -0.2703 \end{bmatrix}$$

$$= \begin{bmatrix} 0.2078 & 0.0371 & -0.0392 \\ 0.0371 & 0.5212 & -0.0482 \\ -0.0392 & -0.0482 & 0.3501 \end{bmatrix}$$

```
Q3: 計算矩陣的 A^(-1/2)
A^(-1/2):
[[ 0.20781878  0.0371604  -0.03929081]
 [ 0.0371604   0.52120645 -0.04822095]
 [-0.03929081 -0.04822095  0.35016166]]

Λ^(-1/2):
[[0.54028393 0.         0.        ]
 [0.         0.34308203 0.        ]
 [0.         0.         0.19582092]]
```

Q4 : Is matrix A a orthogonal matrix? Explain your answer.

Orthogonal matrix 的判別方法 ：

1. $AA^T = I$
2. $A^T = A^{-1}$

AA^T

$$= \begin{bmatrix} 25 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 9 \end{bmatrix} \begin{bmatrix} 25 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 9 \end{bmatrix} \neq \begin{bmatrix} 645 & -54 & 134 \\ -54 & 21 & 5 \\ 134 & 5 & 98 \end{bmatrix}$$

$A^T = A^{-1}$

$$= \begin{bmatrix} 25 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 9 \end{bmatrix} \neq \begin{bmatrix} 0.0461 & 0.0289 & -0.0237 \\ 0.0289 & 0.2753 & -0.0434 \\ -0.0237 & -0.0434 & 0.1264 \end{bmatrix}$$

∴ 可判定 A 不為 Orthogonal matrix

```
Q4: 檢查矩陣是否正交
正交性檢查結果:
1. AA^T = I: False
2. A^T = A^(-1): False
3. 列向量為單位向量且互相正交: False

AA^T:
[[645 -54 134]
 [-54  21   5]
 [134   5  98]]

A^(-1):
[[ 0.04611331  0.02898551 -0.02371542]
 [ 0.02898551  0.27536232 -0.04347826]
 [-0.02371542 -0.04347826  0.12648221]]
```

附錄

```
import numpy as np
from numpy import linalg as LA


def calculate_eigenvalues_eigenvectors(A):
    # 使用 eigh 函數計算對稱矩陣的特徵值和特徵向量
    eigenvalues, eigenvectors = LA.eigh(A)
    # 將特徵值按降序排列，並相應地調整特徵向量
    idx = eigenvalues.argsort()[::-1]
    return eigenvalues[idx], eigenvectors[:, idx]


def is_positive_definite(A):
    # 計算對稱矩陣的特徵值
    eigenvalues = LA.eigvalsh(A)
    # 檢查所有特徵值是否為正
    return np.all(eigenvalues > 0), eigenvalues


def calculate_inverse_sqrt(A):
    # 使用特徵值分解計算矩陣的 A^(-1/2)
    eigenvalues, eigenvectors = LA.eigh(A)
    # 計算 Λ^(-1/2)
```

```
    Lambda_inv_sqrt = np.diag(1.0 / np.sqrt(eigenvalues))

    # 使用公式 A^(-1/2) = Q * Λ^(-1/2) * Q^T 計算 A^(-1/2)

    return eigenvectors @ Lambda_inv_sqrt @ eigenvectors.T, eigenvectors, np.diag(eigenvalues), Lambda_inv_sqrt


def is_orthogonal(A, tolerance=1e-10):
    n = A.shape[0]
    I = np.eye(n)

    # 使用三種方法檢查矩陣是否正交

    return (

        np.allclose(A @ A.T, I, atol=tolerance),   # 檢查 AA^T = I

        np.allclose(A.T, LA.inv(A), atol=tolerance),   # 檢查 A^T = A^(-1)

        np.allclose(A @ A.T, I, atol=tolerance)   # 等同於檢查列向量是否為單位向量且互相正交

    )


def main():

    # 定義待分析的矩陣

    A = np.array([[25, -2, 4],

                  [-2, 4, 1],

                  [4, 1, 9]])


    print("原始矩陣 A:")
```

```
print(A)

print("\n")


# Q1: 計算特徵值和特徵向量

print("Q1: 計算特徵值和特徵向量")

V1, V2 = calculate_eigenvalues_eigenvectors(A)


print("特徵值:")

for i, eigenvalue in enumerate(V1, 1):

    print(f"λ{i} = {eigenvalue:.4f}")


print("\n 特徵向量:")

for i in range(3):

    print(f"v{i+1} = {V2[:, i]}")


print("驗證 A = ΓΛΓ⁻¹:")

Lambda = np.diag(V1)

A_reconstructed = V2 @ Lambda @ V2.T

print(f"是否相等: {np.allclose(A, A_reconstructed)}")

print("\n")
```

```
# Q2: 檢查矩陣是否正定

print("Q2: 檢查矩陣是否正定")

is_positive, eigenvalues = is_positive_definite(A)

print(f"矩陣 A 是否正定: {is_positive}")

print(f"特徵值: {eigenvalues}")

print("\n")



# Q3: 計算矩陣的 A^(-1/2)

print("Q3: 計算矩陣的 A^(-1/2)")

A_inv_sqrt, Gamma, Lambda, Lambda_inv_sqrt = calculate_inverse_sqrt(A)

print("A^(-1/2):")

print(A_inv_sqrt)

print("\nΛ^(-1/2):")

print(Lambda_inv_sqrt)

print("\n")



# Q4: 檢查矩陣是否正交

# 計算 A 的轉置

A_transpose = A.T
```

```
print("\n 矩陣 A 的轉置 A^T:")

print(A_transpose)

print("Q4: 檢查矩陣是否正交")

is_orth_1, is_orth_2, is_orth_3 = is_orthogonal(A)

print("正交性檢查結果:")

print(f"1. AA^T = I: {is_orth_1}")

print(f"2. A^T = A^(-1): {is_orth_2}")

print(f"3. 列向量為單位向量且互相正交: {is_orth_3}")



# 計算 AA^T

AA_T = A @ A_transpose


print("\nAA^T:")

print(AA_T)


# 計算 A^(-1)

A_inverse = LA.inv(A)


print("\nA^(-1):")

print(A_inverse)


if all([is_orth_1, is_orth_2, is_orth_3]):
```

```python
        print("\n 結論: 矩陣 A 是正交的。")
    else:
        print("\n 結論: 矩陣 A 不是正交的。")


if __name__ == "__main__":
    main()
```