习题15.2

试求矩阵

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

的奇异值分解并写出其外积展开式。

矩阵 A 的外积展开式；

a.计算矩阵A的奇异值分解

$$A^TA=\begin{bmatrix}5&11\\11&25\end{bmatrix}$$

$$(A^TA-\lambda I)x=0$$

$$\lambda^2-30\lambda+4=0$$

$$\lambda_1=15-\sqrt{221}\ \ \lambda_2=\sqrt{221}+15$$

- 将λ_1代入$(A^TA-\lambda I)x=0$，得到 $v1$
- 将λ_2代入$(A^TA-\lambda I)x=0$，得到 $v2$
- 归一化$v1,v2$得到V..
- 发现计算较为复杂，直接使用numpy库计算,算数求解过程如书中所述。

> 给定$m\times n$矩阵A，可以写出矩阵奇异值分解的计算过程。
> （1）求A^TA的特征值和特征向量。
> 计算对称矩阵$W=A^TA$。
> 求解特征方程
>
> $$(W-\lambda I)x=0$$
>
> 得到特征值λ_i，并将特征值由大到小排列
>
> $$\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n\geqslant 0$$
>
> 将特征值$\lambda_i\ (i=1,2,\cdots,n)$代入特征方程求得对应的特征向量。
> （2）求n阶正交矩阵V
> 将特征向量单位化，得到单位特征向量$v_1,v_2,\cdots,v_n$，构成n阶正交矩阵V
>
> $$V=[v_1\ v_2\ \cdots\ v_n]$$
>
> （3）求$m\times n$对角矩阵Σ
> 计算A的奇异值
>
> $$\sigma_i=\sqrt{\lambda_i},\quad i=1,2,\cdots,n$$
>
> 构造$m\times n$矩形对角矩阵Σ，主对角线元素是奇异值，其余元素是零
>
> $$\Sigma=\operatorname{diag}(\sigma_1,\sigma_2,\cdots,\sigma_n)$$
>
> （4）求m阶正交矩阵U
> 对A的前r个正奇异值，令
>
> $$u_j=\frac{1}{\sigma_j}Av_j,\quad j=1,2,\cdots,r$$
>
> 得到
>
> $$U_1=[u1\ u2\ \cdots\ u_r]$$
>
> 求A^T的零空间的一组标准正交基$u_{r+1},u_{r+2},\cdots,u_m$，令

> $$U_2 = [u_{r+1}\ u_{r+2}\ \cdots\ u_m]$$
>
> 并令
>
> $$U = [U_1\ U_2]$$
>
> （5）得到奇异值分解
>
> $$A = U\Sigma V^T$$

```
import numpy as np
A = np.array([[2, 4],
              [1, 3],
              [0, 0],
              [0, 0]])
U, S, V = np.linalg.svd(A)
np.set_printoptions()
print("U=", U)
print("S=", S)
print("V=", V.T)
```

```
U= [[-0.81741556 -0.57604844  0.          0.        ]
    [-0.57604844  0.81741556  0.          0.        ]
    [ 0.          0.          1.          0.        ]
    [ 0.          0.          0.          1.        ]]
S= [5.4649857  0.36596619]
V= [[-0.40455358 -0.9145143 ]
    [-0.9145143   0.40455358]]
```

b.根据奇异值分解的结果，求得外积展开式

矩阵A的外积展开式为

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T$$

其中

$$\sigma_1 = 5.4649857, \sigma_2 = 0.36596619$$

$$u_1 = \begin{bmatrix} -0.81741556 \\ -0.57604844 \\ 0 \\ 0 \end{bmatrix}$$

$$u_2 = \begin{bmatrix} -0.57604844 \\ 0.81741556 \\ 0 \\ 0 \end{bmatrix}$$

$$v_1^T = [-0.40455358, -0.9145143]$$

$$v_2^T = [-0.9145143, 0.40455358]$$

```
approx = S[0] * np.outer(U[:, 0], V[:, 0]) + S[1] * np.outer(U[:, 1], V[:, 1])
print("A=", approx)
```

```
A= [[ 2.  4.]
    [ 1.  3.]
    [-0.  0.]
    [-0.  0.]]
```

习题15.4

证明任何一个**秩为1**的矩阵可以写成两个向量的外积形式，并给出实例。

a.假设矩阵 A 的秩为1，求其奇异值分解

存在矩阵A的奇异值分解

$$A = U\Sigma V^T \tag{1}$$

根据矩阵A的秩与对角矩阵Σ的秩之间的关系：

$$\operatorname{rank}(\Sigma) = \operatorname{rank}(A) = 1$$

$\therefore$ 对角矩阵Σ的奇异值是降序排列的，所以其中的非零元素一定位于第一行第一列，其余元素均为零。所以，对角矩阵Σ可以写成如下形式：

$$\begin{bmatrix} \sigma_1 & \\ & O_{(m-1)\times(n-1)} \end{bmatrix}_{m\times n}$$

设 $a = [1, 0, \cdots, 0]^T_{m\times 1}, b = [\sigma_1, 0, \cdots, 0]^T_{n\times 1}$，可得：$\Sigma = ab^T$

代入式（1）中可得：

$$A = Uab^TV^T = (Ua)(Vb)^T$$

其中，Ua 是$m \times 1$阶的列向量，Vb 是$n \times 1$阶的列向量,即矩阵A可以写成向量Ua和向量Vb的外积形式，得证

b.实例

书15.5的矩阵$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \end{bmatrix}$的秩为1，矩阵$A$的奇异值分解为

$$\begin{aligned} A &= U\Sigma V^T \\ &= \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} & 0 \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{10} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \end{aligned}$$

得到：

$$\begin{aligned} \Sigma &= \begin{bmatrix} \sqrt{10} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [\sqrt{10},\ 0] = ab^T \\ Ua &= \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \\ 0 \end{bmatrix}, Vb = \begin{bmatrix} \sqrt{5} \\ \sqrt{5} \end{bmatrix} \end{aligned}$$

$$\therefore (Ua)(Vb)^T = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \end{bmatrix} = A$$

习题15.5

搜索中的点击数据记录用户搜索时提交的查询语句，点击的网页URL以及点击的次数构成一个二部图，其中一个结点集合$\{q_i\}$表示查询，另一个结点集合$\{u_j\}$表示URL，边表示点击关系，边上的权重表示点击次数。图15.2是一个简化的点击数据例。点击数据可以由矩阵表示，试对该矩阵进行奇异值分解，并解释得到的三个矩阵所表示的内容。

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

a.奇异值分解;

```
import numpy as np
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 = np.linalg.svd(A)
print("U=", U)
print("S=", S)
print("V=", V.T)
```

```
U= [[ 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= [20.61695792 10.44030651  1.          0.97007522]
V= [[ 0.00000000e+00  9.57826285e-01 -0.00000000e+00 -7.97105437e-16
   2.87347886e-01]
    [ 9.70007796e-01 -2.31404926e-16 -0.00000000e+00 -2.43073808e-01
  -1.01402229e-16]
    [ 2.43073808e-01  8.02571613e-16  0.00000000e+00  9.70007796e-01
   2.10571835e-16]
    [ 0.00000000e+00  2.87347886e-01  0.00000000e+00  0.00000000e+00
  -9.57826285e-01]
    [ 0.00000000e+00  0.00000000e+00  1.00000000e+00  0.00000000e+00
   0.00000000e+00]]
```

b.解释U、Σ、V代表的内容

1. **矩阵 U**：
   - 矩阵 U 的列向量是点击矩阵 A 的左奇异向量。
   - 每个左奇异向量表示一个查询在低维特征空间中的表示。
   - 例如，矩阵 U 的第一列向量与查询 q_1 的关系密切。
2. **矩阵 Σ**：
   - Σ 是一个对角矩阵，其对角元素是奇异值，表示点击矩阵 A 的能量或重要性。
   - 较大的奇异值表示更多的信息或较高的重要性，较小的奇异值表示较少的信息或较低的重要性。
   - 例如，奇异值 σ_1 和 σ_2 较大，表示前两个特征捕捉了矩阵 A 中的大部分信息。
3. **矩阵 V^T**：
   - 矩阵 V^T 的行向量是点击矩阵 A 的右奇异向量。
   - 每个右奇异向量表示一个 URL 在低维特征空间中的表示。
   - 例如，矩阵 V^T 的第一列表示网页 u_2 与特征 1 的相似度最高，因此在查询 q_1 时点击网页 u_2 的次数最多。

习题16.1

对以下样本数据进行主成分分析：

$$X = \begin{bmatrix} 2 & 3 & 3 & 4 & 5 & 7 \\ 2 & 4 & 5 & 5 & 6 & 8 \end{bmatrix}$$

编程解决

```python
import numpy as np

def pca_svd(samples, num_components):
    ######################################################################
    # Perform Principal Component Analysis (PCA) using  Singular Value Decomposition (SVD).
    # :param samples: The sample matrix (each column represents a sample).
    # :param num_components: The number of principal components to retain.
    # :return: The feature vector matrix V, the sample principal component matrix Y.
    ######################################################################
    num_samples = samples.shape[1]
    # Construct a new matrix T by normalizing the samples matrix
    T = samples.T / np.sqrt(num_samples - 1)
    # Perform truncated SVD on the matrix T
    U, S, V = np.linalg.svd(T)
    V = V[:, :num_components]
    # Compute the k×n sample principal component matrix
    return V, np.dot(V.T, samples)

# Sample data
X = np.array([[2, 3, 3, 4, 5, 7],
              [2, 4, 5, 5, 6, 8]])
X = X.astype("float64")
# Normalize variables
mean = np.mean(X, axis=1)
variance = np.var(X, axis=1)
for i in range(X.shape[0]):
    X[i] = (X[i, :] - mean[i]) / np.sqrt(variance[i])
# Set precision to 3 decimal places
np.set_printoptions(precision=3, suppress=True)
V, principal_components = pca_svd(X, 2)
print("Orthogonal matrix V:")
print(V)
print("Sample principal component matrix Y:")
print(principal_components)
```

```
Orthogonal matrix V:
[[ 0.707  0.707]
 [ 0.707 -0.707]]
Sample principal component matrix Y:
[[-2.028 -0.82  -0.433  0.     0.82   2.461]
 [ 0.296 -0.046 -0.433  0.     0.046  0.137]]
```

习题16.2

证明样本协方差矩阵S是总体协方差矩阵Σ的无偏估计。

无偏估计的定义

> 如果参数θ的某个估计$\hat{\theta}$满足$E(\hat{\theta}) = \theta$，则称$\hat{\theta}$是无偏的。
>
> $$\hat{\Sigma} = \frac{1}{n} A^d = \frac{1}{n} \sum_{\alpha=1}^{n-1} z_\alpha z_\alpha'$$
>
> $z_1, \cdots, z_{n-1}$独立同分布于$N_p(0, \Sigma)$，于是
>
> $$E(\hat{\Sigma}) = \frac{1}{n} \sum_{\alpha=1}^{n-1} z_\alpha z_\alpha' = \frac{n-1}{n} \Sigma$$
>
> $\hat{\Sigma}$不是Σ的无偏估计，但可修正一下使之为无偏估计。令
>
> $$S = \frac{1}{n-1} A$$
>
> 则S是Σ的无偏估计。在实用中普遍采用S来估计Σ。

样本协方差矩阵S：

$$S=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})((x_i-\bar{x}))'$$

令：

$$A=\sum_{i=1}^{n}(x_i-\bar{x})((x_i-\bar{x}))'$$

即证明样本协方差矩阵S的期望等于总体协方差矩阵Σ：

$$E(S)=E(\frac{1}{n-1}A)=\frac{1}{n-1}E(A)=\Sigma$$

即证：

$$E(A)=(n-1)\Sigma$$

$$\begin{aligned} E(A) &= E\Big[\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})'\Big] \\ &= nE(\hat{\Sigma}) \\ &= n\cdot\frac{n-1}{n}\Sigma \\ &= (n-1)\Sigma \end{aligned}$$

- 由此可知，样本协方差矩阵S是总体协方差矩阵Σ的无偏估计。

习题16.3

设X为数据规范化样本矩阵，则主成分等价于求解以下最优化问题：

$$\begin{array}{ll} \min\limits_{L} & \|X-L\|_F \\ \text{s.t.} & \operatorname{rank}(L)\leqslant k \end{array}$$

其中，F是弗罗贝尼乌斯范数，k是主成分个数。试问为什么？

> **定义15.4（弗罗贝尼乌斯范数）** 设矩阵$A\in R^{m\times m}$，$A=[a_{ij}]_{m\times m}$，定义矩阵A的弗罗贝尼乌斯范数为
>
> $$\|A\|_F=\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij})^2\right)^{\frac{1}{2}}$$

优化问题的目标函数进行如下变换：

$$\begin{array}{ll} \min\limits_{L} & \left\|\frac{1}{\sqrt{n-1}}X^T-\frac{1}{\sqrt{n-1}}L^T\right\|_F \\ \text{s.t.} & \operatorname{rank}(L)\leqslant k \end{array}$$

令

$$X'=\frac{1}{\sqrt{n-1}}X^T, L'=\frac{1}{\sqrt{n-1}}L^T$$

原最优化问题等价于

$$\begin{array}{ll} \min\limits_{L} & \|X'-L'\|_F \\ \text{s.t.} & \operatorname{rank}(L)\leqslant k \\ & X'=\frac{1}{\sqrt{n-1}}X^T \\ & L'=\frac{1}{\sqrt{n-1}}L^T \end{array}$$

对X'进行奇异值分解，X'的主成分个数也为k个，则

$$X'=U'\Sigma'V'^T$$

可找到 $\Sigma'=\operatorname{diag}(\lambda_1',\lambda_2',\cdots,\lambda_k',0,0,\cdots,0)$ 使得变换后的目标函数取最小值，即存在矩阵满足主成分条件。