矩阵大作业程序操作说明

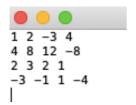
运行程序后,首先会要求输入存储待操作矩阵的文件路径及输出文件路径。然后会提供 5 种分解函数可选择,输入数字后,程序会自动进行不同的矩阵分解操作,如下图所示。

需注意,存储矩阵的.txt 文件中,行内各元素需用空格分隔,行间用换行符分隔。

```
Please input the filepath where store the matrix you want to manage:
/Users/kaifeiwang/Desktop/filename.txt
Please input the filepath where store the results:
/Users/kaifeiwang/Desktop/plu_out.txt
Please choose the function you want to do:
1: PA=LU分解
2: QR_Gram_Schmidt分解
3: Householder reduction
4: Givens reduction
5: URV分解
```

1. PA = LU 分解

使用课件上的例子:



作为输入矩阵,进行 PA=LU 分解。可以看到运行结果存储在自定义的文件中(终端也会输出结果,方便查看)。

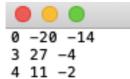
```
plu_out.txt
[ 0.000000
           1.000000 0.000000
                              0.000000]
[ 0.000000 0.000000
                     0.000000
                              1.0000001
[ 1.000000 0.000000
                     0.000000
                              0.000000]
[ 0.000000 0.000000 1.000000
                              0.000000]
L:
[ 1.000000
           0.000000 0.000000
                              0.000000]
[-0.750000
           1.000000
                     0.000000
                              0.000000]
                     1.000000
[ 0.250000 0.000000
                              0.000000]
[ 0.500000 -0.200000 0.333333
                              1.0000000]
U:
[ 4.000000
          8.000000
                     12.000000 -8.000000]
                    10.000000 -10.0000001
[ 0.000000
           5.000000
[ 0.000000
           0.000000 -6.000000 6.000000]
[ 0.000000 0.000000 0.000000
                              1.000000]
```

同时程序还做了异常处理,比如 PA=LU 的分解当 A 为非奇异矩阵时才存在,如果把上例中 A 矩阵的最后一行删掉,则程序会相应作出提醒,如下图所示。其他分解操作也会有类似的异常提醒。

```
Please choose the function you want to do:
1: PA=LU分解
2: QR_Gram_Schmidt分解
3: Householder reduction
4: Givens reduction
5: URV分解
1
Please cheak the format of matrix!!
```

2. Gram_Schmidt 分解

输入矩阵:

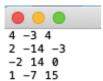


结果矩阵:

```
D:
[ 0.000000 -0.800000 -0.600000]
[ 0.600000 0.480000 -0.640000]
[ 0.800000 -0.360000 0.480000]
R:
[ 5.000000 25.000000 -4.000000]
[ 0.000000 25.000000 10.000000]
[ 0.000000 0.000000 10.000000]
```

3. Householder 分解

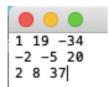
待分解矩阵:



结果矩阵:

4. Givens 分解

输入矩阵:



输出矩阵:

5. URV 分解

输入矩阵:

```
1 1 -1 2 1
0 0 1 3 -1
0 0 2 1 -2
```

输出矩阵:

```
U:
 [[ 1.000000 0.000000
                         0.000000]
             0.447214
                        0.894427]
 [ 0.000000
 [ 0.000000
             0.894427 -0.447214]]
R:
 [[ 1.414214 -1.414214
                         2.000000
                                              0.000000]
                                    0.000000
                        2.236068
 [ 0.000000
             3.162278
                                   0.000000 -0.000000]
 [ 0.000000 -0.000000
                        2.236068
                                   0.000000
                                             0.000000]]
۷:
 [[ 0.707107
              0.000000
                         0.000000 -0.707107 -0.000000]
 [ 0.707107
             0.000000
                        0.000000
                                   0.707107
                                             0.000000]
                                             0.707107]
 [ 0.000000
             0.707107
                        0.000000 -0.000000
 [ 0.000000 -0.000000
                        1.000000 -0.000000
                                            -0.000000]
 [ 0.000000 -0.707107 -0.000000
                                  0.000000
                                             0.707107]
```

同时,对于 URV 分解加了一个验证,U、V 是正交矩阵,R 是由零矩阵和 r*r 的可逆矩阵组成。明显 R 的前 r*r 子矩阵是非奇异的。(其中 r 代表输入矩阵的秩) 其余验证如下:

```
rank of input matrix:
W.T
[[ 1.000000
             0.000000
                        0.000000
                                   0.000000
                                             0.000000]
 [ 0.000000
                        0.000000
                                   0.000000
             1.000000
                                             0.000000]
 [ 0.000000
             0.000000
                        1.000000
                                   0.000000 - 0.000000]
 [ 0.000000
             0.000000
                        0.000000
                                   1.000000
                                             0.000000]
                                             1.000000]]
 [ 0.000000
             0.000000 -0.000000
                                   0.000000
UU.T
[[ 1.000000
                        0.000000]
             0.000000
 [ 0.000000
             1.000000 -0.000000]
 [ 0.000000 -0.000000
                        1.0000000]]
URV.T
[[ 1.000000
             1.000000 -1.000000
                                   2.000000 1.000000]
 [ 0.000000
             0.000000
                        1.000000
                                   3.000000 -1.000000]
                                   1.000000 -2.000000]]
 [ 0.000000
             0.000000
                        2.000000
```

可以看到 U 的转置是 U 的逆, V 的转置是 V 的逆, URV.T 的结果和原矩阵相等