李航老师《统计学习方法》第二版第十五章奇异值分解课后题答案

1、试求矩阵

$$A=\begin{bmatrix}1 & 2 & 0\\ 2 & 0 & 2\end{bmatrix}$$

的奇异值分解 。

解：感觉没有必要自己按照书上的算法流程算一遍，理论看懂，然后知道怎么计算就可以，无非也即是解个线性 方程，将特征向量正交化等等简单的线性代数知识。

```
import numpy as np
def SVD(A):
    U, s, V = np.linalg.svd(A)
    return U, s, V

if __name__ == '__main__':
    A = np.array([[1, 2, 0],
                  [2, 0, 2]])
    U, s, V = SVD(A)
    print('U is :', U)
    print('s is :', s)
    print('V is :', V)
```

输出是：

```
U is : [[ 0.4472136  -0.89442719]
 [ 0.89442719  0.4472136 ]]
s is : [3. 2.]
V is : [[ 7.45355992e-01  2.98142397e-01  5.96284794e-01]
 [ 1.94726023e-16 -8.94427191e-01  4.47213595e-01]
 [-6.66666667e-01  3.33333333e-01  6.66666667e-01]]
```

2、试求矩阵

$$A=\begin{bmatrix}2 & 4\\ 1 & 3\\ 0 & 0\\ 0 & 0\end{bmatrix}$$

的奇异值分解并且写出外积表达式。

```
import numpy as np
def SVD(A):
    U, s, V = np.linalg.svd(A)
    return U, s, V

if __name__ == '__main__':
    A = np.array([[2, 4],
                  [1, 3],
                  [0, 0],
                  [0, 0]])
    U, s, V = SVD(A)
    print('U is :', U)
    print('s is :', s)
    print('V is :', V)
    #下面开始使用外积展开，这里计算2，也可以计算1的，将矩阵的秩降为1
    B = np.matmul(s[0]*U[:,0:1], V[0:1,:]) + np.matmul(s[1] * U[:, 1:2], V[1:2, :])
    print(A == B)
```


输出为：

```
U is : [[-0.81741556 -0.57604844  0.          0.        ]
 [-0.57604844  0.81741556  0.          0.        ]
 [ 0.          0.          1.          0.        ]
 [ 0.          0.          0.          1.        ]]
s is : [5.4649857  0.36596619]
V is : [[-0.40455358 -0.9145143 ]
 [-0.9145143   0.40455358]]
[[ True  True]
 [ True  True]
 [ True  True]
 [ True  True]]
```

3、试比较奇异值分解和对称矩阵对角化的异同。

相同点：都需要计算特征值和 特征向量
不同点：

1. 对称的矩阵的对角化只需要计算一次特征向量，但是求出的特征向量构成的矩阵可能不是正交矩阵，还需要进行**施密特正交化**，然后正交化之后的向量按照和特征值对应的位置摆放构成一个正交阵。到此处就分解完成。
2. 对于矩阵A的奇异值分解，需要先计算A^TA的特征值和**正交**的特征向量，需要对特征值开方构成奇异值，正交向量按照奇异值的顺序构成V，然后再按照A和U，V的关系式计算U，以及计算A^T的零子空间的正交基。

4、证明任何一个秩为1的矩阵可以写程两个向量的的外积形式，并给出实例。

证明：设矩阵A的秩为1
证法1：不使用奇异值分解。
因为矩阵A的秩是1，我们设$A=[A_1,A_2,...,A_n]$,因为矩阵的秩是1，那么必然存在i_0使得A_{i_0}不是0向量。秩为1说明，可以使用n-1个常数$c_1,...,c_{i_0-1},c_{i_0+1},...,c_n$乘上$A_{i_0}$再加到第$1,2,...,i_0-1,i_0+1,...,n$列使得矩阵的这些列的元素都为0
也就是

$$A_j=-c_j*A_{i_0},j=1,2,...,i_0-1,i_0+1,...,n$$

所以

$$\begin{aligned}A&=[-c_1A_{i_0},...,-c_{i_0-1}A_{i_0},A_{i_0},c_{i_0+1}A_{i_0},...,A_n]\\&=A_{i_0}*[-c_1,...,-c_{i_0-1},1,-c_{i_0+1},...,-c_n]\end{aligned}$$

因为A_{i_0}是一个列向量，因为得证。

证法2：使用奇异值分解算法
因为矩阵A秩为1,那么利用奇异值分解，我们可以得到

$$A=U\Sigma V$$

秩为1说明Σ里面只有一个第一个元素为正值,设为a，其余的都是0
所以按照外积的展开式可以得到$A=aU_1*V_1^T$
其中U_1是正交矩阵的第一列，是个列向量，V_1是正交矩阵V的第一列，也是个向量，得证。

5、对点击数据进行奇异值分解，并且解释得到的三个矩阵所表示的内容。

这里已经设计到了推荐算法了。
点击数据构成的矩阵是：

$$A=\begin{bmatrix}0&20&5&0&0\\10&0&0&3&0\\0&0&0&0&1\\0&0&1&0&0\end{bmatrix}$$

```python
import numpy as np
def SVD(A):
    U, s, V = np.linalg.svd(A)
    return U, s, V

if __name__ == '__main__':
    A = np.array([[0, 20, 5, 0, 0],
                  [10, 0, 0, 3, 0],
                  [0, 0, 0, 0, 1],
```

```
                   [0, 0, 1, 0, 0]])
    U, s, V = SVD(A)
    print('U is :', U)
    print('s is :', s)
    print('V is :', V)
```

输出是:

```
U is : [[ 9.99930496e-01 -1.01352447e-16  0.00000000e+00 -1.17899939e-02]
 [ 0.00000000e+00  1.00000000e+00  0.00000000e+00 -8.65973959e-15]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00  0.00000000e+00]
 [ 1.17899939e-02  8.65973959e-15  0.00000000e+00  9.99930496e-01]]
s is : [20.61695792 10.44030651  1.          0.97007522]
V is : [[ 0.00000000e+00  9.70007796e-01  2.43073808e-01  0.00000000e+00
   0.00000000e+00]
 [ 9.57826285e-01 -2.31404926e-16  8.02571613e-16  2.87347886e-01
   0.00000000e+00]
 [-0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   1.00000000e+00]
 [-7.97105437e-16 -2.43073808e-01  9.70007796e-01  0.00000000e+00
   0.00000000e+00]
 [ 2.87347886e-01 -1.01402229e-16  2.10571835e-16 -9.57826285e-01
   0.00000000e+00]]
```

首先解释矩阵V

	0	1	2	3	4
0	0	0.970008	0.243074	0	0
1	0.957826	-2.31405e-16	8.02572e-16	0.287348	0
2	-0	0	0	0	1
3	-7.97105e-16	-0.243074	0.970008	0	0

我们将每一列看作是一个URL，因为第五个奇异值为0，根据外积形式，第五行直接不要了。第一列的表示的URL1的第二个维度的特征较为显著，第一列表示的URL2的第一个位置的特征比较显著，第三列表示的URL3的第四个特征比较显著，，，我们等会要根据矩阵U来区分查询q会倾向于点击哪个URL，也就是当用户输入一个查询的时候，推荐系统应该首先推荐哪个URL链接给用户。

下面解释矩阵U

	0	1	2	3
0	0.99993	-1.01352e-16	0	-0.01179
1	0	1	0	-8.65974e-15
2	0	0	1	0
3	0.01179	8.65974e-15	0	0.99993

将U的每一列看作是一个查询，第一个查询q1倾向于第一个维度的特征比较显著的URL，看上面的V矩阵，可以知道，URL2的第一个维度的特征比较显著，因而输入查询q1的时候，推荐系统应该首先推荐URL2给用户，这也解释了为啥查询q1对于的URL2的点击次数为20。U的第二列是q2，偏向于第二个特征较大的URL，看上面的V发现，URL1的第二个位置的特征较为显著，对于输入q2时，点击URL1高达10次。

下面是奇异值

	0
0	20.617
1	10.4403
2	1
3	0.970075

可以视作URL各个特征的重要性。可以看到前面的两个比较重要。