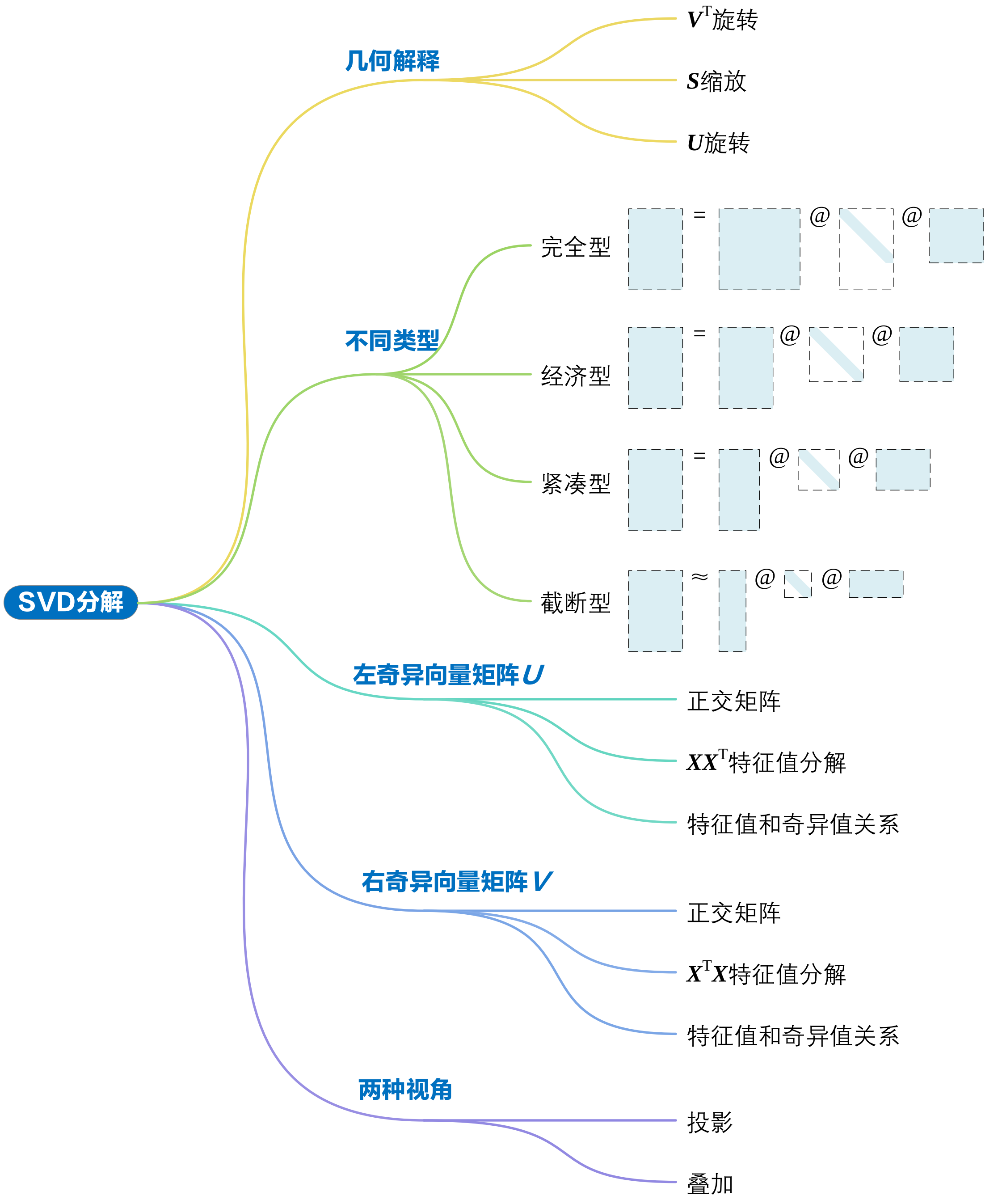
15 Singular Value Decomposition 奇异值分解

应该是最重要的矩阵分解，没有之一

> 人不过是一根芦苇，世界最脆弱的生灵；但是，人是会思考的芦苇。
>
> ***Man is but a reed, the most feeble thing in nature, but he is a thinking reed.***
>
> —— 布莱兹·帕斯卡 (Blaise Pascal) | 法国哲学家、科学家 | 1623 ~ 1662

- ◄ `matplotlib.pyplot.quiver()` 绘制箭头图
- ◄ `numpy.linspace()` 在指定的间隔内，返回固定步长的数据
- ◄ `numpy.linalg.svd()` 进行 SVD 分解
- ◄ `numpy.diag()` 以一维数组的形式返回方阵的对角线元素，或将一维数组转换成对角阵


SVD分解
几何解释
V^T旋转
S缩放
U旋转
不同类型
完全型
经济型
紧凑型
截断型
=
≈
@
左奇异向量矩阵 U
正交矩阵
XX^T特征值分解
特征值和奇异值关系
右奇异向量矩阵 V
正交矩阵
X^TX特征值分解
特征值和奇异值关系
两种视角
投影
叠加

15.1 几何视角：旋转 → 缩放 → 旋转

本书前文简要介绍过**奇异值分解** (Singular Value Decomposition, SVD)，这个宇宙中最重要的矩阵分解。本节将从几何视角解剖奇异值分解。

对矩阵 $\boldsymbol{X}_{n\times D}$ 奇异值分解得到：

$$\boldsymbol{X}_{n\times D} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}} \tag{1}$$

其中，$\boldsymbol{S}$ 为对角阵，其主对角线元素 s_i 为**奇异值** (singular value)。

$\boldsymbol{U}$ 的列向量称作**左奇异值向量** (left singular vector)。

$\boldsymbol{V}$ 的列向量称作**右奇异值向量** (left singular vector)。

特别注意，(1) 中矩阵 $\boldsymbol{V}$ 的转置运算。

$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 为正交矩阵。即，$\boldsymbol{U}$ 和自己转置 $\boldsymbol{U}^{\mathrm{T}}$ 的乘积为单位矩阵；$\boldsymbol{V}$ 和自己转置 $\boldsymbol{V}^{\mathrm{T}}$ 的乘积也是单位矩阵。

根据这三个矩阵的形态，我们知道正交矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 矩阵作用是旋转，而对角矩阵 $\boldsymbol{S}$ 的作用是缩放。

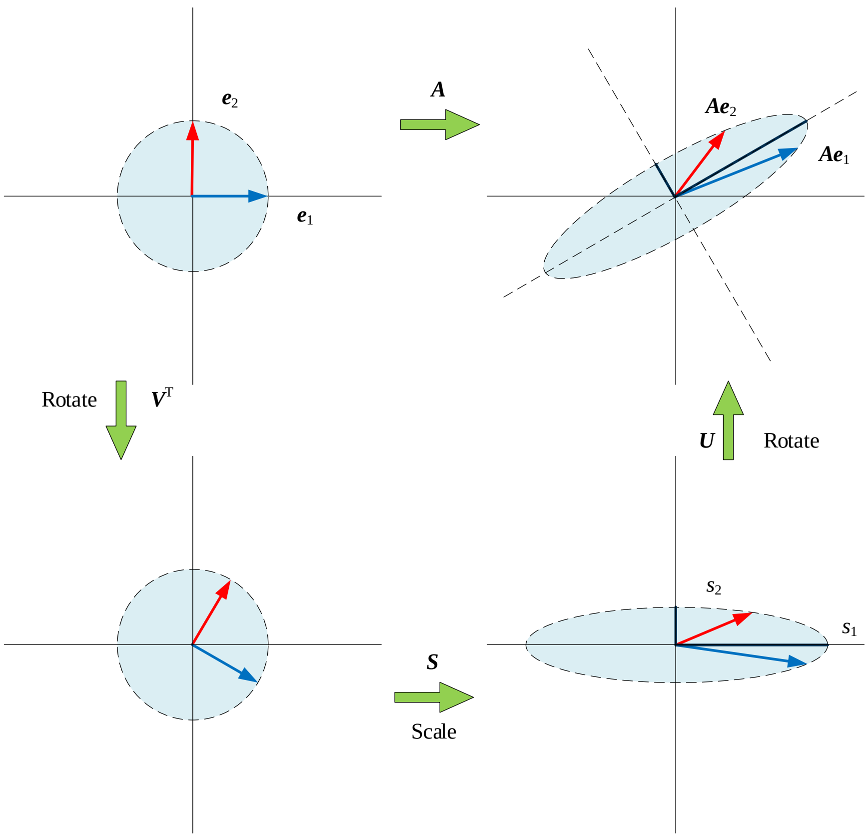


图 1. 几何角度解释奇异值分解

大家可能会问这和特征值分解对应的“”有何不同？

特征值分解中，三步几何变换是旋转 ($\boldsymbol{V}^{-1}$) → 缩放 ($\boldsymbol{\Lambda}$) →旋转 ($\boldsymbol{V}$)。

奇异值分解中，三步几何变换是旋转 ($\boldsymbol{V}^{\mathrm{T}}$) → 缩放 ($\boldsymbol{S}$) →旋转 ($\boldsymbol{U}$)。

为了方便解释，我们用 2 × 2 矩阵 $\boldsymbol{A}$ 做例子。

矩阵 $\boldsymbol{A}$ 坐标向量 $\boldsymbol{x}$ 得到 $\boldsymbol{z}$。对 $\boldsymbol{A}$ 奇异值分解，得到 $\boldsymbol{U}$、$\boldsymbol{S}$ 和 $\boldsymbol{V}$ 三个矩阵：

$$\boldsymbol{A}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underset{\text{Rotate}}{\boldsymbol{U}}\ \underset{\text{Scale}}{\boldsymbol{S}}\ \underset{\text{Rotate}}{\boldsymbol{V}^{\mathrm{T}}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{2}$$

图 1 所示为几何角度解释奇异值分解，$\boldsymbol{A}$ 乘 $[x_1, x_2]^{\mathrm{T}}$，相当于先用 $\boldsymbol{V}^{\mathrm{T}}$ 旋转，再用 $\boldsymbol{S}$ 缩放，最后用 $\boldsymbol{U}$ 旋转。

举个实例

下面用具体实例解释图 1。

给定 2 × 2 矩阵 $\boldsymbol{A}$：

$$\boldsymbol{A} = \begin{bmatrix} 1.625 & 0.6495 \\ 0.6495 & 0.875 \end{bmatrix} \tag{3}$$

对矩阵 $\boldsymbol{A}$ 进行 SVD 分解：

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}} = \underbrace{\begin{bmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{bmatrix}}_{\boldsymbol{U}} \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}}_{\boldsymbol{S}} \underbrace{\begin{bmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{bmatrix}}_{\boldsymbol{V}^{\mathrm{T}}} \tag{4}$$

即，

$$\boldsymbol{U} = \begin{bmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{bmatrix},\quad \boldsymbol{S} = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix},\quad \boldsymbol{V} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix} \tag{5}$$

给定 $\boldsymbol{e}_1$ 和 $\boldsymbol{e}_2$ 两个单位向量：

$$\boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\quad \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{6}$$

$\boldsymbol{e}_1$ 和 $\boldsymbol{e}_2$ 经过 $\boldsymbol{A}$ 转换分别得到：

$$\begin{aligned} \boldsymbol{A}\boldsymbol{e}_1 &= \begin{bmatrix} 1.625 & 0.6495 \\ 0.6495 & 0.875 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.625 \\ 0.6495 \end{bmatrix} \\ \boldsymbol{A}\boldsymbol{e}_2 &= \begin{bmatrix} 1.625 & 0.6495 \\ 0.6495 & 0.875 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.6495 \\ 0.875 \end{bmatrix} \end{aligned} \tag{7}$$

图 2 所示为转换前后的结果对比。请大家分别注意转换前后向量的方向和长度 (模) 的变化。

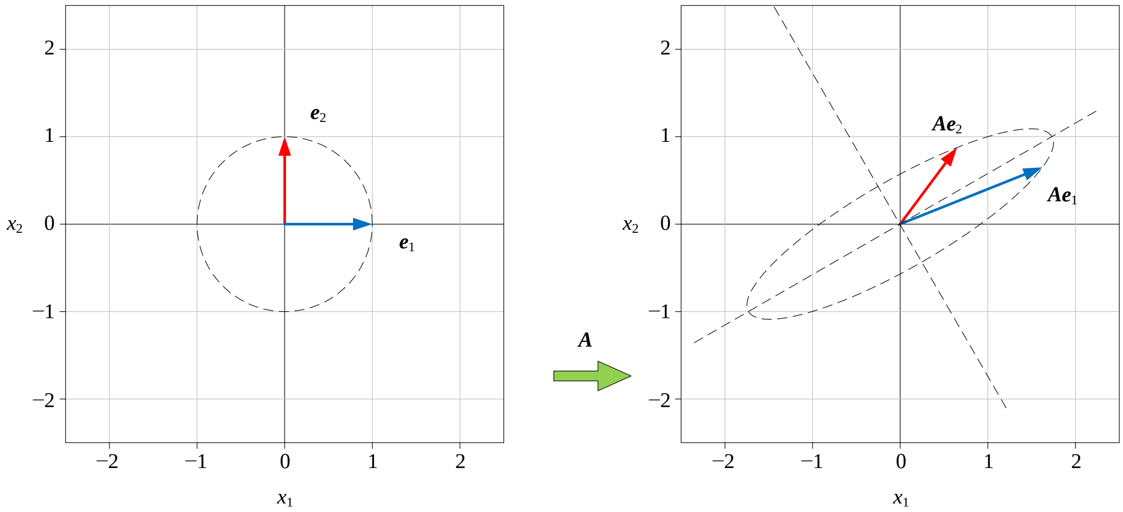


图 2. e_1 和 e_2 经过 A 线性转换

分步几何变换

(7) 等价于"旋转 (V^T) → 缩放 (S) → 旋转 (U)", 具体如图 3 所示。

e_1 和 e_2 两个向量先通过 V^T 进行旋转，得到：

$$
\begin{aligned}
V^{T}e_1 &= \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.866 \\ -0.5 \end{bmatrix} \\
V^{T}e_2 &= \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.866 \end{bmatrix}
\end{aligned} \tag{8}
$$

在 (8) 基础上，再用对角矩阵 S 进行缩放，得到：

$$
\begin{aligned}
SV^{T}e_1 &= \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}\begin{bmatrix} 0.866 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 1.732 \\ -0.25 \end{bmatrix} \\
SV^{T}e_2 &= \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}\begin{bmatrix} 0.5 \\ 0.866 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.433 \end{bmatrix}
\end{aligned} \tag{9}
$$

在之前"旋转"和"缩放"两步基础上，最后再利用 U 进行旋转，得到：

$$
\begin{aligned}
USV^{T}e_1 &= \begin{bmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{bmatrix}\begin{bmatrix} 1.732 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 1.625 \\ 0.6495 \end{bmatrix} \\
USV^{T}e_2 &= \begin{bmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{bmatrix}\begin{bmatrix} 1 \\ 0.433 \end{bmatrix} = \begin{bmatrix} 0.6495 \\ 0.875 \end{bmatrix}
\end{aligned} \tag{10}
$$

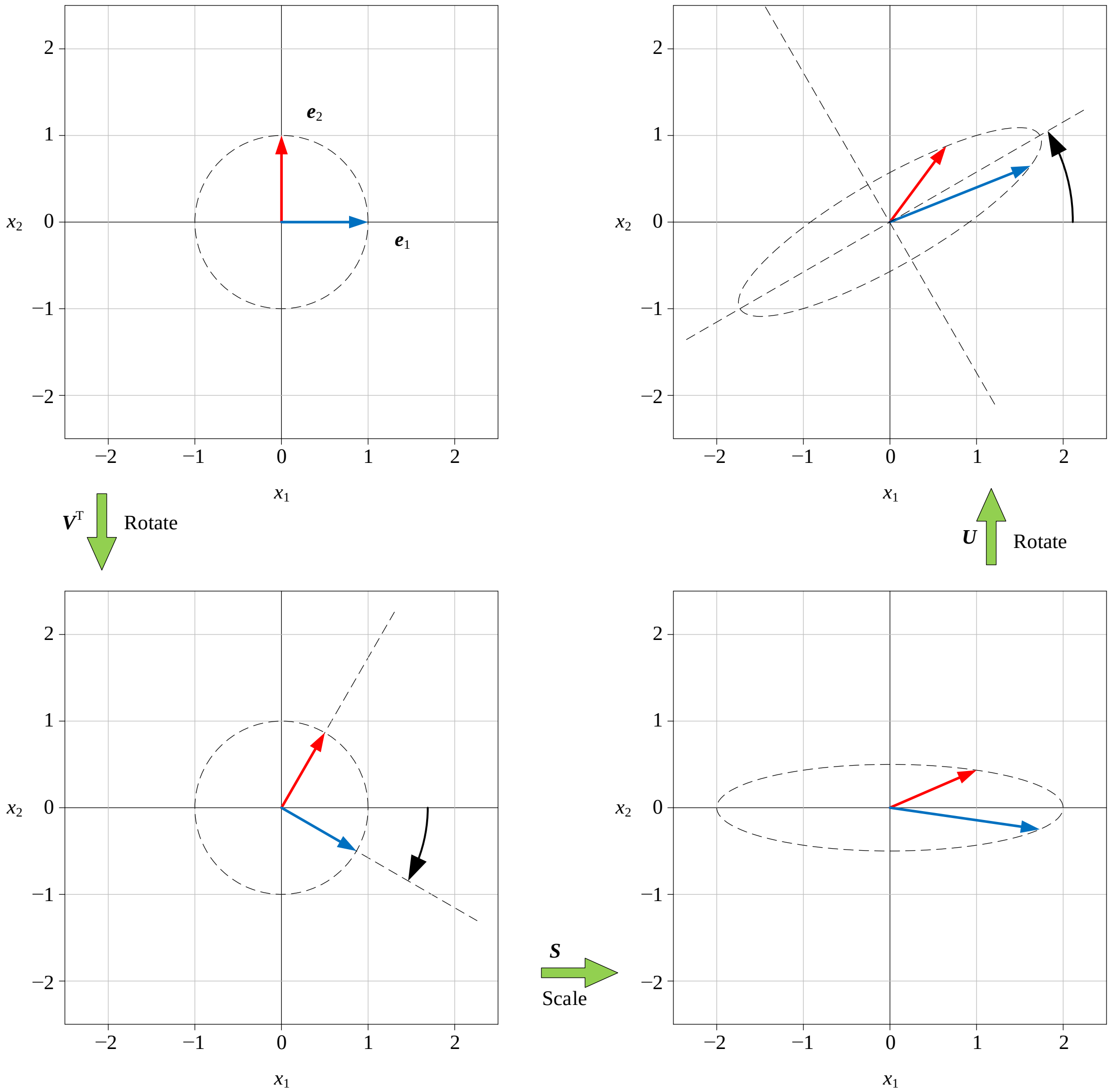


图 3. e_1 和 e_2 分别经过 V^T、S 和 U 转换

Bk4_Ch15_01.py 绘制图 3 所有子图。

```
# Bk4_Ch15_01.py

import numpy as np
import matplotlib.pyplot as plt

def visualize(X_circle,X_vec,title_txt):

    fig, ax = plt.subplots()

    plt.plot(X_circle[:,0], X_circle[:,1],'k',
             linestyle = '--',
             linewidth = 0.5)

    plt.quiver(0,0,X_vec[0,0],X_vec[0,1],
```

```
               angles='xy', scale_units='xy',scale=1,
               color = [0, 0.4392, 0.7529])

    plt.quiver(0,0,X_vec[1,0],X_vec[1,1],
               angles='xy', scale_units='xy',scale=1,
               color = [1,0,0])

    plt.axvline(x=0, color= 'k', zorder=0)
    plt.axhline(y=0, color= 'k', zorder=0)

    plt.ylabel('$x_2$')
    plt.xlabel('$x_1$')

    ax.set_aspect(1)
    ax.set_xlim([-2.5, 2.5])
    ax.set_ylim([-2.5, 2.5])
    ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
    ax.set_xticks(np.linspace(-2,2,5));
    ax.set_yticks(np.linspace(-2,2,5));
    plt.title(title_txt)
    plt.show()

theta = np.linspace(0, 2*np.pi, 100)

circle_x1 = np.cos(theta)
circle_x2 = np.sin(theta)

X_vec = np.array([[1,0],
                 [0,1]])

X_circle = np.array([circle_x1, circle_x2]).T

# plot original circle and two vectors
visualize(X_circle,X_vec,'Original')

A = np.array([[1.6250, 0.6495],
              [0.6495, 0.8750]])

# plot the transformation of A

visualize(X_circle@A.T, X_vec@A.T,'$A$')

#%% SVD
# A = U @ S @ V.T

U, S, V = np.linalg.svd(A)
S = np.diag(S)
V[:,0] = -V[:,0] # reverse sign of first vector of V
U[:,0] = -U[:,0] # reverse sign of first vector of U

print('=== U ===')
print(U)
print('=== S ===')
print(S)
print('=== V ===')
print(V)

# plot the transformation of V

visualize(X_circle@V, X_vec@V,'$V^T$')

# plot the transformation of V @ S

visualize(X_circle@V@S, X_vec@V@S,'$SV^T$')

# plot the transformation of V @ S @ U.T

visualize(X_circle@V@S@U.T, X_vec@V@S@U.T,'$USV^T$')

e1 = np.array([[1],
               [0]])
```

```
e2 = np.array([[0],
               [1]])

# Calculate step by step from e1 and e2
VT_e1 = V.T@e1
VT_e2 = V.T@e2

S_VT_e1 = S@VT_e1
S_VT_e2 = S@VT_e2

U_S_VT_e1 = U@S_VT_e1
U_S_VT_e2 = U@S_VT_e2
```

15.2 不同类型 SVD 分解

SVD 分解分为**完全型** (full)、**经济型** (economy-size, thin)、**紧凑型** (compact) 和**截断型** (truncated)。

本节将简要介绍完全型和经济型两种奇异值分解之间的关系。下一章将深入讲解这四种 SVD 分解。

完全型

图 4 所示为完全型 SVD 分解热图。左奇异值矩阵 $\boldsymbol{U}$ 为方阵，形状为 $n \times n$。$\boldsymbol{S}$ 的形状和 $\boldsymbol{X}$ 相同，为 $n \times D$。$\boldsymbol{S}$ 的主对角线元素 s_i 为奇异值，具体形式为：

$$\boldsymbol{S} = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D \\ & & & \\ & & & \end{bmatrix} \tag{11}$$

如图 4 所示，$\boldsymbol{S}$ 可以分块为上下两个子块——方阵对角矩阵和全 0 矩阵 $\boldsymbol{O}$。右奇异值矩阵 $\boldsymbol{V}$ 形状为 $D \times D$。

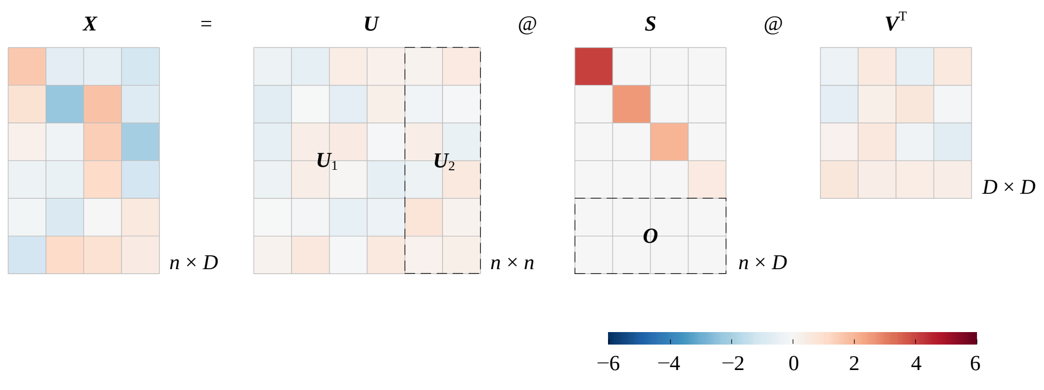


图 4. 完全型 SVD 分解

注意，一般情况，数据矩阵为“细高”长方形，偶尔大家也会见到“宽矮”长方形的数据矩阵。(1) 中 $\boldsymbol{X}$ 就是细高长方形，对 $\boldsymbol{X}$ 转置便得到宽矮长方形。相应的，$\boldsymbol{X}^{\mathrm{T}}$ 的 SVD 分解为：

$$\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} \tag{12}$$

经济型

图 5 所示为经济型 SVD 分解结果热图。可以发现，左奇异值矩阵 $\boldsymbol{U}$ 形状和 $\boldsymbol{X}$ 相同，均为为 $n \times D$。而 $\boldsymbol{S}$ 为方阵。

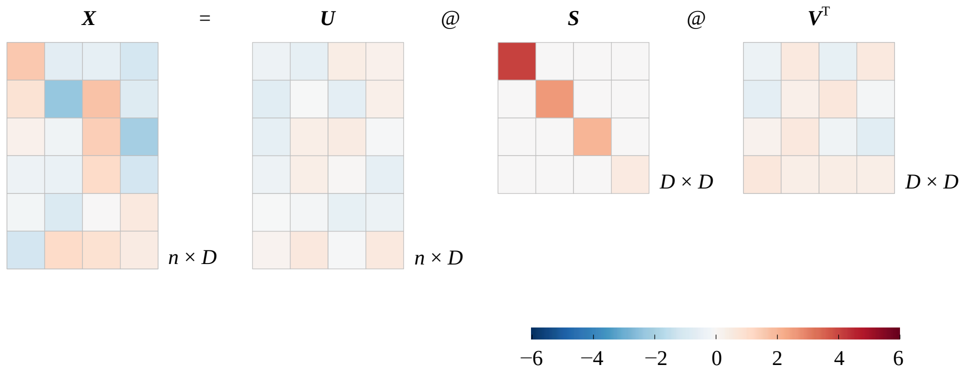


图 5. 经济型 SVD 分解

在经济型 SVD 分解中，$\boldsymbol{S}$ 为对角方阵：

$$\boldsymbol{S} = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D \end{bmatrix} \tag{13}$$

当 $\boldsymbol{S}$ 为对角方阵时，(12) 可以写成：

$$\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^{\mathrm{T}} \tag{14}$$

Bk4_Ch15_02.py 中 Bk4_Ch15_02_A 部分绘制图 4 和图 5。

```
# Bk4_Ch15_02_A

import numpy as np
from matplotlib import pyplot as plt
plt.rcParams['image.cmap'] = 'RdBu_r'
import seaborn as sns
PRECISION = 3

def svd(X):
```

```
    full_matrices = True

    U, s, Vt = np.linalg.svd(X,full_matrices = full_matrices)

    # Put the vector singular values into a padded matrix

    if full_matrices:
        S = np.zeros(X.shape)
        np.fill_diagonal(S, s)
    else:
        S = np.diag(s)

    # Rounding for display
    return np.round(U, PRECISION), np.round(S, PRECISION), np.round(Vt.T, PRECISION)


def visualize_svd(X,title_X,title_U,title_S,title_V, fig_height=5):

    # Run SVD, as defined above
    U, S, V = svd(X)

    all_ = np.r_[X.flatten(order='C'),U.flatten(order='C'),
                 S.flatten(order='C'),V.flatten(order='C')]

    # all_max = max(all_.max(),all_.min())
    # all_min = -max(all_.max(),all_.min())
    all_max = 6
    all_min = -6
    # Visualization
    fig, axs = plt.subplots(1, 7, figsize=(12, fig_height))

    plt.sca(axs[0])
    ax = sns.heatmap(X,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                     cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title(title_X)

    plt.sca(axs[1])
    plt.title('@')
    plt.axis('off')

    plt.sca(axs[2])
    ax = sns.heatmap(U,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                     cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title(title_U)

    plt.sca(axs[3])
    plt.title('@')
    plt.axis('off')

    plt.sca(axs[4])
    ax = sns.heatmap(S,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                     cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title(title_S)

    plt.sca(axs[5])
    plt.title('@')
    plt.axis('off')

    plt.sca(axs[6])
    ax = sns.heatmap(V.T,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                     cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title(title_V)
    return X, U, S, V

# Repeatability
np.random.seed(1)
```

```
# Generate random matrix
X = np.random.randn(6, 4)

# manipulate X and reduce rank to 3
# X[:,3] = X[:,0] + X[:,1]

X, U, S, V = visualize_svd(X,'$X$','$U$','$S$','$V^T$', fig_height=3)

X_2, U_2, S_2, V_2 = visualize_svd(X.T@X,'$X^TX$','$V$','$S^TS$','$V^T$', fig_height=3)

X_3, U_3, S_3, V_3 = visualize_svd(X@X.T,'$XX^T$','$U$','$SS^T$','$U^T$', fig_height=3)
```

15.3 左奇异向量矩阵 U

$\boldsymbol{U}$ 的列向量称作左奇异值向量 (left singular vector)，$\boldsymbol{U}$ 和自己转置 $\boldsymbol{U}^{\mathrm{T}}$ 的乘积为单位矩阵：

$$\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}=\boldsymbol{I} \tag{15}$$

图 6 所示为运算热图。

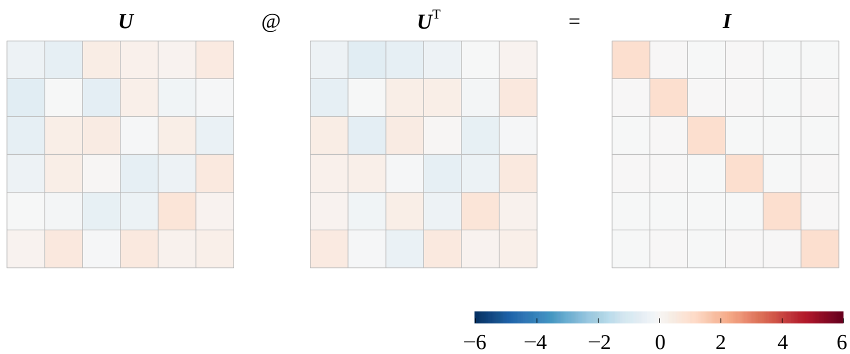


图 6. $\boldsymbol{U}$ 和自己转置 $\boldsymbol{U}^{\mathrm{T}}$ 的乘积为单位矩阵

特征值分解

图 7 所示为 $\boldsymbol{X}$ 和自己转置 $\boldsymbol{X}^{\mathrm{T}}$ 相乘得到 Gram 矩阵 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 的热图，$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 为 $n \times n$ 方阵。

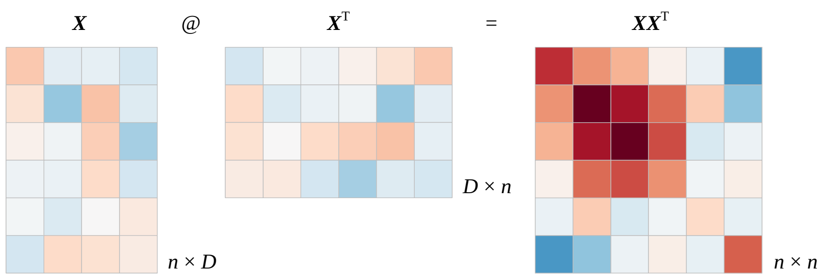


图 7. $\boldsymbol{X}$ 和自己转置 $\boldsymbol{X}^{\mathrm{T}}$ 的乘积热图

对方阵 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 进行特征值分解，可以发现 $\boldsymbol{U}$ 是特征向量，而 $\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$ 是特征值矩阵：

$$\begin{aligned}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} &= \left(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\right)\left(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \boldsymbol{U}\boldsymbol{S}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right)\boldsymbol{S}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} \\ &= \boldsymbol{U}\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\end{aligned} \tag{16}$$

图 8 所示为 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 特征值分解热图。

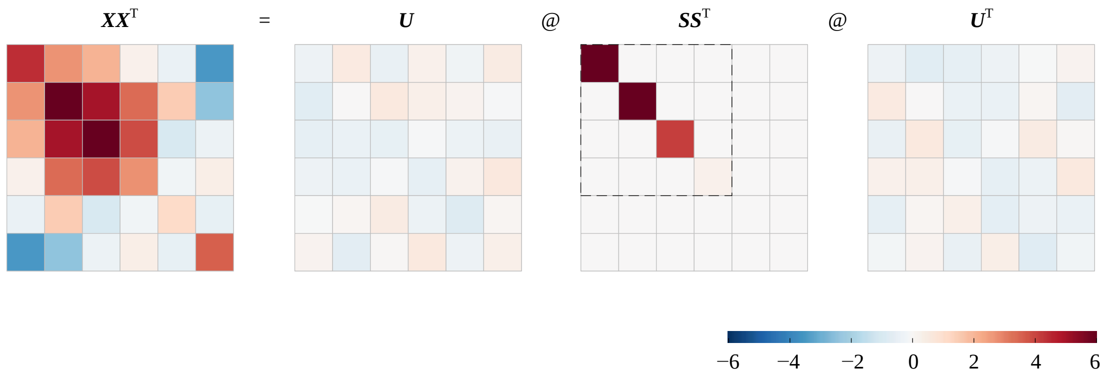


图 8. 对 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 特征值分解

$\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$ 主对角线为特征值，对 $\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$ 展开得到：

$$\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix} s_1 & & & & & \\ & s_2 & & & & \\ & & \ddots & & & \\ & & & s_D & & \end{bmatrix} = \begin{bmatrix} s_1^2 & & & & & \\ & s_2^2 & & & & \\ & & \ddots & & & \\ & & & s_D^2 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix} = \begin{bmatrix} \lambda_1 & & & & & \\ & \lambda_2 & & & & \\ & & \ddots & & & \\ & & & \lambda_D & & \\ & & & & \ddots & \\ & & & & & \lambda_n \end{bmatrix} \tag{17}$$

观察上式，发现当 $j = 1 \sim D$ 时，特征值 λ_j 和奇异值 s_j 存在如下关系：

$$\lambda_j = s_j^2 \tag{18}$$

特别地，在经济型 SVD 分解中，$\boldsymbol{S}$ 为对角方阵；因此 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 特征值分解可以写作：

$$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{S}^2\boldsymbol{U}^{\mathrm{T}} \tag{19}$$

向量空间

本书前文提到过两次，细高的长方形矩阵 $\boldsymbol{X}$ 不能进行特征值分解。但是，它的格拉姆矩阵 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 是对称矩阵，形状为 $n \times n$，可以进行特征值分解。

如图 9 所示，$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 进行特征值分解得到正交矩阵 $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]$，它是个规范正交基，张起的空间为 $\mathbb{R}^n$。

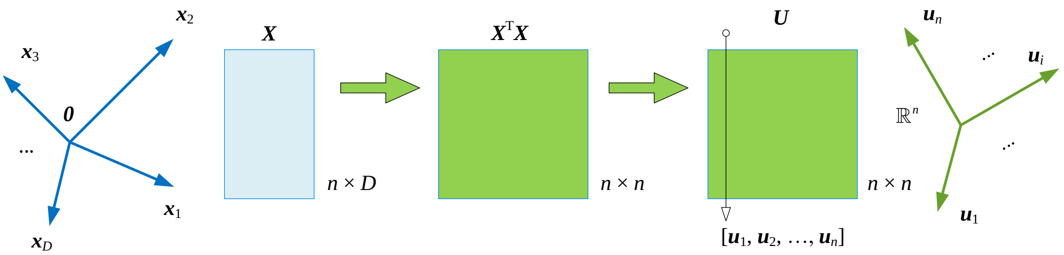


图 9. 对 Gram 矩阵 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 特征值分解得到规范正交基 $\boldsymbol{U}$

类别 QR 分解

对于数据矩阵 $\boldsymbol{X}$ 进行 QR 分解得到：

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R} \tag{20}$$

对于完全型 QR 分解，$\boldsymbol{Q}$ 为正交矩阵，即一个规范正交基 $[\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_D]$。

对 $\boldsymbol{X}$ 进行完全型 SVD 分解，把结果写成：

$$\boldsymbol{X} = \boldsymbol{U}\left(\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\right) \tag{21}$$

对比 (20) 和 (21)，$\boldsymbol{Q}$ 和 $\boldsymbol{U}$ 都是正交矩阵，形状相同；但是，两者显然不同。对于 QR 分解，$\boldsymbol{x}_1$ 和 $\boldsymbol{q}_1$ 平行。打个比方，$\boldsymbol{x}_1$ 像是一个锚，确定了 $[\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_D]$ 的空间位置。而 SVD 分解需要一个优化视角，逐个最大化奇异值。本书第 18 章将深入介绍这个优化视角。

对比 (20) 和 (21)，$\boldsymbol{R}$ 则对应 $\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$。

Bk4_Ch15_02.py 中 Bk4_Ch15_02_B 部分绘制图 6。请读者自行编写代码绘制图 7 和图 8。

```
# Bk4_Ch15_02_B

#%% U*U.T = I

all_max = 6
all_min = -6

fig, axs = plt.subplots(1, 3, figsize=(12, 3))

plt.sca(axs[0])
ax = sns.heatmap(U,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
ax.set_aspect("equal")
plt.title('$U$')


plt.sca(axs[1])
ax = sns.heatmap(U.T,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
ax.set_aspect("equal")
plt.title('$U^T$')
```

```
plt.sca(axs[2])
ax = sns.heatmap(U@U.T,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
ax.set_aspect("equal")
plt.title('$I$')
```

15.4 右奇异向量矩阵 *V*

$\boldsymbol{V}$ 的列向量称作**右奇异值向量** (left singular vector)，$\boldsymbol{V}$ 和其转置 $\boldsymbol{V}^{\mathrm{T}}$ 的乘积也是单位矩阵：

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}=\boldsymbol{I} \tag{22}$$

图 10 所示为上式运算对应热图。

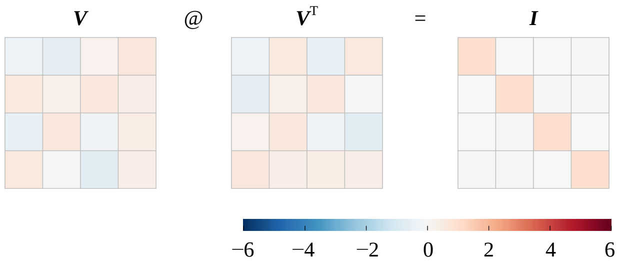


图 10. $\boldsymbol{V}$ 和其转置 $\boldsymbol{V}^{\mathrm{T}}$ 的乘积也是单位矩阵

特征值分解

图 11 所示为转置 $\boldsymbol{X}^{\mathrm{T}}$ 和 $\boldsymbol{X}$ 相乘得到 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的热图，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 为 $D\times D$ 方阵。

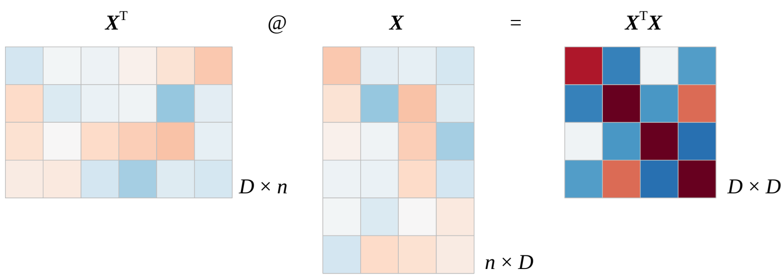


图 11. 转置 $\boldsymbol{X}^{\mathrm{T}}$ 和 $\boldsymbol{X}$ 乘积热图

对 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 特征值分解得到：

$$\begin{aligned}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}&=\left(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\right)\\&=\boldsymbol{V}\boldsymbol{S}^{\mathrm{T}}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right)\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\\&=\boldsymbol{V}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\end{aligned} \tag{23}$$

$\boldsymbol{V}$ 是 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的特征向量，$\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}$ 为特征值矩阵。图 12 所示为对 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 进行特征值分解热图。

特别地，在经济型 SVD 分解中，$\boldsymbol{S}$ 为对角方阵；因此 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 特征值分解可以写成：

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^{\mathrm{T}} \tag{24}$$

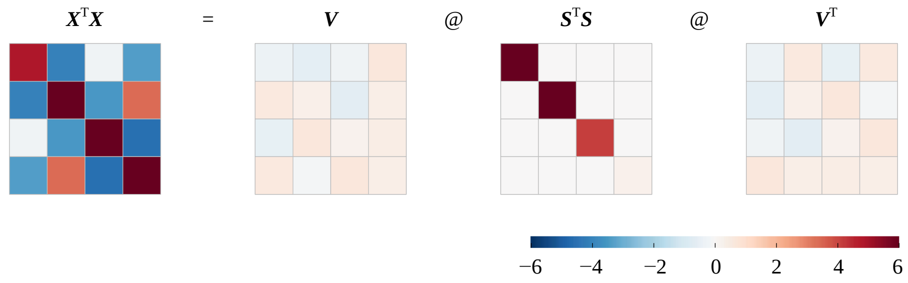


图 12. 对 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 进行特征值分解

前文提过，经济型 SVD 分解中，$\boldsymbol{S}$ 为对角方阵，$\boldsymbol{S}$ 主对角元素 s_i 为奇异值：

$$\boldsymbol{S} = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D \end{bmatrix} \tag{25}$$

对 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 进行特征值分解，$\boldsymbol{S}^2$ 为特征值矩阵：

$$\boldsymbol{S}^2 = \begin{bmatrix} s_1^2 & & & \\ & s_2^2 & & \\ & & \ddots & \\ & & & s_D^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix} \tag{26}$$

(25) 和 (26) 奇异值和特征值之间的关系如图 13 所示。

本书后续要讨论数据矩阵分解奇异值，和协方差矩阵特征值分解得到的特征值之间的关系，请大家特别关注。

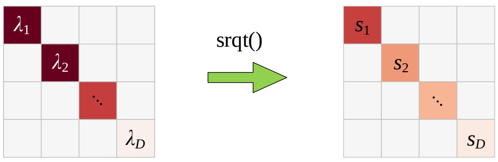


图 13. 奇异值和特征值之间关系

向量空间

同样，计算格拉姆矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$，我们知道它是对称矩阵，可以进行特征值分解。

如图 9 所示，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 进行特征值分解得到正交矩阵 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$，它也是个规范正交基，张起的空间为 $\mathbb{R}^D$。

奇异值分解强大之处在于，它可以分解一切实数矩阵，而且一次性获得 $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]$ 和 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$ 两个规范正交基。

大家可能好奇，$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 这两个规范正交基有怎么样的联系和应用？这个问题留给本书最后三章回答。

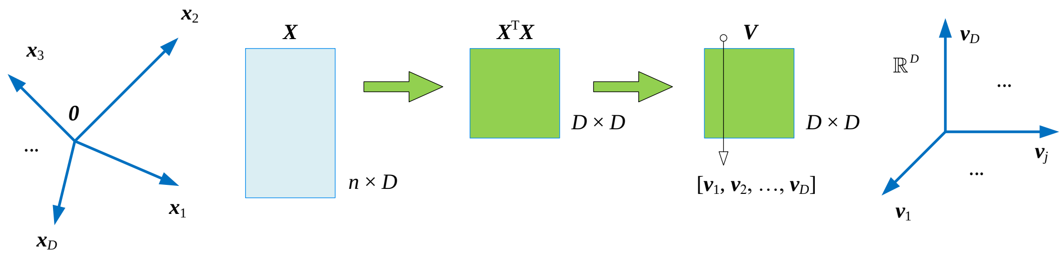


图 14. 对 Gram 矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 特征值分解得到规范正交基 $\boldsymbol{V}$

Bk4_Ch15_02.py 中 Bk4_Ch15_02_C 部分绘制图 10。请读者自行编写代码绘制图 11 和图 12。

```
# Bk4_Ch15_02_C

#%% V*V.T = I

all_max = 6
all_min = -6

fig, axs = plt.subplots(1, 3, figsize=(12, 3))

plt.sca(axs[0])
ax = sns.heatmap(V,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
ax.set_aspect("equal")
plt.title('$V$')

plt.sca(axs[1])
ax = sns.heatmap(V.T,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
ax.set_aspect("equal")
plt.title('$V^T$')

plt.sca(axs[2])
ax = sns.heatmap(V@V.T,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
ax.set_aspect("equal")
plt.title('$I$')
```

15.5 两个视角：投影和数据叠加

本节用两个数据视角观察 SVD 分解。

投影

将 (1) 等式左右两侧右乘 $\boldsymbol{V}$，可以得到：

$$\boldsymbol{X}_{n\times D}\boldsymbol{V}=\boldsymbol{U}\boldsymbol{S} \tag{27}$$

将 $\boldsymbol{V}$ 和 $\boldsymbol{U}$ 本身分别写成左右排列的列向量：

$$\boldsymbol{X}_{n\times D}\begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_D\end{bmatrix}=\begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_D\end{bmatrix}\begin{bmatrix}s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D\end{bmatrix} \tag{28}$$

(28) 进一步展开得到：

$$\begin{bmatrix}\boldsymbol{X}\boldsymbol{v}_1 & \boldsymbol{X}\boldsymbol{v}_2 & \cdots & \boldsymbol{X}\boldsymbol{v}_D\end{bmatrix}=\begin{bmatrix}s_1\boldsymbol{u}_1 & s_2\boldsymbol{u}_2 & \cdots & s_D\boldsymbol{u}_D\end{bmatrix} \tag{29}$$

因此，

$$\boldsymbol{X}\boldsymbol{v}_j=s_j\boldsymbol{u}_j \tag{30}$$

上式可以理解为 $\boldsymbol{X}$ 向 $\boldsymbol{v}_j$ 投影，结果为 $s_j\boldsymbol{u}_j$。对应运算热图如图 15 所示。

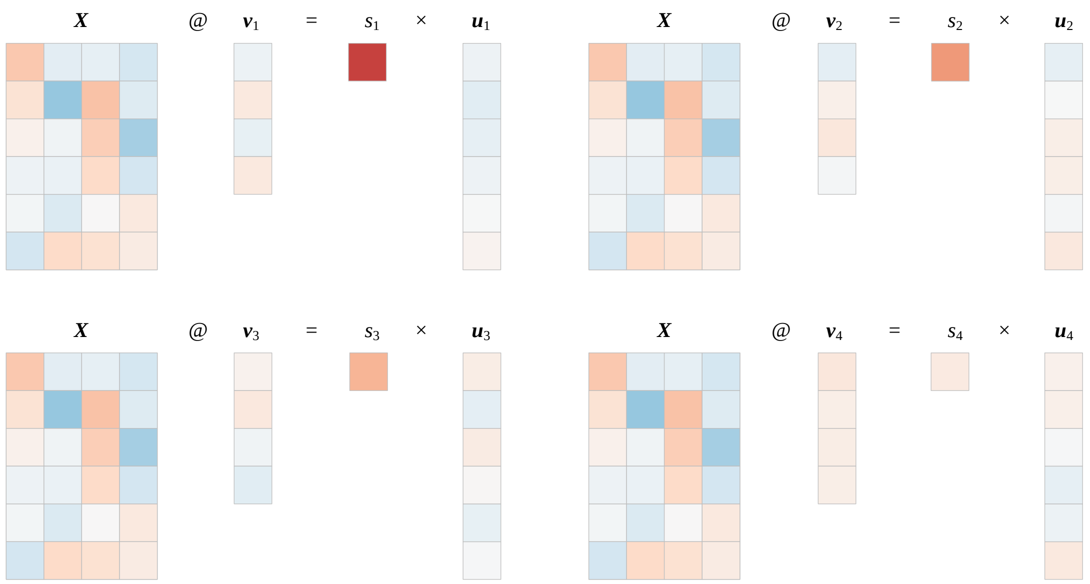


图 15. $\boldsymbol{X}$ 向 $\boldsymbol{v}_j$ 映射结果为 $s_j\boldsymbol{u}_j$

(30) 左右都是向量，等式两侧求模，即 L^2 范数，得到：

$$\left\|\boldsymbol{X}\boldsymbol{v}_j\right\|=\left\|s_j\boldsymbol{u}_j\right\|=s_j \tag{31}$$

也就是说 $\boldsymbol{X}\boldsymbol{v}_j$ 的模为对应奇异值 s_j。由于奇异值 s_1 到 s_4 从大到小排列，也就是说 $\boldsymbol{X}\boldsymbol{v}_1$ 的模最大。这个角度对于理解**主成分分析** (principal component analysis, PCA) 极为重要。

叠加

第二种展开方式如下：

$$
\begin{aligned}
\boldsymbol{X}_{n\times D} &= \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_D \end{bmatrix} \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^{\mathrm{T}} \\ \boldsymbol{v}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_D^{\mathrm{T}} \end{bmatrix} \\
&= \begin{bmatrix} s_1\boldsymbol{u}_1 & s_2\boldsymbol{u}_2 & \cdots & s_D\boldsymbol{u}_D \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^{\mathrm{T}} \\ \boldsymbol{v}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_D^{\mathrm{T}} \end{bmatrix} = s_1\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}} + s_2\boldsymbol{u}_2\boldsymbol{v}_2^{\mathrm{T}} + \cdots + s_D\boldsymbol{u}_D\boldsymbol{v}_D^{\mathrm{T}}
\end{aligned}
\tag{32}
$$

举个例子，对于 $D = 4$ 时：

$$
\boldsymbol{X} = s_1\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}} + s_2\boldsymbol{u}_2\boldsymbol{v}_2^{\mathrm{T}} + s_3\boldsymbol{u}_3\boldsymbol{v}_3^{\mathrm{T}} + s_4\boldsymbol{u}_4\boldsymbol{v}_4^{\mathrm{T}}
\tag{33}
$$

(33) 中奇异值 s_1 到 s_4 从大到小排列，即 $s_1 \geqslant s_2 \geqslant s_3 \geqslant s_4$。

注意，$s_j\boldsymbol{u}_j\boldsymbol{v}_j^{\mathrm{T}}$ 的秩为 1。

观察图 16 左侧四副热图由上到下颜色深浅变化，可以发现对应 (33) 等式右侧从左到右的四项相当于逐步还原 $\boldsymbol{X}$。下一章，我们将深入介绍这一视角。

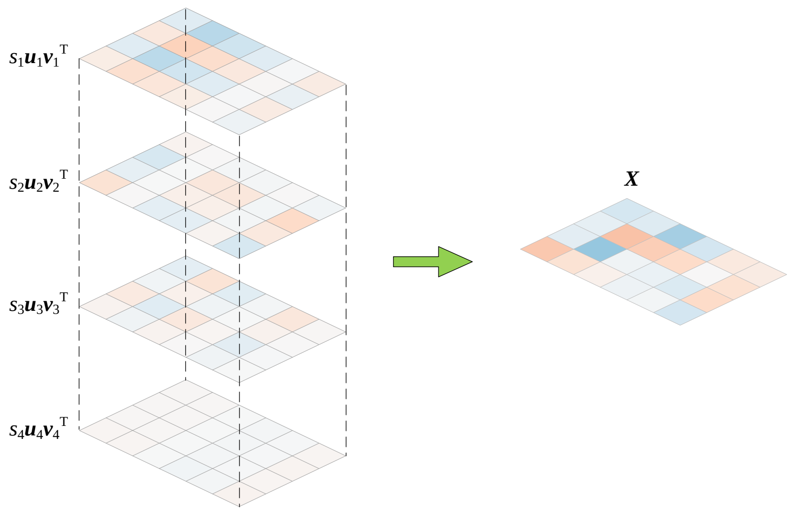


图 16. 四幅热图叠加还原原始图像

相信大家看到这里早已发现，本节前两个视角实际上就是矩阵乘法的不同视角。

张量积

再进一步，利用 (30) 给出的关系，我们将 (33) 写成张量积之和的形式

$$\begin{aligned}\boldsymbol{X} &= s_1\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}} + s_2\boldsymbol{u}_2\boldsymbol{v}_2^{\mathrm{T}} + s_3\boldsymbol{u}_3\boldsymbol{v}_3^{\mathrm{T}} + s_4\boldsymbol{u}_4\boldsymbol{v}_4^{\mathrm{T}} \\ &= \boldsymbol{X}\boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{T}} + \boldsymbol{X}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{T}} + \boldsymbol{X}\boldsymbol{v}_3\boldsymbol{v}_3^{\mathrm{T}} + \boldsymbol{X}\boldsymbol{v}_4\boldsymbol{v}_4^{\mathrm{T}} \\ &= \boldsymbol{X}\left(\boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{T}} + \boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{T}} + \boldsymbol{v}_3\boldsymbol{v}_3^{\mathrm{T}} + \boldsymbol{v}_4\boldsymbol{v}_4^{\mathrm{T}}\right) \\ &= \boldsymbol{X}\left(\boldsymbol{v}_1 \otimes \boldsymbol{v}_1 + \boldsymbol{v}_2 \otimes \boldsymbol{v}_2 + \boldsymbol{v}_3 \otimes \boldsymbol{v}_3 + \boldsymbol{v}_4 \otimes \boldsymbol{v}_4\right)\end{aligned} \tag{34}$$

这就是本书之前讲解的思路——数据在不同向量方向上投影再叠加。

能完成类似 (34) 投影的规范正交基有无数组，为什么 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$ 脱颖而出？$\boldsymbol{V}$ 的特殊性体现在哪？

回答这个问题需要优化方面的知识，这是本书后续要回答的问题。

Bk4_Ch15_02.py 中 Bk4_Ch15_02_D 部分绘制本节图像。

```
# Bk4_Ch15_02_D

#%% analysis of singular value matrix

fig, axs = plt.subplots(1, 4, figsize=(12, 3))

for j in [0, 1, 2, 3]:
    X_j = S[j,j]*U[:,j][:, None]@V[:,j][None, :];
    plt.sca(axs[j])
    ax = sns.heatmap(X_j,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                     cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    title_txt = '$s_'+ str(j+1) + 'u_'+ str(j+1) + 'v_'+ str(j+1) + '^T$'
    plt.title(title_txt)

#%% projection

for j in [0, 1, 2, 3]:

    fig, axs = plt.subplots(1, 7, figsize=(12, 3))

    v_j = V[:,j]
    v_j = np.matrix(v_j).T
    s_j = S[j,j]
    s_j = np.matrix(s_j)
    u_j = U[:,j]
    u_j = np.matrix(u_j).T

    plt.sca(axs[0])
    ax = sns.heatmap(X,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                     cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title('X')

    plt.sca(axs[1])
    plt.title('@')
    plt.axis('off')

    plt.sca(axs[2])
    ax = sns.heatmap(v_j,cmap='RdBu_r',vmax = all_max,vmin = all_min,
```

```
                 cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title('v_'+ str(j+1))

    plt.sca(axs[3])
    plt.title('=')
    plt.axis('off')

    plt.sca(axs[4])
    ax = sns.heatmap(s_j,cmap='RdBu_r',vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title('s_'+ str(j+1))

    plt.sca(axs[5])
    plt.title('@')
    plt.axis('off')

    plt.sca(axs[6])
    ax = sns.heatmap(u_j,cmap='RdBu_r', vmax = all_max,vmin = all_min,
                 cbar_kws={"orientation": "horizontal"})
    ax.set_aspect("equal")
    plt.title('u_'+ str(j+1))
```

下图四副子图总结本章主要内容。请大家奇异值分解对应“旋转 → 缩放 → 旋转”，不同于特征值分解的“旋转 → 缩放 → 旋转”。

任何实数矩阵都可以进行奇异值分解，但是只有可对角矩阵才能进行特征值分解。此外，奇异值分解得到的两个正交矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 一般形状不同。

Gram 矩阵是奇异值分解和特征值分解的桥梁，这一点本书后面还要提到。

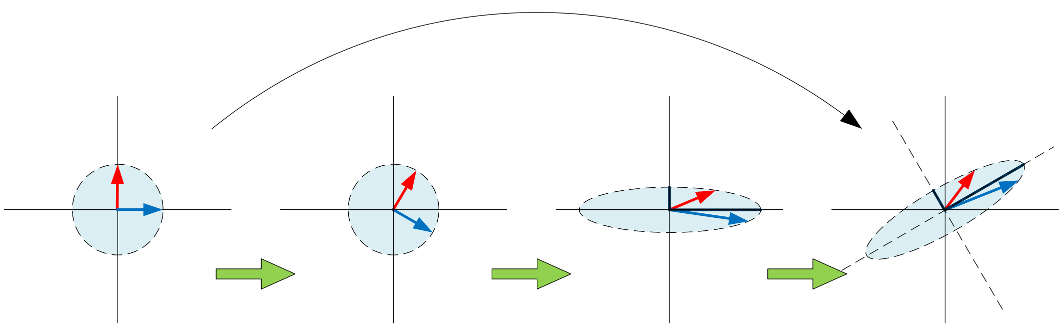


图 17. 总结本章重要内容的四副图