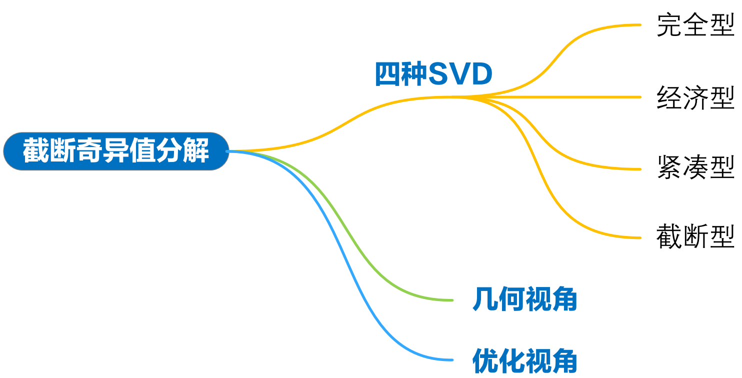
15 Truncated Singular Value Decomposition 截断奇异值分解

用 SVD 完成主成分分析

> 给我一个立足之地，一个足够长的杠杆，我将撬动世界。
>
> ***Give me a place to stand, and a lever long enough, and I will move the world.***
>
> —— 阿基米德 (Archimedes) | 古希腊数学家、物理学家 | 287 BC ~ 212 BC

- `seaborn.heatmap()` 绘制数据热图
- `numpy.linalg.eig()` 特征值分解
- `numpy.linalg.svd()` 奇异值分解
- `sklearn.decomposition.TruncatedSVD()` 截断 SVD 分解



15.1 几何视角看奇异值分解

奇异值分解 (Singular Value Decomposition, SVD) 是机器学习重要的数学利器；因此，鸢尾花书从《编程不难》开始就从各个角度展示奇异值分解。

比如，《可视之美》介绍过 4 种不同形状矩阵 (2×2 方阵、3×3 方阵、3×2 细高矩阵、2×3 矮胖矩阵) SVD 分解结果对应的几何变换。下面，我们简单回顾图 1 所示 3×2 细高矩阵的完全型 SVD 分解的几何视角。

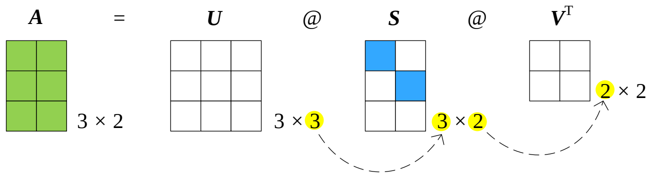


图 1. 细高型矩阵的完全型 SVD 分解

矩阵 $\boldsymbol{A}$ 的完全型 SVD 分解结果为。

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}} \tag{1}$$

其中，$\boldsymbol{S}$ 为对角阵，其主对角线元素 s_j ($j = 1, 2, \dots, D$) 为**奇异值** (singular value)。

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

$\boldsymbol{V}$ 的列向量称作**右奇异向量** (right singular vector)。

SVD 分解有四种主要形式，完全型是其中一种。

在完全型 SVD 分解中，$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 为正交矩阵，即 $\boldsymbol{U} @ \boldsymbol{U}^{\mathrm{T}} = \boldsymbol{I}$ 且 $\boldsymbol{V} @ \boldsymbol{V}^{\mathrm{T}} = \boldsymbol{I}$。

举个例子，对形状为 3×2 的矩阵 $\boldsymbol{A}$ 进行 SVD 分解

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1/\sqrt{6} & \sqrt{2}/2 & \sqrt{3}/3 \\ 2/\sqrt{6} & 0 & -\sqrt{3}/3 \\ 1/\sqrt{6} & -\sqrt{2}/2 & \sqrt{3}/3 \end{bmatrix}}_{\boldsymbol{U}} @ \underbrace{\begin{bmatrix} \sqrt{3} & \\ & 1 \\ & \end{bmatrix}}_{\boldsymbol{S}} @ \underbrace{\begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}}_{\boldsymbol{V}^{\mathrm{T}}} \tag{2}$$

从几何角度来看，图 2 中 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ 完成的几何操作可以写成 $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{y}$。

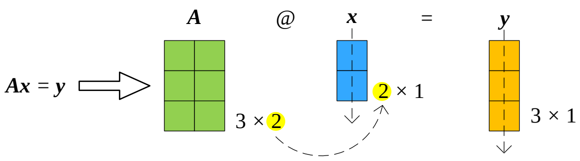


图 2．列向量 $\boldsymbol{x}$ 在细高矩阵 $\boldsymbol{A}$ 映射下结果为列向量 $\boldsymbol{y}$

也就是说，矩阵 $\boldsymbol{A}$ 完成的几何变换可以拆解为三步——旋转 ($\boldsymbol{V}^{\mathrm{T}}$) → 缩放 ($\boldsymbol{S}$) →旋转 ($\boldsymbol{U}$)。

如图 3 所示，$\boldsymbol{V}^{\mathrm{T}}$ 的旋转发生在 $\mathbb{R}^2$，$\boldsymbol{U}$ 的旋转则发生在 $\mathbb{R}^3$。缩放 ($\boldsymbol{S}$) 虽然将数据“升维”，但是结果还是在三维空间的一个 (过原点) 斜面上。

有关图 3 介绍的可视化方案，请大家参考《可视之美》；有关奇异值的数学原理请大家参考《矩阵力量》。

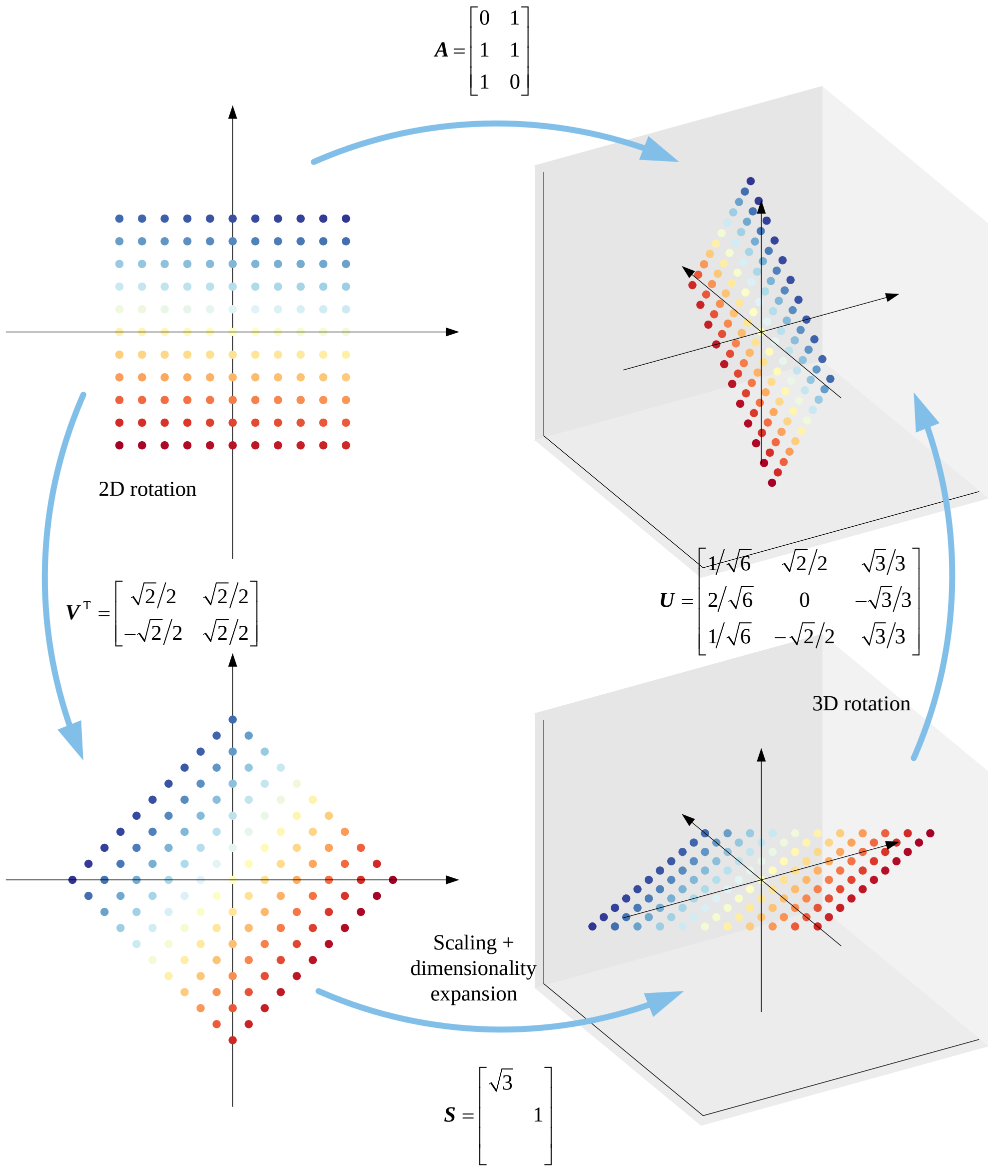


图 3. 完全型 SVD 分解的几何视角

15.2 四种 SVD 分解

《矩阵力量》第 16 章介绍了四种奇异值分解——**完全型** (full)、**经济型** (economy-size, thin)、**紧凑型** (compact)、**截断型** (truncated)。图 4 ~ 图 7 展示了它们之间的关系。图 7 中截断型 SVD 分解就是本章用于主成分分析的数学工具，并注意图中的约等号。请大家格外注意紧缩型 SVD 分解存在的前提。

sklearn.decomposition.TruncatedSVD() 这个函数就是用截断型 SVD 分解完成 PCA。

请大家参考矩阵力量第 16 章将推导过程写到对应图像上。

请大家顺便回顾《矩阵力量》第 6 章有关分块矩阵乘法相关内容。

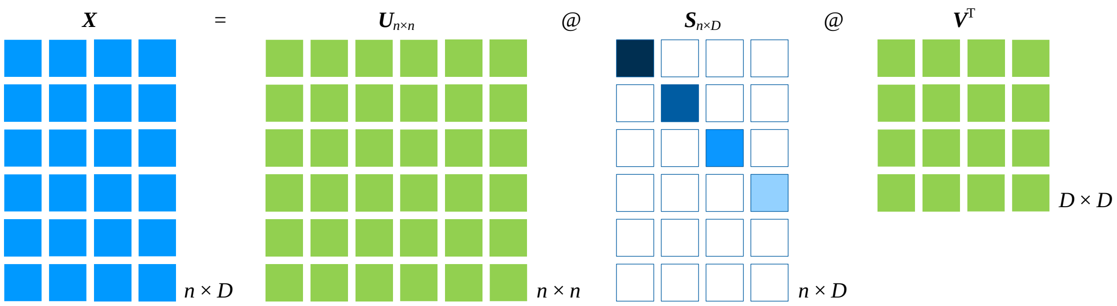


图 4. 完全型 SVD 分解

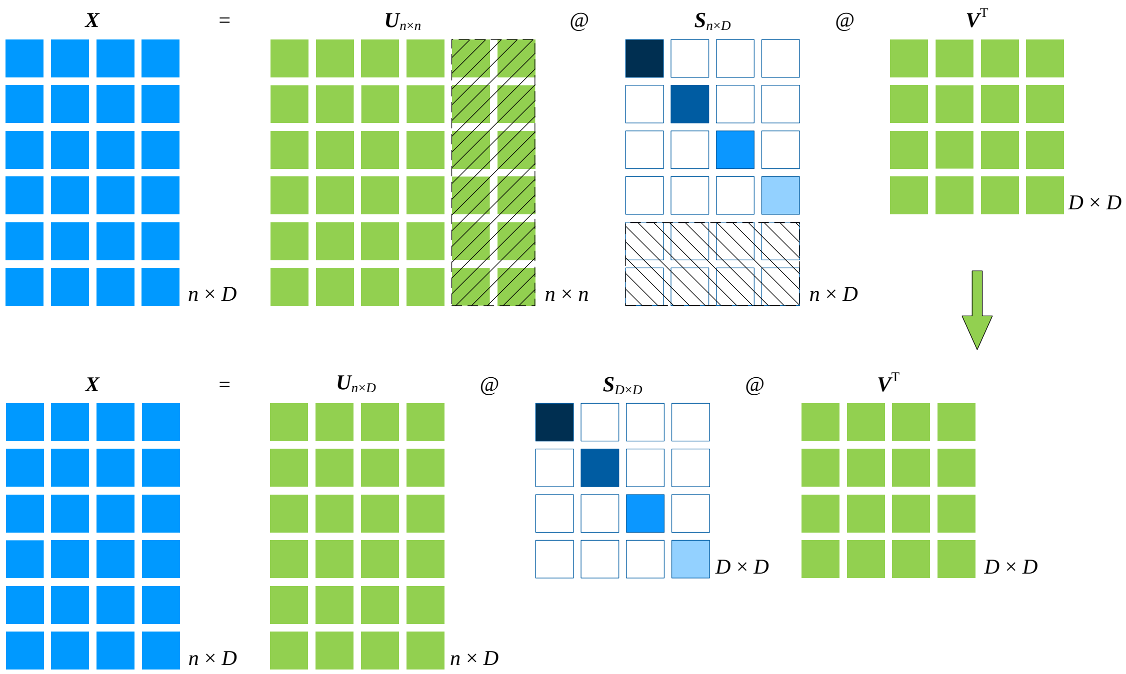


图 5. 从完全型到经济型

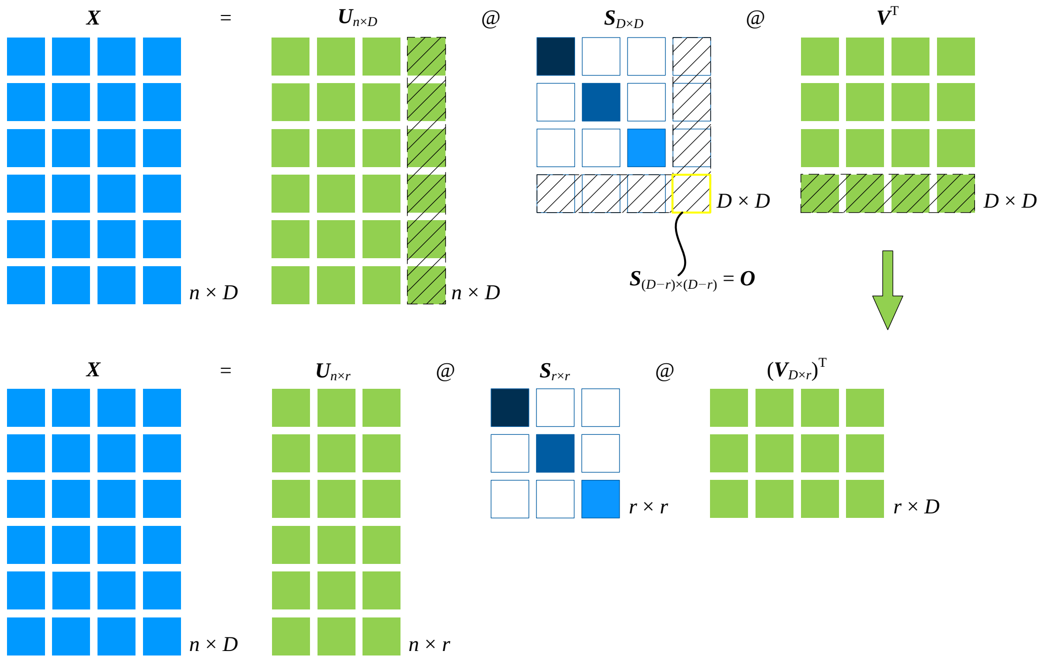


图 6. 从经济型到紧缩型

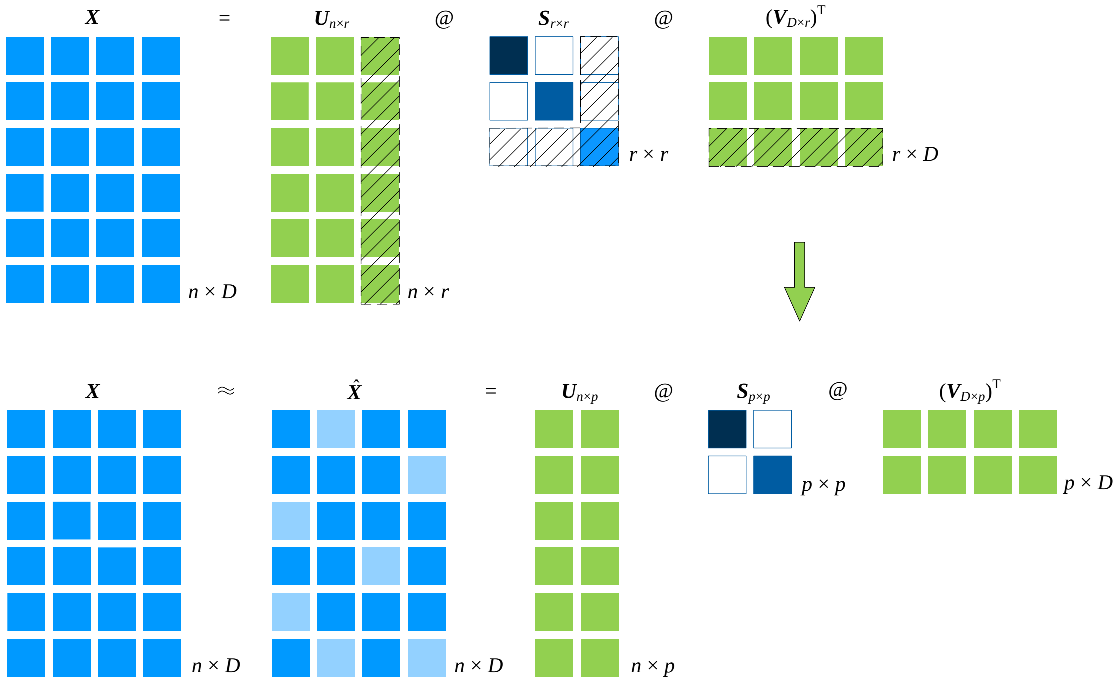


图 7. 从紧缩型到截断型

15.3 几何视角看截断型 SVD

如图 5 下图所示，对于形状为 $n \times D$ 原始数据矩阵 $\boldsymbol{X}$，其经济型 SVD 分解为。

$$X_{n\times D} = U_{n\times D} S_{D\times D} V_{D\times D}^{\mathrm{T}} \tag{3}$$

其中，U 和 X 的形状相同，U 的列向量为单位向量且两两正交；V 还是 $D \times D$ 方阵，V 的列向量也是单位向量且两两正交。

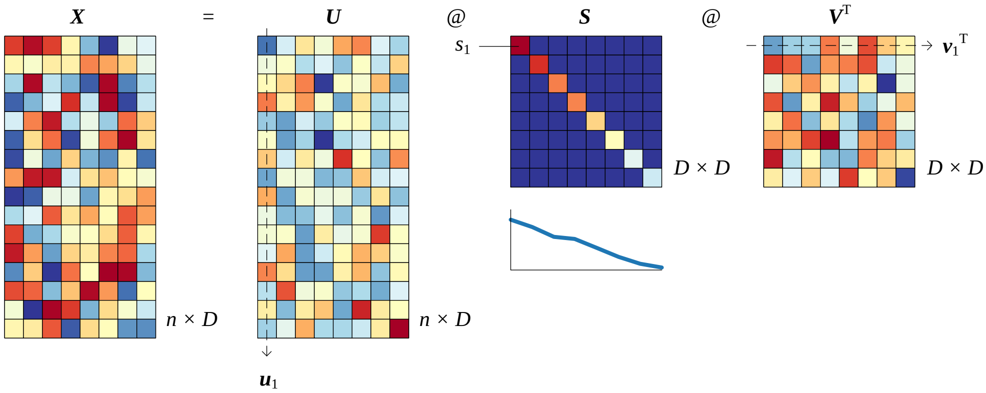


图 8. 原始数据经济型 SVD 分解

矩阵乘法第二视角

《矩阵力量》介绍过理解矩阵乘法的两个视角。根据矩阵乘法第二视角，原始数据矩阵 X 的经济型 SVD 分解可以展开写成 D 个矩阵相加。

$$X_{n\times D} = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_D \end{bmatrix}}_{U_{n\times D}} \underbrace{\begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D \end{bmatrix}}_{S_{D\times D}} \underbrace{\begin{bmatrix} v_1^{\mathrm{T}} \\ v_2^{\mathrm{T}} \\ \vdots \\ v_D^{\mathrm{T}} \end{bmatrix}}_{V_{D\times D}} = s_1 u_1 v_1^{\mathrm{T}} + s_2 u_2 v_2^{\mathrm{T}} + \cdots + s_D u_D v_D^{\mathrm{T}} = \sum_{j=1}^{D} s_j u_j v_j^{\mathrm{T}} \tag{4}$$

由于 u_j 和 v_j 都是单位向量，即 L^2 模都为 1；它们之间只存在方向分别，不存在大小的分别。因此奇异值 s_j 的大小体现出主成分的重要性。上一章介绍的陡坡图可以用特征值来算，也可以用奇异值。

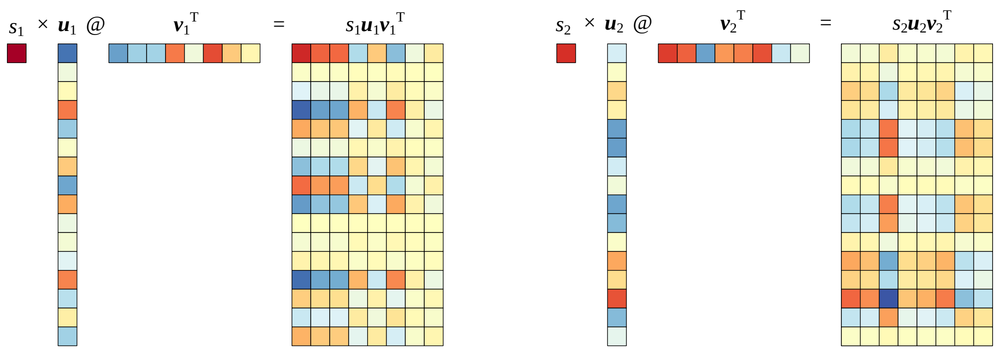


图 9. 前两个主成分还原部分原始数据

如果奇异值 s_1、s_2 … s_D 由小到大排列，v_1 就是第一主成分载荷，u_1 就是第一主成分因子得分。

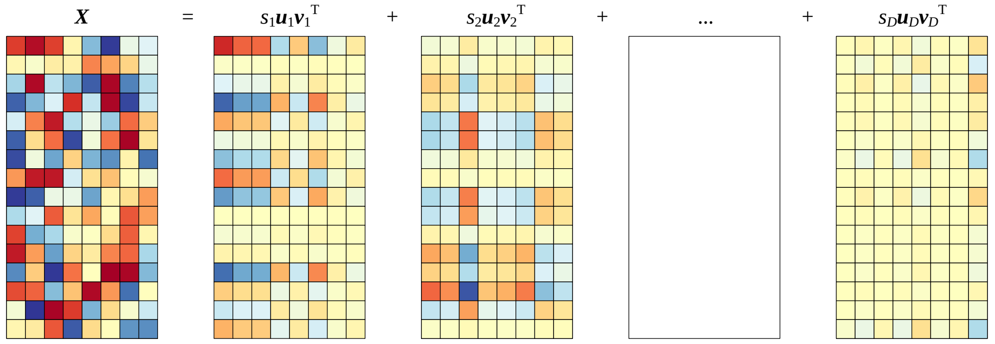


图 10. 原始数据相当于由 D 个形状相同矩阵求和的结果

如图 11 所示，用前 p 个主成分还原原始数据

$$\boldsymbol{X}_{n\times D} \approx \hat{\boldsymbol{X}}_{n\times D} = \boldsymbol{U}_{n\times p} \boldsymbol{S}_{p\times p} \left(\boldsymbol{V}_{D\times p}\right)^{\mathrm{T}} = \sum_{j=1}^{p} s_j \boldsymbol{u}_j \boldsymbol{v}_j^{\mathrm{T}} \tag{5}$$

注意，如果假设前 p 个奇异值 s_j 均大于 0。这样，矩阵 $\hat{\boldsymbol{X}}_{n\times D}$ 的秩为 p。也就是说，在 s_j 均大于 0 的前提下，上式中每叠加一层 $s_j \boldsymbol{u}_j \boldsymbol{v}_j^{\mathrm{T}}$，$\hat{\boldsymbol{X}}_{n\times D}$ 的秩就增大 1。

举个例子，$\hat{\boldsymbol{X}}_{n\times D} = s_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}}$ 的秩为 1；$\hat{\boldsymbol{X}}_{n\times D} = s_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} + s_2 \boldsymbol{u}_2 \boldsymbol{v}_2^{\mathrm{T}}$ 的秩为 2。

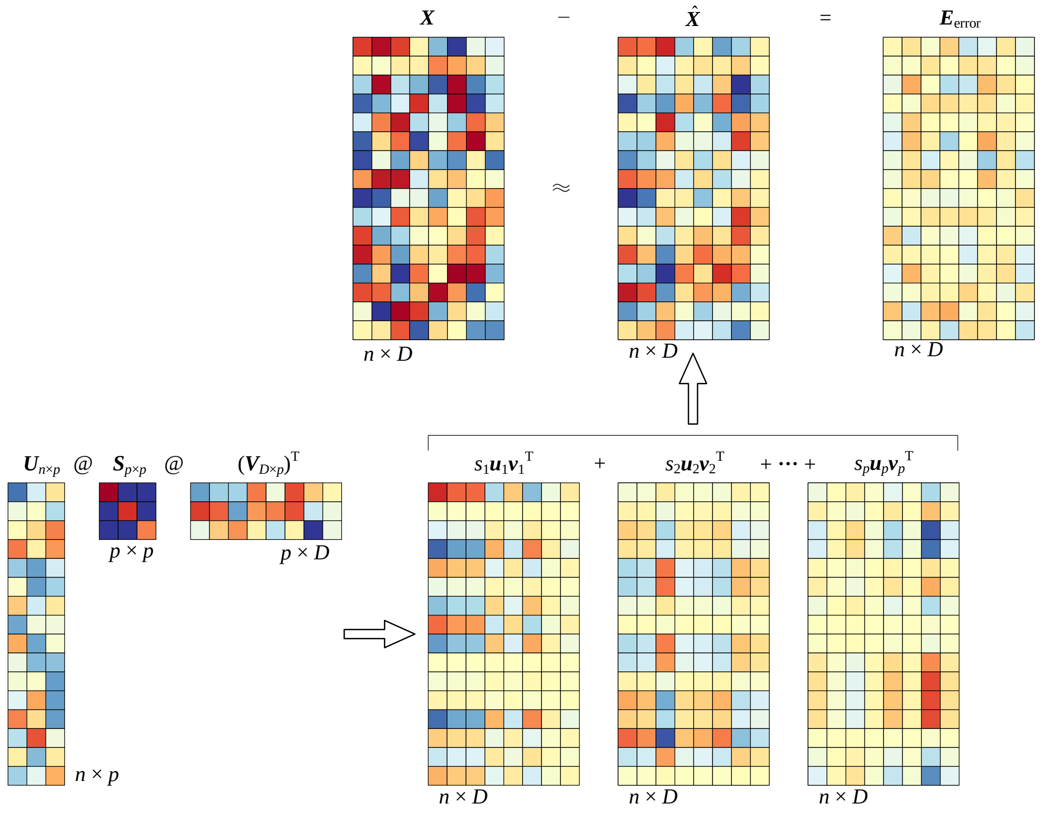


图 11. 用前 p 个主成分还原原始数据

投影视角

下面，我们再从投影视角理解截断型 SVD 分解。

将 (3) 写成

$$X_{n\times D}V_{D\times D}=U_{n\times D}S_{D\times D} \tag{6}$$

上式相当于将 $\boldsymbol{X}$ 投影到 $\boldsymbol{V}$ 空间中。

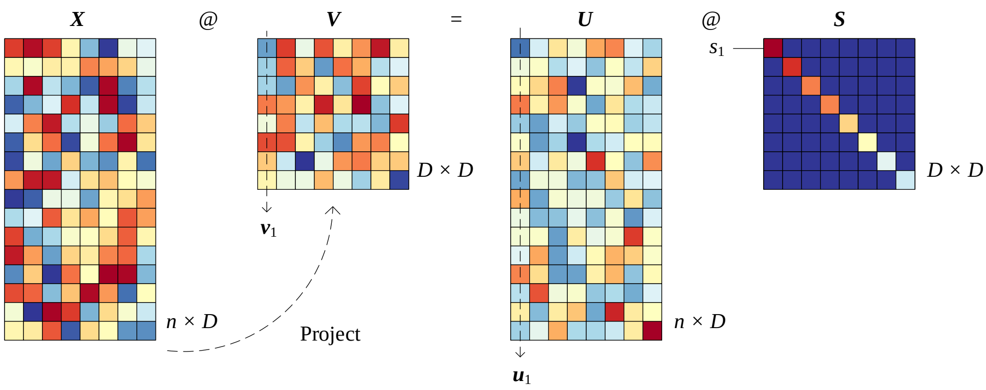


图 12. 原始数据向 $\boldsymbol{V}$ 投影

也用矩阵乘法第二视角，将 (6) 写成

$$X_{n\times D}\underbrace{\begin{bmatrix}v_1 & v_2 & \cdots & v_D\end{bmatrix}}_{V_{D\times D}}=\underbrace{\begin{bmatrix}u_1 & u_2 & \cdots & u_D\end{bmatrix}}_{U_{n\times D}}\begin{bmatrix}s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D\end{bmatrix} \tag{7}$$

进一步展开得到

$$\begin{bmatrix}Xv_1 & Xv_2 & \cdots & Xv_D\end{bmatrix}=\begin{bmatrix}s_1u_1 & s_2u_2 & \cdots & s_Du_D\end{bmatrix} \tag{8}$$

从几何角度，$\boldsymbol{X}$ 朝 $\boldsymbol{v}_j$ 投影结果为 $s_j\boldsymbol{u}_j$ 。

$$Xv_j=s_ju_j \tag{9}$$

图 13 所示为原始数据朝 $\boldsymbol{v}_1$ 投影结果为 $s_1\boldsymbol{u}_1$ 。由于 $\|\boldsymbol{u}_j\|=1$ ，$\|\boldsymbol{X}\boldsymbol{v}_j\|=s_j$ 。

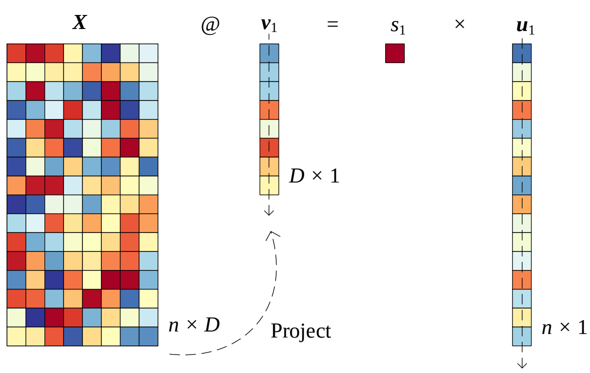


图 13. 原始数据向 $\boldsymbol{v}_1$ 投影

那么问题来了，如何找到 $\boldsymbol{v}_1$？这就需要构造优化问题。

15.4 优化视角看截断型 SVD

有了上面的铺垫，我们便可以讨论奇异值分解中的优化问题。

最大 L2 范数

如图 14 所示，我们要在 $\mathbb{R}^D$ 中找到一个单位向量 $\boldsymbol{v}$ 让投影结果 $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{v}$ 的 L^2 范数最大，即。

$$\begin{aligned} &\underset{\boldsymbol{v}}{\arg\max} \left\| \boldsymbol{X}\boldsymbol{v} \right\| \\ &\text{subject to: } \left\| \boldsymbol{v} \right\| = 1 \end{aligned} \tag{10}$$

而上式的最大值为奇异值 s_1。

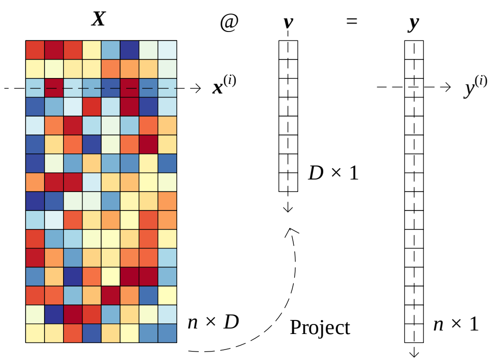


图 14. 原始数据向 $\boldsymbol{v}$ 投影

如图 15 所示，列向量 $\boldsymbol{y}$ 的元素 $y^{(i)}$ 就是在 $\boldsymbol{v}$ 方向上 $y^{(i)}$ 到原点的距离。

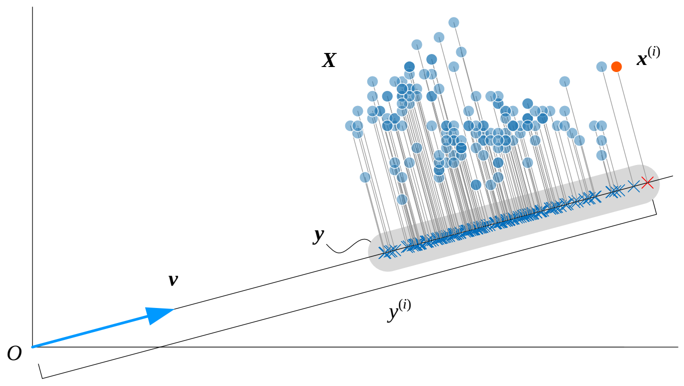


图 15. 几何角度来看原始数据向 $\boldsymbol{v}$ 投影

格拉姆矩阵最大特征值

将 $\|\boldsymbol{X}\boldsymbol{v}\|$ 平方后，(10) 等价于。

$$\begin{aligned}&\underset{\boldsymbol{v}}{\arg\max}\ \|\boldsymbol{X}\boldsymbol{v}\|_2^2\\&\text{subject to: } \|\boldsymbol{v}\|=1\end{aligned} \tag{11}$$

这便是最大化图 15 中投影结果平方和。

(11) 相当于找到格拉姆矩阵 $\boldsymbol{G} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的最大特征值 λ_1，即

$$\begin{aligned}&\underset{\boldsymbol{v}}{\arg\max}\ \boldsymbol{v}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{v}\\&\text{subject to: } \|\boldsymbol{v}\|=1\end{aligned} \tag{12}$$

显然，最大值特征值和最大奇异值之间的关系为 $\lambda_1 = s_1^2$。请大家回顾如何用拉格朗日乘子法求解上述优化问题。

瑞利商

《矩阵力量》第 18 章介绍过，(12) 等价于

$$\underset{\boldsymbol{x}}{\arg\max}\ \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} \tag{13}$$

其中，$\boldsymbol{x}$ 不为零向量。上式就是求解瑞利商的最大值。图 16 所示为理解瑞利商的两个视角，请大家自行回顾《可视之美》相关内容。

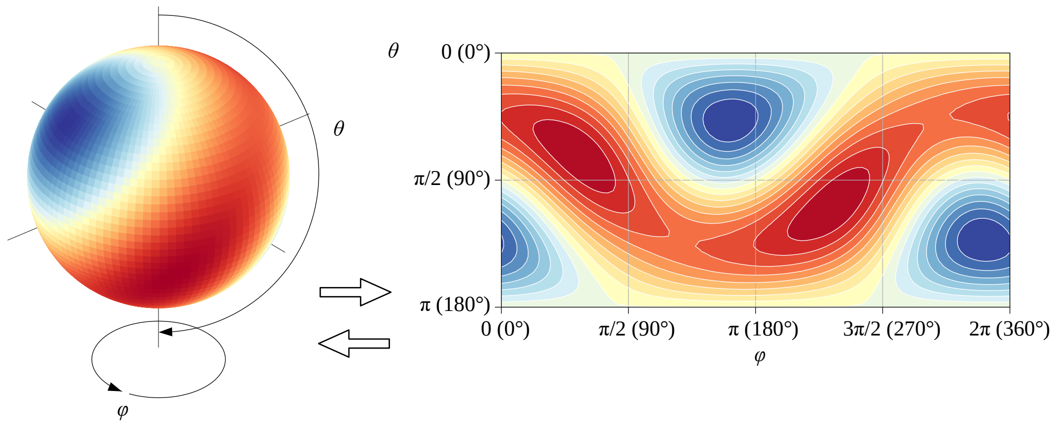


图 16．两个理解瑞利商的视角

矩阵 F-范数

《矩阵力量》第 18 章专门介绍过矩阵的**弗罗贝尼乌斯范数** (Frobenius norm)，简称 F-范数：

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{i,j}\right|^2} \tag{14}$$

也就是说，矩阵 $\boldsymbol{A}$ 的 F-范数就是矩阵所有元素的平方和，再开方。

由于矩阵 $\boldsymbol{A}$ 的所有元素平方和就是 $\boldsymbol{A}$ 的格拉姆矩阵 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$ 的迹，即：

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{i,j}|^2} = \sqrt{\mathrm{tr}\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)} \tag{15}$$

而上述结果还可以写成：

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{\min(m,n)} \lambda_i} \tag{16}$$

其中，$\sqrt{\sum_{i=1}^{\min(m,n)} \lambda_i}$ 为格拉姆矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的特征值之和。

由于，格拉姆矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的特征值和 $\boldsymbol{A}$ 的奇异值存在等式关系 $\lambda_i = s_i^2$，(16) 还可以写成：

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{n} \lambda_i} = \sqrt{\sum_{i=1}^{n} s_i^2} \tag{17}$$

如果矩阵 $\boldsymbol{A}$ 的奇异值分解为 $\boldsymbol{A} = \boldsymbol{USV}$，这样 $\boldsymbol{A}$ 的 F-范数还可以写成

$$\|\boldsymbol{A}\|_F = \|\boldsymbol{S}\|_F = \sqrt{\sum_{i=1}^{n} s_i^2} \tag{18}$$

有了矩阵 F-范数，我们便多了一个理解截断奇异值分解的视角。

对于数据矩阵 $\boldsymbol{X}$，$\hat{\boldsymbol{X}}$ 是其秩不超过 p 的最优近似，则。

$$\left\|\boldsymbol{X} - \hat{\boldsymbol{X}}\right\|_F = \sqrt{\sum_{i=p+1}^{D} s_i^2} = s_{p+1}^2 + s_{p+2}^2 + \cdots s_D^2 \tag{19}$$

上式便代表降维数据相对原始数据的“信息损失”。

还是回到图 15，我们可以发现 (11) 是最大化投影结果的平方和；而上式则代表另外一个优化问题的解——最小化真实数据点和投影数据之间距离的平方和。

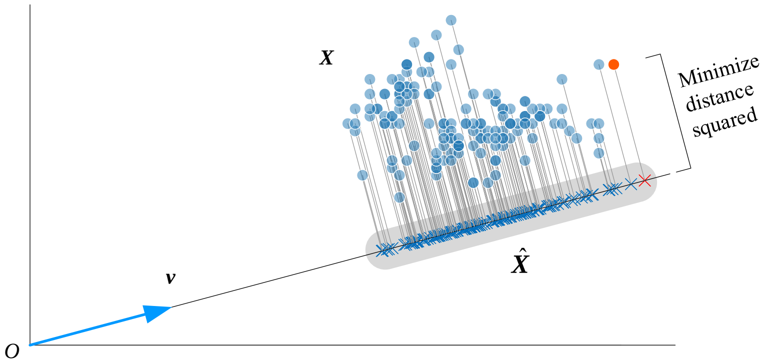


图 17. 最小化原始数据 $\boldsymbol{X}$ 和近似数据 $\hat{\boldsymbol{X}}$ 之间“距离”

数据是否中心化、标准化

如图 18 所示，当数据中心化后，其质心移动到了原点。对中心化数据进行 SVD 分解相当于对原始数据协方差矩阵的 EVD 分解。而当数据标准化后，对标准化数据进行 SVD 分解相当于对原始数据相关性系数矩阵的 EVD 分解。这是下一章要重点展开讨论的内容。

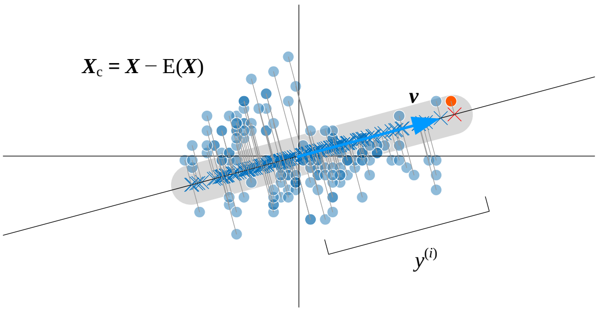


图 18. 几何角度来看中心化数据向 v 投影

15.5 分析鸢尾花照片

本节用截断奇异值分解分析鸢尾花照片。图 19 所示为作者拍的一章鸢尾花照片，经过黑白化处理后的每个像素都是 [0, 1] 范围内的数字。所以整幅图片可以看成一个数据矩阵。

《可视之美》一册专门介绍过彩色和黑白图像之间转换。

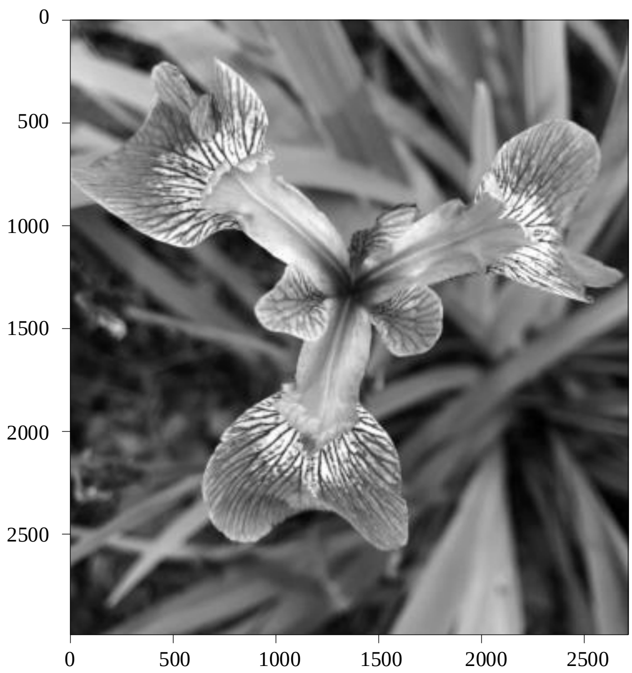


图 19. 鸢尾花图片，经过黑白处理

图 20 所示为利用 SVD 分解得到的奇异值随主成分变化。图 21 所示为特征值随主成分变化。图 22 所示为累积解释方差百分比随主成分变化。我们可以发现前 10 个主成分已经解释超过 90%的方差。

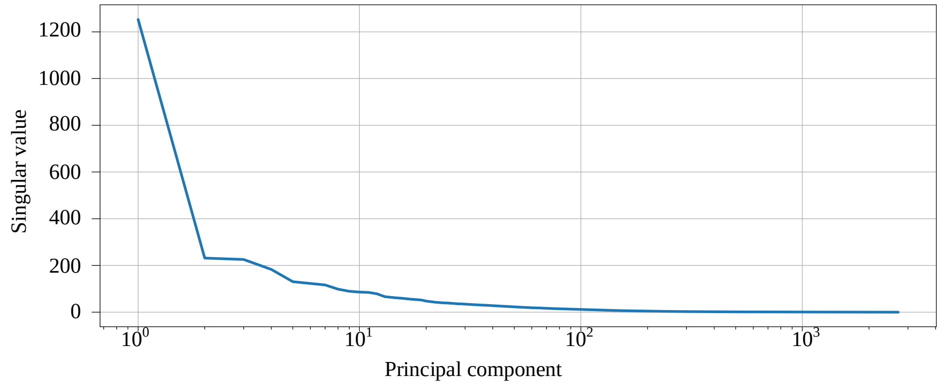


图 20. 奇异值随主成分变化

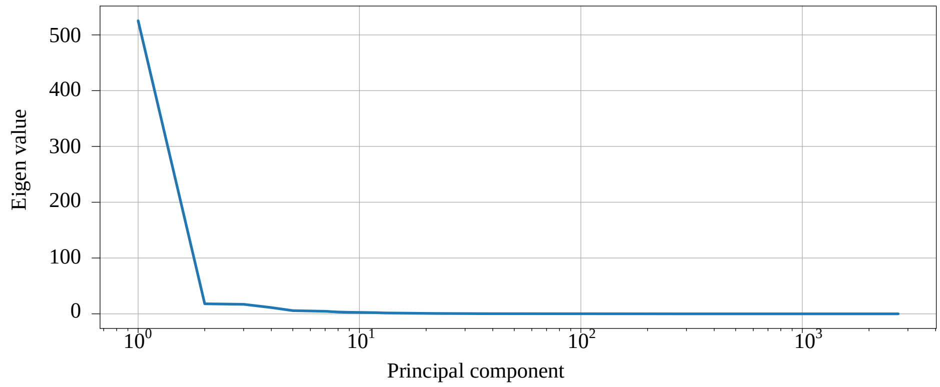


图 21. 特征值随主成分变化

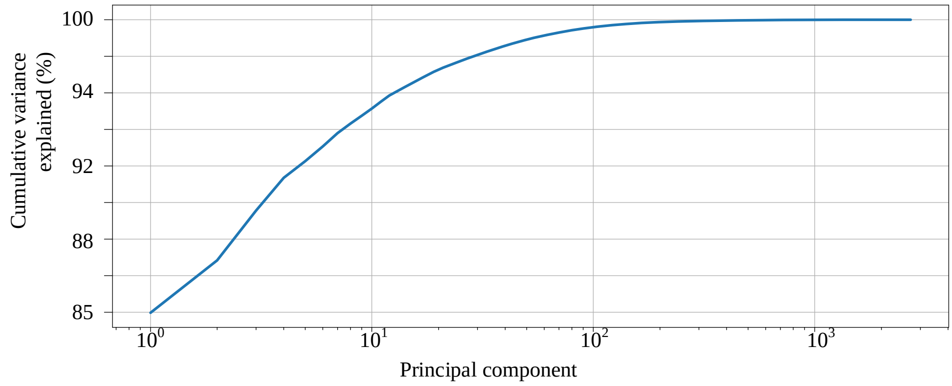


图 22. 累积解释方差百分比随主成分变化

图 23 所示为利用第 1 主元还原鸢尾花图片，左图为还原结果，右图为误差。左图中，鸢尾花还难觅踪影。图 24 所示为利用第 1、2 主元还原鸢尾花照片。如图 25 所示，这幅图相当于由 2 个秩一矩阵叠加而成。图 26 所示为利用前 4 个主元还原鸢尾花照片。这幅图相当于由 4 个秩一矩阵叠加而成，具体如图 27 所示。

在图 24 和图 26 两幅图的左图中我们仅仅能够看到“格子”。

图 28 的左图利用前 16 个主元还原照片，我们已经能够看到鸢尾花的样子，注意这幅图的秩为 16。图 29 所示为利用前 64 个主元还原鸢尾花图片，图形已经很清晰。相比原图片，图 29 的数据发生大幅压缩。

这种利用 PCA 进行图像降维方法用途很广泛。比如，在人脸识别中，**特征脸** (eigenface) 是一种基于 PCA 的特征提取方法，用于将人脸图像转换成低维特征向量进行分类或识别。特征脸是指由 PCA 分解出来的主成分图像，它们是一组基于训练数据集的线性组合，每个特征脸表示了一个数据集中的特定方向，可以看作是数据集的主要特征或重要性征。

特征脸的提取过程可以分为以下几步：1) 对人脸图像进行预处理，比如灰度化、尺度归一化、去除噪声等。2) 将预处理后的图像转换成向量形式。3) 将向量集合进行 PCA 降维，得到一组主成分向量，也就是特征脸。4） 将人脸图像向量投影到主成分向量上，得到每个人脸的特征向量表示。

特征脸在人脸识别中的作用是对人脸图像进行有效的特征提取和降维，使得原始图像数据被压缩到一个低维空间中，并且保留了原始数据中的大部分信息。通过比较人脸图像的特征向量之间的相似度，可以进行人脸识别、验证等应用。

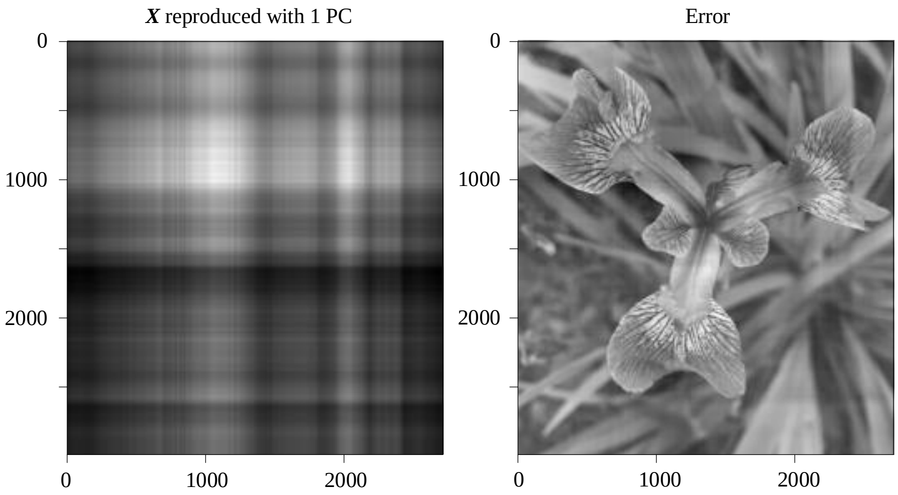


图 23. 利用第 1 主元还原鸢尾花照片

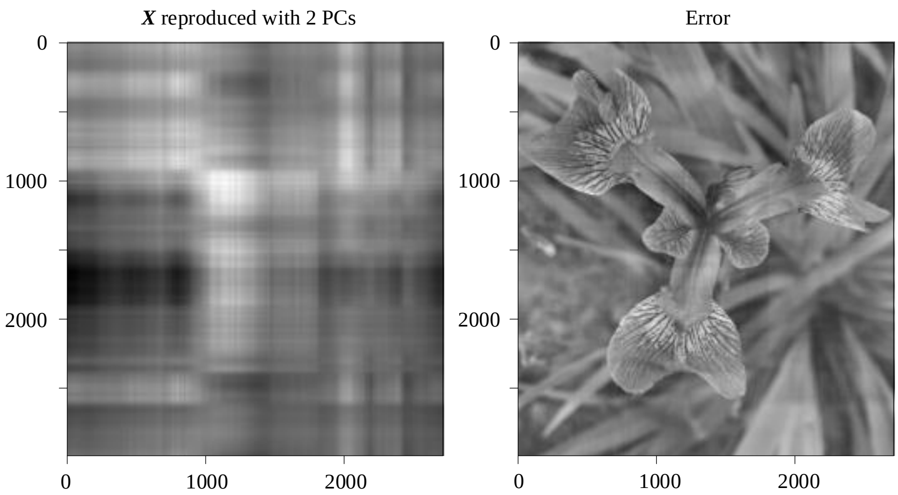


图 24. 利用第 1、2 主元还原鸢尾花照片

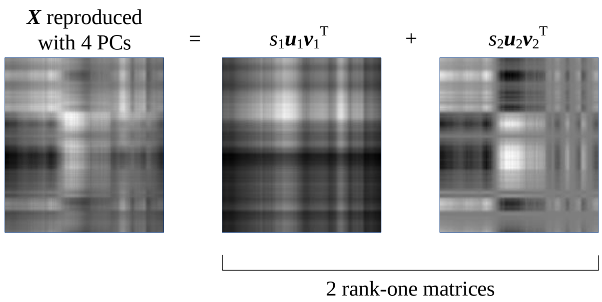


图 25. 前 2 个秩一矩阵叠加

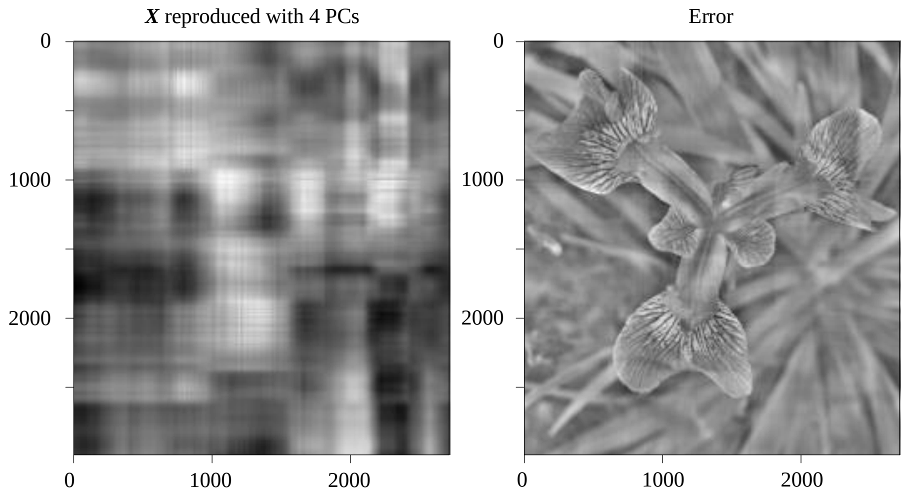


图 26. 利用第 1、2、3、4 主元还原鸢尾花照片

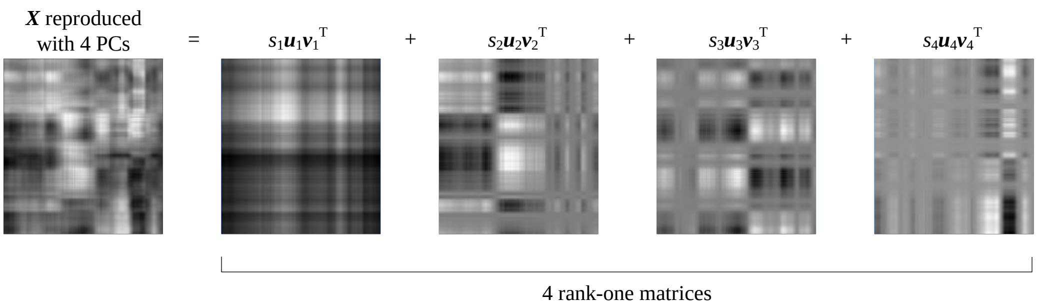


图 27. 前 4 个秩一矩阵叠加

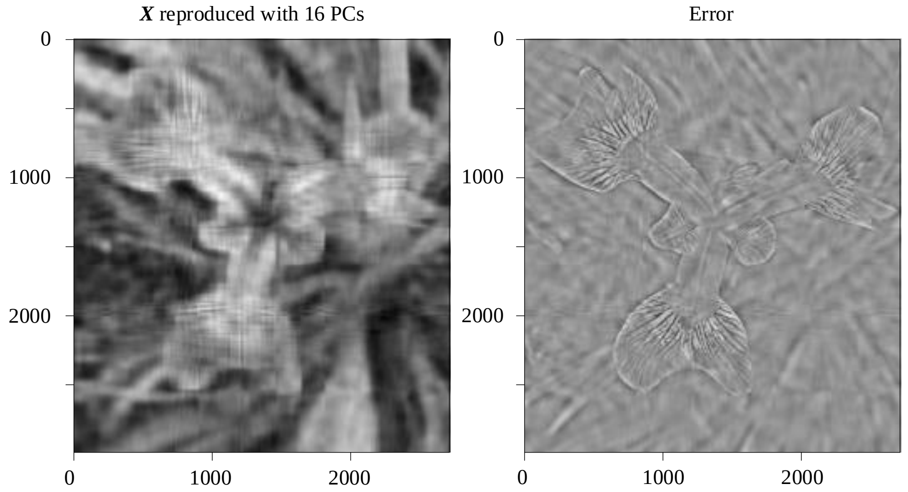


图 28. 利用前 16 个主元还原鸢尾花照片

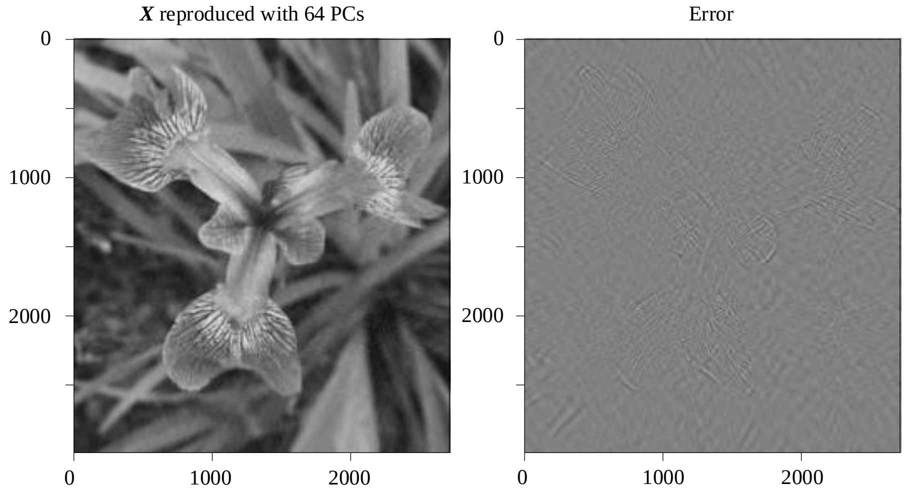


图 29. 利用前 64 个主元还原鸢尾花照片

Bk7_Ch15_01.ipynb 绘制本节图片。鸢尾花照片也在文件夹中。下面首先聊聊代码 1 中关键语句。

ⓐ用 `skimage.io.imread()`先将照片读入，然后再用 `skimage.color.rgb2gray()`将照片转化为灰度图片，数据保存在 X 中。矩阵 X 中元素的取值范围为 [0, 1]，代表灰度值。

利用 `X.shape` 大家可以知道矩阵 X 的形状为(2990, 2714)。

同时，利用 `np.linalg.matrix_rank(X)`，我们可以知道数据矩阵的秩为 2714。

ⓑ用 `matplotlib.pyplot.imshow()`可视化灰度照片，并指定颜色映射为灰度。

ⓒ利用 `numpy.linalg.svd()`对矩阵 X 进行 SVD 分解。

ⓓ可视化奇异值变化。Bk7_Ch15_01.ipynb 还给出代码展示特征值变化，请大家自行学习。

ⓔ设置横轴为对数刻度以更清晰地显示奇异值的分布。

ⓕ设定近似矩阵的秩为 16，也就是用 16 层秩一矩阵叠加还原原始数据，计算过程对应ⓖ。

ⓗ还是用 `matplotlib.pyplot.imshow()`可视化近似矩阵。

ⓘ可视化误差。

```python
from skimage import color
from skimage import io

# 读入照片，并将其转化为黑白
X = color.rgb2gray(io.imread('iris_photo.jpg'))

# 可视化照片
fig, axs = plt.subplots()

plt.imshow(X, cmap='gray')

# SVD分解
U, S, V = np.linalg.svd(X)

# 可视化奇异值
fig, ax = plt.subplots()

plt.plot(component_idx, S)
plt.grid()
ax.set_xscale('log')
plt.xlabel("Principal component")
plt.ylabel("Singular value")

rank = 16

# 近似数据
X_reconstruction = U[:, :rank] * S[:rank] @ V[:rank,:]

fig, axs = plt.subplots(1, 2)
axs[0].imshow(X_reconstruction, cmap='gray')
axs[0].set_title('X_reproduced with ' +
                 str(rank) +
                 ' PCs')

# 误差
axs[1].imshow(X - X_reconstruction, cmap='gray')
axs[1].set_title('Error')
```

代码 1. numpy.linalg.svd()完成截断奇异值分解 | Bk7_Ch15_01.ipynb

代码 2 介绍如何使用 sklearn.decomposition.TruncatedSVD() 完成截断奇异值分解。

ⓐ利用 sklearn.decomposition.TruncatedSVD() 创建截断奇异值分解实例 svd，并指定要保留的主成分数量为 16 (n_components=16)。

ⓑ使用 fit_transform 方法对输入数据 X 进行降维，得到降维后的结果 X_reduced。

ⓒ打印其形状。在这个例子中，降维后的数据矩阵的形状为(2990, 16)，即 2990 个样本，每个样本有 16 个特征。

ⓓ使用 inverse_transform 方法对降维后的数据进行反变换，得到近似原始数据 X_approx。

ⓔ打印 X_approx 形状。在这个例子中，近似的原始数据矩阵的形状为(2990, 2714)，与原始数据的形状相同。

ⓕ用 numpy.linalg.matrix_rank() 函数计算近似的原始数据矩阵 X_approx 的秩，打印结果为 16。这表明近似的原始数据矩阵中确实只包含了截断 SVD 所保留的 16 个主成分。

```python
from sklearn.decomposition import TruncatedSVD
svd = TruncatedSVD(n_components=16)

# 降维后的结果
X_reduced = svd.fit_transform(X)
print(X_reduced.shape)
# 结果为(2990, 16)

# 反变换，获取近似数据
X_approx = svd.inverse_transform(X_reduced)

print(X_approx.shape)
# 结果为(2990, 2714)

print(np.linalg.matrix_rank(X_approx))
# 结果为16
```

代码 2. sklearn.decomposition.TruncatedSVD()完成截断奇异值分解 | Bk7_Ch15_01.ipynb

我们还用 Streamlit 搭建了图 30 所示 App，用来展示主元数量对图片还原的影响，请大家自行学习 Streamlit_Bk7_Ch15_02.py。

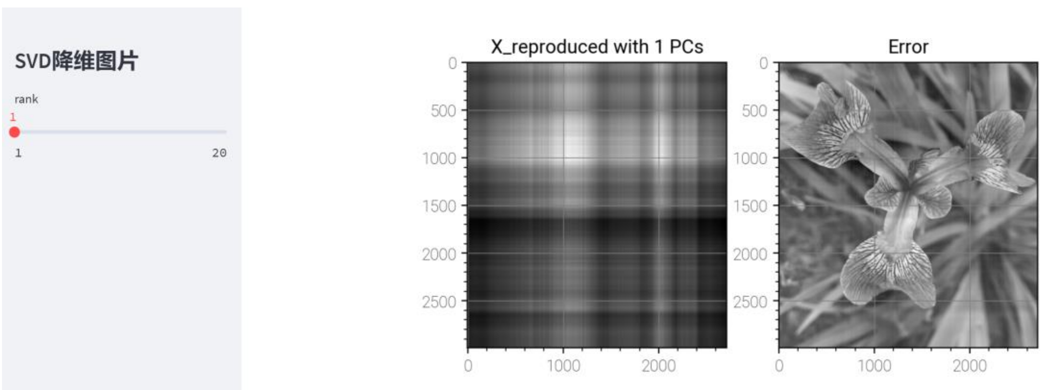


图 30. 展示主元还原图片的 App，Streamlit 搭建 | Streamlit_Bk7_Ch15_02.py

本章介绍了如何用截断型 SVD 分解完成主成分分析。这一章也是回顾奇异值分解的好机会。此外，请大家注意 EVD 和 SVD 的联系。

下一章，我们将比较六种 PCA 技术路线。