

Homework 6

Exercise 2

求解线性方程组和矩阵的基本运算

题目：

生成一个由实数组成的秩为 4 的 4 行 4 列的矩阵 **A** 和一个由实数组成的包含 4 个元素的列向量 **b**。求解线性方程组 $\mathbf{Ax} = \mathbf{b}$ 。计算矩阵 **A** 的转置、行列式、秩、逆矩阵、特征值和特征向量。

源代码：

```
print("This is a python code for homework6-1: Calculate of Matrix")
import numpy as np
a = np.array([(1, 9, 1, 9), (8, 1, 0, 1), (1, 4, 5, 1), (4, 0, 0, 1)])
b = np.array([1, 9, 1, 9])
aT = np.transpose(a)
aI = np.linalg.pinv(a)
c = np.linalg.eig(a)
x = aI.dot(b)
print("The 4*4 matrix A is:\n", a)
print("The 4 array b is:\n", b)
print("The answer X of equation AX = b is:\n", x)
print("The transpose of A is:\n", aT)
print("The det(A) is:\n", np.linalg.det(a))
print("The rank(A) is:\n", np.linalg.matrix_rank(a))
print("The Inverse of A is:\n", aI)
print("The Eigenvalues of A are:\n", c[0])
print("The Eigenvectors of A are:\n", c[1])
```

运行结果：

```
/usr/local/bin/python3.9 "/Users/wangyijie/Library/Mobile
Documents/com~apple~CloudDocs/Study_in_USTC/杂事/python科学计算/hw6/hw6_1.py"

This is a python code for homework6-1: Calculate of Matrix
The 4*4 matrix A is:
[[1 9 1 9]
 [8 1 0 1]
 [1 4 5 1]
 [4 0 0 1]]
The 4 array b is:
[1 9 1 9]
The answer X of equation AX = b is:
[ 1.16666667 -4.66666667  2.83333333  4.33333333]
The transpose of A is:
```

```

[[1 8 1 4]
 [9 1 4 0]
 [1 0 5 0]
 [9 1 1 1]]
The det(A) is:
-336.0
The rank(A) is:
4
The Inverse of A is:
[[-0.01488095  0.12202381  0.00297619  0.00892857]
 [ 0.05952381  0.51190476 -0.01190476 -1.03571429]
 [-0.05654762 -0.33630952  0.21130952  0.63392857]
 [ 0.05952381 -0.48809524 -0.01190476  0.96428571]]
The Eigenvalues of A are:
[-9.1355856  11.8783503  4.581386  0.67584931]
The Eigenvectors of A are:
[[-0.75646452  0.6828826  0.03386454 -0.04834603]
 [ 0.56762167  0.52527805  0.08620661 -0.647288  ]
 [-0.12822675  0.44125299 -0.99498294  0.47197961]
 [ 0.29853806  0.25109785  0.03782283  0.59658709]]

Process finished with exit code 0

```

程序运行正常，结果正确。

Exercise 2

计算最小二乘解和矩阵分解

题目：

生成一个由实数组成的秩为 4 的 6 行 4 列的实矩阵 \mathbf{B} 和一个由实数组成的包含 6 个元素的列向量矩阵 \mathbf{b} 。计算最小二乘解 $(\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{b}$ 。计算矩阵 \mathbf{B} 的奇异值分解并验证6.1。计算矩阵 \mathbf{B} 的LU分解。

```

print("This is a python code for homework6-2: Least Square Method and Factorization")
import numpy as np
from scipy.linalg import lu
from scipy import linalg
a = np.array([(1, 9, 1, 9), (8, 1, 0, 1), (1, 4, 5, 1), (4, 1, 1, 4), (5, 1, 4, 1), (9,
1, 9, 8)])
b = np.array([1, 9, 1, 9, 8, 1])
aT = np.transpose(a)
C = aT.dot(a)
CI = np.linalg.pinv(C)
CII = CI.dot(aT)
CIII = CII.dot(b)
U, s, V = linalg.svd(a)
p, l, u = lu(a)
print("***** Matrix Setup *****")

```

```

print("The 4*4 matrix B is:\n", a)
print("***** Least Square Method *****")
print("The least square answer is:\n", CIII)
print("***** SVD Factorization *****")
print("The U of SVD is:\n", U)
print("The s of SVD is:\n", s)
print("The V of SVD is:\n", V)
s_matrix = np.zeros((6, 4))
s_matrix[:4, :4] = np.diag(s)
VT = np.transpose(V)
UT = np.transpose(U)
print("The verify Avi of SVD is:\nAv1:", a.dot(V[0]))
print("sU1:", UT[0]*s[0])
print("Av2:", a.dot(V[1]))
print("sU2:", UT[1]*s[1])
print("Av3:", a.dot(V[2]))
print("sU3:", UT[2]*s[2])
print("Av4:", a.dot(V[3]))
print("sU4:", UT[3]*s[3])
print("The verify ATui of SVD is:\nATu1:", aT.dot(UT[0]))
print("sV1:", V[0]*s[0])
print("ATu2:", aT.dot(UT[1]))
print("sV2:", V[1]*s[1])
print("ATu3:", aT.dot(UT[2]))
print("sV3:", V[2]*s[2])
print("ATu4:", aT.dot(UT[3]))
print("sV4:", V[3]*s[3])
print("ATu5:", aT.dot(UT[4]))
print("ATu6:", aT.dot(UT[5]))
print("***** LU Factorization *****")
print("The P of LU is:\n", p)
print("The L of LU is:\n", l)
print("The U of LU is:\n", u)

```

运行结果:

```

/usr/local/bin/python3.9 "/Users/wangyijie/Library/Mobile
Documents/com~apple~CloudDocs/Study_in_USTC/杂事/python科学计算/hw6/hw6_2.py"

This is a python code for homework6-2: Least Square Method and Factorization
***** Matrix Setup *****

The 4*4 matrix B is:
[[1 9 1 9]
 [8 1 0 1]
 [1 4 5 1]
 [4 1 1 4]
 [5 1 4 1]
 [9 1 9 8]]

```

***** Least Square Method *****

The least square answer is:

[1.4027281 0.68490738 -0.59525357 -0.58874886]

***** SVD Factorization *****

The U of SVD is:

[[-0.45425403 0.83441203 0.21565222 0.01904128 0.10824814 0.19703739]
[-0.27599422 -0.33391357 0.69842011 0.44235687 -0.32620475 0.14983032]
[-0.23777597 0.11913344 -0.53289167 0.63838742 -0.35132535 -0.33812463]
[-0.26922678 -0.02412137 0.28686882 -0.25265592 0.12208644 -0.87515786]
[-0.28192403 -0.2638072 -0.09405295 0.36630561 0.83807728 0.07433245]
[-0.71342752 -0.32846528 -0.30098215 -0.44542622 -0.20289077 0.23015709]]

The s of SVD is:

[20.20811106 10.57941213 6.2244532 4.23845122]

The V of SVD is:

[[-0.5842881 -0.32560999 -0.46817362 -0.57740715]
[-0.57559646 0.66505913 -0.2462763 0.40710326]
[0.52028547 0.06419153 -0.84296376 0.12080797]
[0.23790579 0.66899473 0.09784446 -0.69733303]]

The verify Avi of SVD is:

Av1: [-9.17961594 -5.57732192 -4.80500329 -5.4405646 -5.69715211
-14.41702265]

sU1: [-9.17961594 -5.57732192 -4.80500329 -5.4405646 -5.69715211
-14.41702265]

Av2: [8.82758871 -3.53260926 1.26036181 -0.25518996 -2.79092509 -3.4749696]

sU2: [8.82758871 -3.53260926 1.26036181 -0.25518996 -2.79092509 -3.4749696]

Av3: [1.34231717 4.34728329 -3.31695927 1.78560154 -0.58542819 -1.87344932]

sU3: [1.34231717 4.34728329 -3.31695927 1.78560154 -0.58542819 -1.87344932]

Av4: [0.08070552 1.87490801 2.70577395 -1.07086978 1.55256847 -1.8879173]

sU4: [0.08070552 1.87490801 2.70577395 -1.07086978 1.55256847 -1.8879173]

The verify ATui of SVD is:

ATu1: [-11.80735877 -6.57996274 -9.46090452 -11.66830782]

sV1: [-11.80735877 -6.57996274 -9.46090452 -11.66830782]

ATu2: [-6.08947212 7.03593459 -2.60545847 4.30691315]

sV2: [-6.08947212 7.03593459 -2.60545847 4.30691315]

ATu3: [3.23849258 0.39955715 -5.2469885 0.75196355]

sV3: [3.23849258 0.39955715 -5.2469885 0.75196355]

ATu4: [1.00835208 2.83550151 0.41470896 -2.95561203]

sV4: [1.00835208 2.83550151 0.41470896 -2.95561203]

ATu5: [0.00000000e+00 -1.11022302e-16 4.44089210e-16 4.44089210e-16]

ATu6: [-1.33226763e-15 -1.66533454e-15 -4.44089210e-16 1.55431223e-15]

***** LU Factorization *****

The P of LU is:

[[0. 1. 0. 0. 0. 0.]
[0. 0. 1. 0. 0. 0.]
[0. 0. 0. 1. 0. 0.]
[0. 0. 0. 0. 0. 1.]
[0. 0. 0. 0. 1. 0.]
[1. 0. 0. 0. 0. 0.]]

The L of LU is:

```
[[ 1.          0.          0.          0.          ]
 [ 0.11111111  1.          0.          0.          ]
 [ 0.88888889  0.0125      1.          0.          ]
 [ 0.11111111  0.4375     -0.5         1.          ]
 [ 0.55555556  0.05        0.125       0.46967526 ]
 [ 0.44444444  0.0625     0.375       -0.34646609]]
```

The U of LU is:

```
[[ 9.          1.          9.          8.          ]
 [ 0.          8.88888889  0.          8.11111111 ]
 [ 0.          0.          -8.         -6.2125      ]
 [ 0.          0.          0.          -6.54375     ]]
```

Process finished with exit code 0

程序运行正常，结果正确。