

# 第六次作业讲评

2022-03-21

郭明非

## Q1.1 (1分) 创建矩阵并计算。

```
import numpy as np
np.random.seed(1)
# TODO
# 不用print
A = np.random.randn(20, 40)
B = np.random.randn(40, 40)
C = np.random.randn(40, 1)
D = np.random.randn(40, 1)

# TODO
# 不用print
R = A + A  A: {ndarray}(20, 40)  A: {ndarray}(20, 40)
R = A.T @ A  A: {ndarray}(20, 40)  A: {ndarray}(20, 40)
R = A @ B  A: {ndarray}(20, 40)  B: {ndarray}(40, 40)
R = B - np.eye(40)  B: {ndarray}(40, 40)
```

Q1.2 (1分) 计算方程组的解。

$$\begin{cases} 2x_1 + x_2 + x_3 = 5 \\ 3x_1 + 2x_2 + 2x_3 = 8 \\ x_2 + 2x_3 = 2 \end{cases}$$

```
# 黄楠 1900012126
A = np.array(
    [
        [2, 1, 1],
        [3, 2, 2],
        [0, 1, 2],
    ]
)
b = np.array([5, 8, 2])
x = np.linalg.solve(A, b)
x.astype(int)
```

```
# 章晨昊 1900013096
A1=np.array([[2,1,1],[3,2,2],[1,2,0]])
B=np.array([5,8,2])
C=np.linalg.inv(A1).dot(B)
C=C.reshape(3,1)
print(C)
```

调用solve(A, B)，求解线性矩阵方程，计算AX=B。

直接计算X=A\*B，  
A\*AX=X=A\*B。

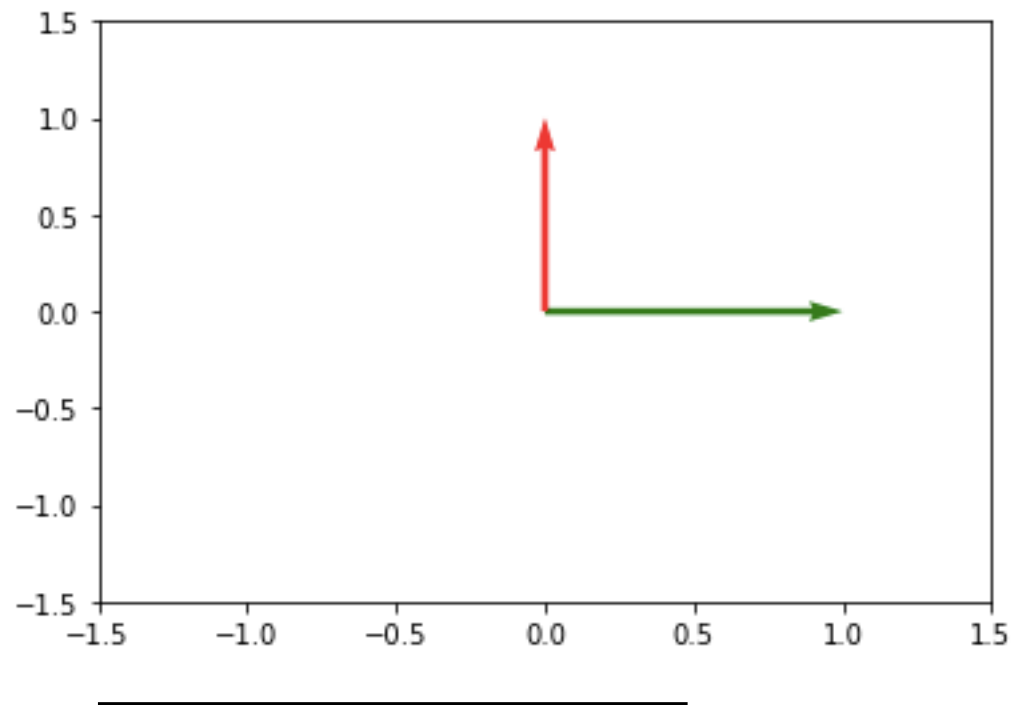
Q1.3 (2分) 求解矩阵的特征值以及对应的特征向量，输出最大最小特征值以及对应的特征向量并画图。

```
# 韦锡宇 2000013085
R = np.array([[2,0],[0,3]])
# TODO
A,B = np.linalg.eig(R)
idx_max= np.argmax(A)
idx_min=np.argmin(A)
print(B[:,[idx_max]])
print(B[:,[idx_min]])
```

特征向量矩阵按列排列。eig返回的特征值不一定有序，需要使用argmax, argmin来获取最大最小特征值的下标。

## Q1.3 (1分) 画图。

```
# 徐丁涵 1900017794
import matplotlib.pyplot as plt
plt.quiver(eigVec[0][0],eigVec[0][1],color='green',units='x',scale=1)
plt.quiver(eigVec[1][0],eigVec[1][1],color='red',units='y',scale=1)
plt.xlim(-1.5,1.5)
plt.ylim(-1.5,1.5)
plt.show()
```



units和scale都要写。如果没有写scale，会自动填一个scale。没有写units，default使用width，红scale=4.5 绿scale=3。

## Q1.4 (1分) 计算Pearson相关系数。

```
# 陈福康 1900013049
np.random.seed(1)

xy = np.random.randn(2, 10)
# TODO
# Pearson相关系数包括样本相关系数和总体（变量）相关系数
# np.cov默认计算是样本协方差,除以的是N-1, 若bias= True则除以N
covar = np.cov(xy)
print(covar)
corr = covar[0,1]/np.sqrt(np.cov(xy[0])*np.cov(xy[1])) #当作样本计算
print(corr)

covar = np.cov(xy,bias = True)
print(covar)
corr = covar[0,1]/np.sqrt(np.var(xy[0])*np.var(xy[1])) #当作变量计算
print(corr)
```

原本作业给的答案有问题，分子分母使用的计算方法不同。

## Q1.5 (1分) 计算哈达玛矩阵。

```
# 熊梓因 2000017818
def hada(k):
    # TODO
    if k == 1:
        return np.array([[1, 1], [1, -1]])
    hada_pre = hada(k-1)
    up = np.hstack((hada_pre, hada_pre))
    down = np.hstack((hada_pre, np.multiply(hada_pre, -1)))
    return np.vstack((up, down))

h8 = hada(3)
print(h8 @ h8.T)
```

## Q1.6 (1分) 变成蛇形数组。

```
# 赖勋豪 1900017816
a = np.arange(9).reshape(3, 3)
# TODO
print(a, '\n')
a[[1, 2, 2, 1, 1], [2, 2, 1, 0, 1]] = a[[1, 1, 1, 2, 2], [0, 1, 2, 1, 2]]
print(a)
```

第一种：直接赋值需要的数值，或直接交换位置。

```
# 邓朝萌 1900013039
a = np.arange(9).reshape(3, 3)
# TODO
print(a)
a[1, 0] = a[1, 0] ^ a[1, 2]
a[1, 2] = a[1, 0] ^ a[1, 2]
a[1, 0] = a[1, 0] ^ a[1, 2]
a[1, 1] = a[1, 1] ^ a[2, 2]
a[2, 2] = a[1, 1] ^ a[2, 2]
a[1, 1] = a[1, 1] ^ a[2, 2]
a[1, 0] = a[1, 0] ^ a[2, 1]
a[2, 1] = a[1, 0] ^ a[2, 1]
a[1, 0] = a[1, 0] ^ a[2, 1]
print(a)
```

<b>0 1 2</b>	<b>0 1 2</b>	<b>0 1 2</b>	<b>0 1 2</b>
<b>3 4 5</b>	<b>5 4 3</b>	<b>5 8 3</b>	<b>7 8 3</b>
<b>6 7 8</b>	<b>6 7 8</b>	<b>6 7 4</b>	<b>6 5 4</b>



## Q1.6 (1分) 变成蛇形数组。

```
# 徐帆 1900017734
a = np.arange(9).reshape(3, 3)
print(a)
# TODO
a[1:3,1:3] = np.rot90(a[1:3,1:3], 2)
a[1] = np.flip(a[1])
print(a)
```

第二种：使用矩阵旋转/翻转变换。

<b>0 1 2</b>	<b>0 1 2</b>	<b>0 1 2</b>
<b>3 4 5</b>	<b>3 8 7</b>	<b>7 8 3</b>
<b>6 7 8</b>	<b>6 5 4</b>	<b>6 5 4</b>

## Q1.6 (1分) 变成蛇形数组。

```
# 时有恒 2000013151
# 对于任意n*n的数组都有效的递归做法，然而最后一次会多反转一次，需要反转回来
def change(X, n):

    if n == 1:
        return X

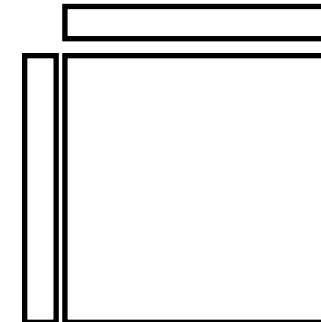
    X[1:] = np.hstack((change(X[1:].reshape(n, n-1)[1:], n-1),
                       X[1:].reshape(n, n-1)[0:1].T))

    X[:n, :n] = np.flip(X[:n, :n])

    return X

n = 8 # 这里可以设置为任意值
a = np.arange(n*n).reshape(n, n)
re = np.flip(change(a, n)) # 额外需要一次
print(re)
```

通解做法。



## Q2.1 (2分) PCA降维。

```
# 时有恒 2000013151
S = np.cov(X, rowvar=0) # S:样本协方差矩阵(4,4)
eigval, eigvec = np.linalg.eig(S) # 计算特征值与特征向量
# 排序
l = np.vstack((eigval, eigvec)).T
l = l[np.argsort(-l[:, 0])]
l = l[0:2].T

print("两个最大的特征值: ", l[0][0], l[0][1])

U = l[1:] # U:主成分特征矩阵(4,2)
Z = X @ U # Z降维后的矩阵(100,2)
reconstruct_X = (Z @ U.T) # 重构后的矩阵

# END TODO
print(reconstruct_X[7][2])
```

centerData按照行做  
norm, 所以是4个  
100维变量, 需要添  
加rowvar=0。

## Q2.2 (2分) SVD降维。

```
# 黄楠 1900012126
U, Sig, V = np.linalg.svd(A)
U = U[:, :3]
Sig = np.diag(Sig[:3])
V = V[:3, :]

np.allclose(A, U.dot(Sig).dot(V))
```

按照题目要求进行降维。np.allclose判断两个矩阵中的元素是不是相等，默认在 $1e-05$ 的误差范围内。

本题降维之后很难恢复原矩阵，这与选择特征值的数目有关。之后可能会讲解到如何确定特征值的数目来恢复矩阵。