

10

Data Projection

数据投影

以鸢尾花数据集为例



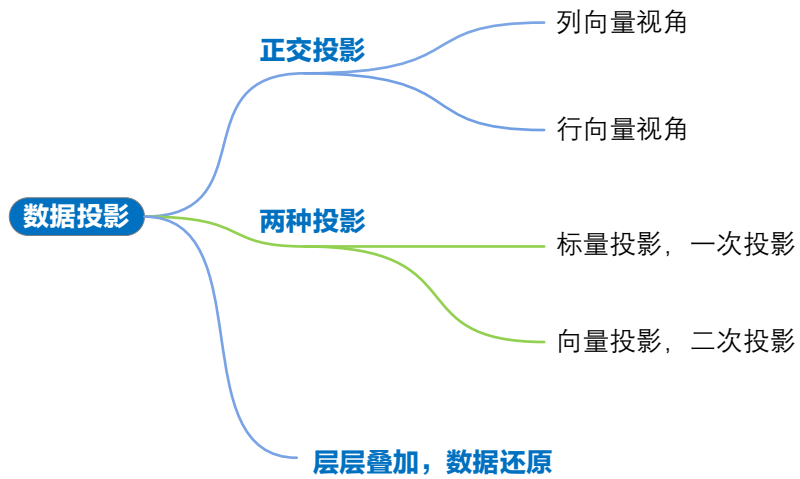
人生就像骑自行车。为了保持平衡，你必须不断移动。

Life is like riding a bicycle. To keep your balance, you must keep moving .

—— 阿尔伯特·爱因斯坦 (Albert Einstein) | 理论物理学家 | 1879 ~ 1955



- ◀ `numpy.linalg.eig()` 特征值分解
- ◀ `seaborn.heatmap()` 绘制热图



10.1 从一个矩阵乘法运算说起

有数据的地方，就有向量；有向量的地方，就有几何！

本章将结合数据、几何、向量三个元素，总结本书前九章主要内容。此外，本章承前启后，它将开启本书下一个重要板块——矩阵分解。

本节和下一节内容会稍微枯燥，请大家耐心读完。之后，本章会用鸢尾花数据集作为例子，给大家展开讲解这两节内容。

正交投影

本章从一个矩阵乘法运算说起：

$$\mathbf{Z} = \mathbf{XV} \quad (1)$$

\mathbf{X} 是数据矩阵，形状为 $n \times D$ ，即 n 行、 D 列。大家很清楚，以鸢尾花数据集为例， \mathbf{X} 每一行代表一个数据点，每一列代表一个特征。

\mathbf{V} 是正交矩阵，即满足 $\mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I}$ 。这意味着 $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_D]$ 是 \mathbb{R}^D 空间的一组规范正交基。

几何视角下，矩阵乘积 \mathbf{XV} 完成的是 \mathbf{X} 向规范正交基 $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_D]$ 投影，乘积 \mathbf{XV} 结果 \mathbf{Z} 代表 \mathbf{X} 在新的规范正交基下的坐标。

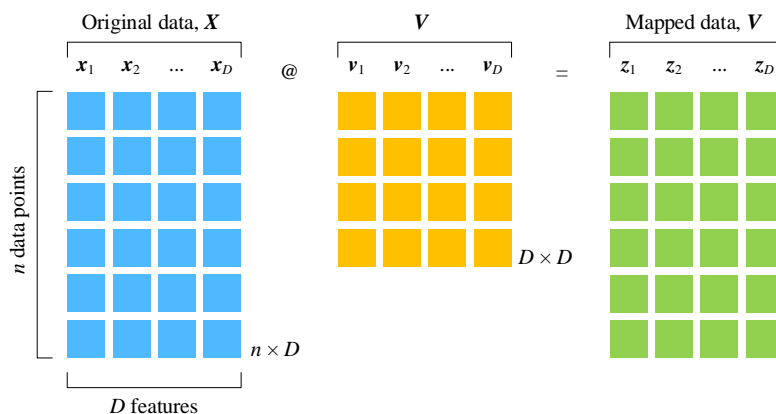


图 1. 数据矩阵 \mathbf{X} 到 \mathbf{Z} 线性变换

本书前文反复提到，一个矩阵可以看成由一系列行向量或列向量构造得到。下面，我们分别从这两个视角来分析。

列向量

将 \mathbf{Z} 和 \mathbf{V} 分别写成各自列向量, (1) 可以展开写成:

$$\begin{aligned} [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \cdots \quad \mathbf{z}_D] &= \mathbf{X} [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_D] \\ &= [\mathbf{X}\mathbf{v}_1 \quad \mathbf{X}\mathbf{v}_2 \quad \cdots \quad \mathbf{X}\mathbf{v}_D] \end{aligned} \quad (2)$$

(2) 这个视角是数据列向量 (即特征) 之间的转换。

提取 (2) 等式左右第 j 列, 得到 \mathbf{Z} 矩阵的第 j 列向量 \mathbf{z}_j 的计算式:

$$\mathbf{z}_j = \mathbf{X}\mathbf{v}_j \quad (3)$$

如图 2 所示, (3) 相当于 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D$ 通过线性组合得到 \mathbf{z}_j , 即:

$$\mathbf{z}_j = \underbrace{[\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_D]}_{\mathbf{X}} \underbrace{\begin{bmatrix} v_{1,j} \\ v_{2,j} \\ \vdots \\ v_{D,j} \end{bmatrix}}_{\mathbf{v}_j} = v_{1,j}\mathbf{x}_1 + v_{2,j}\mathbf{x}_2 + \cdots v_{D,j}\mathbf{x}_D \quad (4)$$

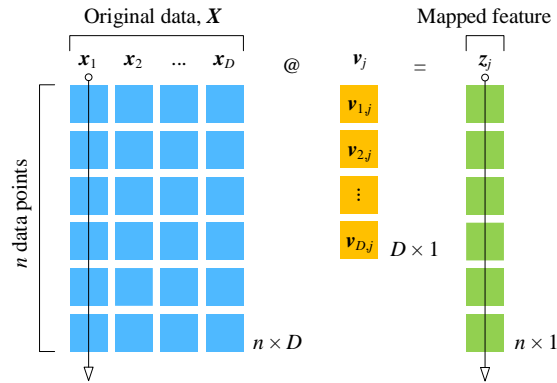


图 2. \mathbf{Z} 第 j 列向量 \mathbf{z}_j 的计算过程

行向量：点坐标

矩阵 \mathbf{X} 的任意行向量 $\mathbf{x}^{(i)}$ 代表其在 \mathbb{R}^D 的坐标, 注意 \mathbb{R}^D 基底为标准正交基 $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_1, \dots, \mathbf{e}_D]$ 。

将 \mathbf{X} 和 \mathbf{Z} 写成行向量形式, (1) 可以写作:

$$\begin{bmatrix} \mathbf{z}^{(1)} \\ \mathbf{z}^{(2)} \\ \vdots \\ \mathbf{z}^{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(1)}\mathbf{V} \\ \mathbf{x}^{(2)}\mathbf{V} \\ \vdots \\ \mathbf{x}^{(n)}\mathbf{V} \end{bmatrix} \quad (5)$$

如图 3 所示, (5) 代表每一行数据点之间的转换关系。即, (5) 的第 i 行 $\mathbf{x}^{(i)}$ 投影得到 \mathbf{Z} 的第 i 行向量 $\mathbf{z}^{(i)}$:

$$\mathbf{z}^{(i)} = \mathbf{x}^{(i)} \mathbf{V} \quad (6)$$

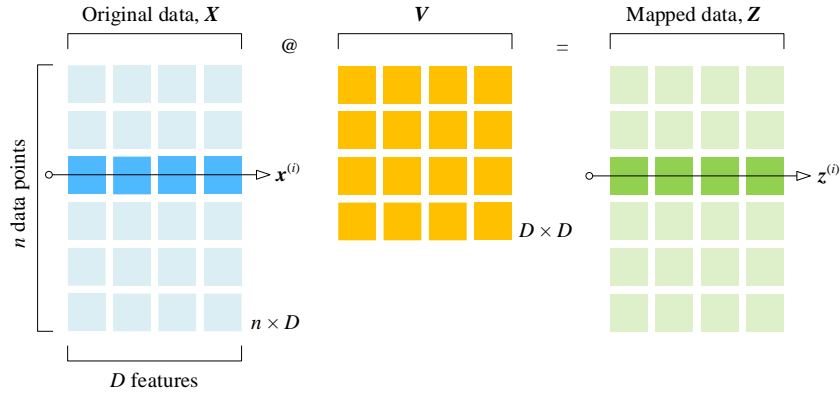


图 3. 每一行数据点之间的转换关系

进一步将 (6) 中 \mathbf{V} 写成 $[\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_D]$, (6) 可以展开得到:

$$\begin{bmatrix} z_{i,1} & z_{i,2} & \cdots & z_{i,D} \end{bmatrix} = \mathbf{x}^{(i)} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_D \end{bmatrix} \quad (7)$$

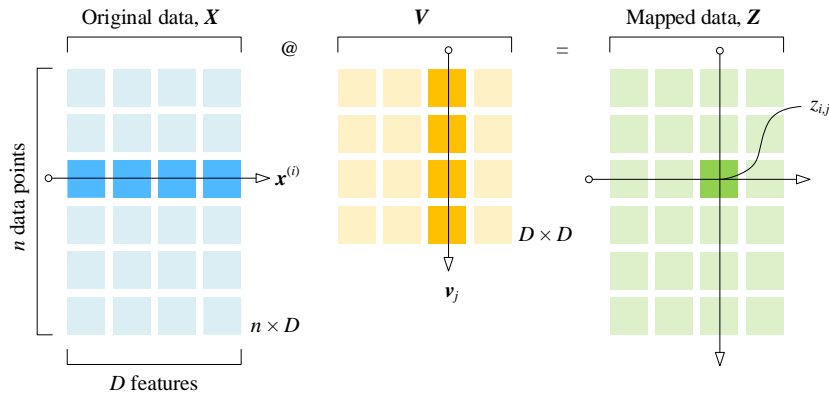


图 4. 每一行数据点向 \mathbf{v}_j 投影

取出 (7) 中向量 $\mathbf{z}^{(i)}$ 第 j 列元素 $z_{i,j}$, 对应的运算为:

$$z_{i,j} = \mathbf{x}^{(i)} \mathbf{v}_j \quad (8)$$

图 4 对应 (8) 运算。

从空间视角来看，如图 5 所示， $\mathbf{x}^{(i)}$ 位于 \mathbb{R}^D 空间，而 $\mathbf{x}^{(i)}$ 正交投影到子空间 (subspace) $\text{span}(\mathbf{v}_j)$ 对应的坐标点就是 $z_{i,j}$ 。换句话说， $z_{i,j}$ 是 $\mathbf{x}^{(i)}$ 在 $\text{span}(\mathbf{v}_j)$ 的像 (image)。 $\mathbf{x}^{(i)}$ 在 \mathbb{R}^D 空间是 D 维，在 $\text{span}(\mathbf{v}_j)$ 仅是 1 维。图 5 中，从左边 \mathbb{R}^D 空间到右侧 $\text{span}(\mathbf{v}_j)$ 这个投影过程，是个降维过程，数据发生压缩。

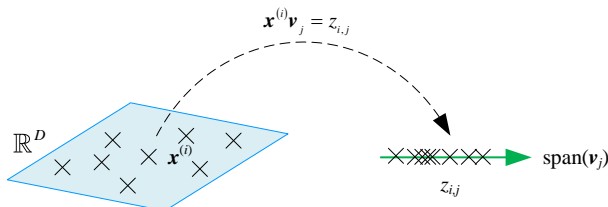


图 5. \mathbb{R}^D 空间数据点投影到 $\text{span}(\mathbf{v}_j)$

10.2 二次投影 + 层层叠加

本书前文给出下面这个看似莫名其妙的矩阵乘法：

$$\mathbf{X} = \mathbf{X} \mathbf{I} = \mathbf{X} \mathbf{V} \mathbf{V}^T = \mathbf{X} \quad (9)$$

也就是说，数据矩阵 \mathbf{X} 乘以单位阵，结果为其本身！这个显而易见的等式，有何意义？

其实，这个看似再简单不过的矩阵运算背后实际藏着“二次投影”和“层层叠加”这两重几何操作！下面，我们就解密这两个几何操作。

将 \mathbf{V} 写成 $[\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_D]$ ，代入 (9) 得到：

$$\begin{aligned} \mathbf{X} &= \mathbf{X} \mathbf{V} \mathbf{V}^T = \mathbf{X} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_D \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_D^T \end{bmatrix} \\ &= \underbrace{\mathbf{X} \mathbf{v}_1 \mathbf{v}_1^T}_{\mathbf{X}_1} + \underbrace{\mathbf{X} \mathbf{v}_2 \mathbf{v}_2^T}_{\mathbf{X}_2} + \cdots + \underbrace{\mathbf{X} \mathbf{v}_D \mathbf{v}_D^T}_{\mathbf{X}_D} \end{aligned} \quad (10)$$

令，

$$\mathbf{X}_j = \mathbf{X} \mathbf{v}_j \mathbf{v}_j^T \quad (11)$$

图 6 所示为上述运算， \mathbf{X}_j 的形状和原数据矩阵 \mathbf{X} 完全相同。我们称图 6 为二次投影，一会将解释原因。

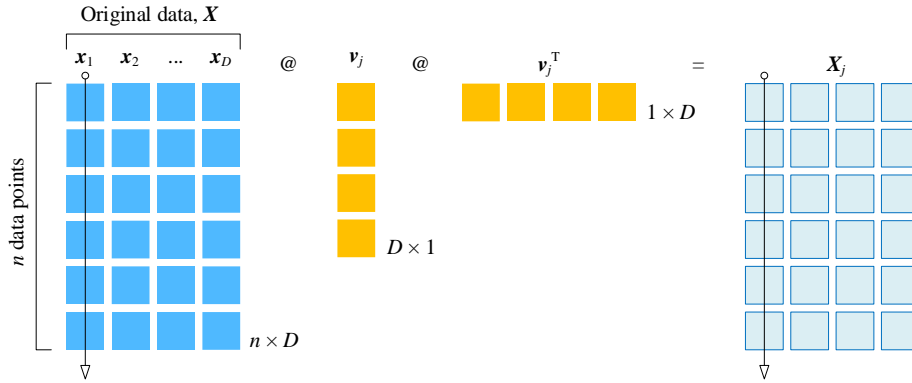


图 6. 二次投影

(10) 可以写成：

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_D \quad (12)$$

上式就是“层层叠加”。如图 7 所示， D 个形状完全相同的数据，层层叠加还原原始数据 \mathbf{X} 。

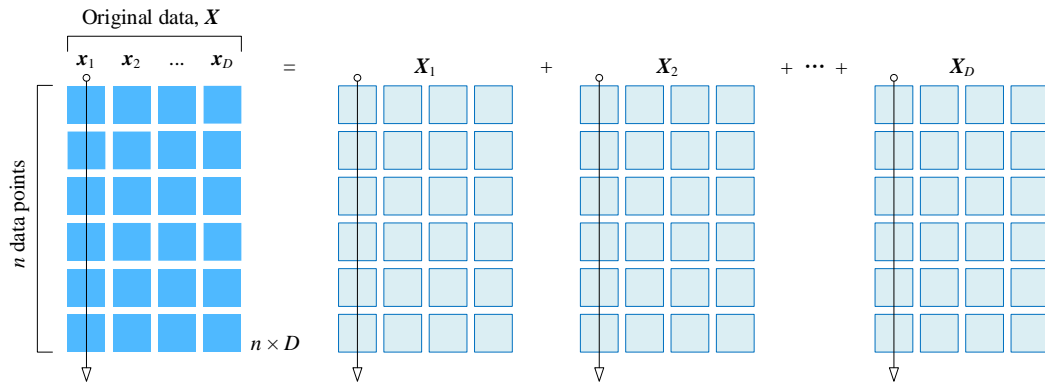


图 7. 层层叠加

二次投影

下面，我们聊聊“二次投影”。

取出 (7) 中向量 \mathbf{X}_j 第 j 行元素，对应的运算为：

$$\mathbf{x}_j^{(i)} = \mathbf{x}^{(i)} \mathbf{v}_j \mathbf{v}_j^T = z_{i,j} \mathbf{v}_j^T \quad (13)$$

如 (8) 所示，上式中 $z_{i,j}$ 就是 $\mathbf{x}^{(i)}$ 正交投影到子空间 $\text{span}(\mathbf{v}_j)$ 对应的坐标点，这是第一次投影。

而 $z_{i,j} \mathbf{v}_j^T$ 得到的是 $z_{i,j}$ 在 \mathbb{R}^D 的坐标点，这是第二次投影。

整个二次投影的过程如图 8 所示。注意， $\mathbf{x}^{(i)}$ 和 $z_{i,j}\mathbf{v}_j^T$ 都用行向量表达坐标点。

可以这样理解， $\mathbf{x}^{(i)} \rightarrow z_{i,j}$ 代表“标量投影”； $\mathbf{x}^{(i)} \rightarrow \mathbf{x}^{(i)}\mathbf{v}_j\mathbf{v}_j^T$ 则是“向量投影”。

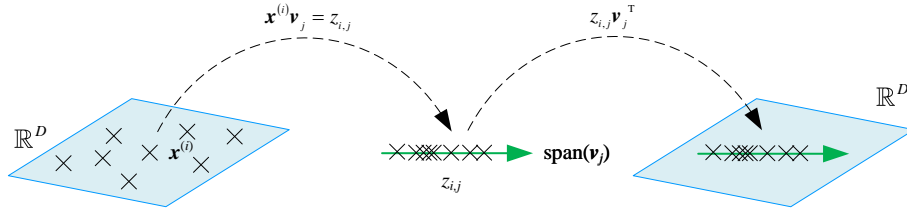


图 8. \mathbb{R}^D 空间数据点先投影到 $\text{span}(\mathbf{v}_j)$ ，再投影回到 \mathbb{R}^D

向量投影：张量积

将 (11) 写成张量积的形式：

$$\mathbf{X}_j = \mathbf{X}\mathbf{v}_j \otimes \mathbf{v}_j \quad (14)$$

\mathbf{X}_j 就是 \mathbf{X} 经过“降维”到 $\text{span}(\mathbf{v}_j)$ 后，再正交投影到 \mathbb{R}^D 中得到的“像”。 \mathbf{X}_j 也是 \mathbf{X} 在 \mathbf{v}_j 上的向量投影。

张量积 $\mathbf{v}_j \otimes \mathbf{v}_j$ 本身完成“多维 \rightarrow 一维”+“一维 \rightarrow 多维”这两步投影。

很显然，

$$\text{rank}(\mathbf{v}_j \otimes \mathbf{v}_j) = 1 \Rightarrow \text{rank}(\mathbf{X}_j) = 1 \quad (15)$$

所以，在 \mathbb{R}^D 空间中， \mathbf{X}_j 所有数据点在一条直线上，和 \mathbf{v}_j 同向。也就是说，虽然 \mathbf{X}_j 在 D 维空间 \mathbb{R}^D 中，它实际上只有 1 个维度，即 $\dim(\mathbf{v}_j) = \dim(\mathbf{v}_j \otimes \mathbf{v}_j) = 1$ ， $\text{rank}(\mathbf{X}_j) = 1$ 。

利用张量积，(10) 可以写成：

$$\mathbf{X} = \underbrace{\mathbf{X}\mathbf{v}_1 \otimes \mathbf{v}_1}_{\mathbf{X}_1} + \underbrace{\mathbf{X}\mathbf{v}_2 \otimes \mathbf{v}_2}_{\mathbf{X}_2} + \cdots + \underbrace{\mathbf{X}\mathbf{v}_D \otimes \mathbf{v}_D}_{\mathbf{X}_D} \quad (16)$$

可以这样理解上式， \mathbf{X} 分别二次投影到规范正交基 $[\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_D]$ 每个列向量 \mathbf{v}_j 所代表的子空间中，获得 $\mathbf{X}_1, \mathbf{X}_2 \dots \mathbf{X}_D$ 。而 $\mathbf{X}_1, \mathbf{X}_2 \dots \mathbf{X}_D$ 层层叠加还原原始数据 \mathbf{X} 。

再进一步，(16) 消去得到：

$$\mathbf{I} = \mathbf{v}_1 \otimes \mathbf{v}_1 + \mathbf{v}_2 \otimes \mathbf{v}_2 + \cdots + \mathbf{v}_D \otimes \mathbf{v}_D \quad (17)$$

标准正交基：便于理解

标准正交基是特殊的规范正交基。为了方便理解，我们用标准正交基 $[e_1, e_2, \dots, e_D]$ 替换 (16) 中的 $[v_1, v_2, \dots, v_D]$ ，得到：

$$X = Xe_1 \otimes e_1 + Xe_2 \otimes e_2 + \dots + Xe_D \otimes e_D \quad (18)$$

展开上式中第一项得到：

$$X_1 = Xe_1 \otimes e_1 = X \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} x_1 & x_2 & \dots & x_D \end{bmatrix}}_X \begin{bmatrix} 1 & 0 & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{bmatrix} = \begin{bmatrix} x_1 & 0 & \dots & 0 \end{bmatrix} \quad (19)$$

Xe_1 得到的是 X 的每一行在 $\text{span}(e_1)$ 这个子空间的坐标，即 x_1 。而 $Xe_1 \otimes e_1$ 告诉我们的是 Xe_1 在 D 维空间 \mathbb{R}^D 中坐标值。

此后每一项 X_j 可以写成：

$$\begin{aligned} X_2 &= Xe_2 \otimes e_2 = \begin{bmatrix} 0 & x_2 & \dots & 0 \end{bmatrix} \\ &\vdots \\ X_D &= Xe_D \otimes e_D = \begin{bmatrix} 0 & 0 & \dots & x_D \end{bmatrix} \end{aligned} \quad (20)$$

也就是说，这个每次计算 $Xe_j \otimes e_j$ 投影就是仅保留 X 的第 j 列 x_j ，其他元素置 0。

因此，(18) 可以写成：

$$X = \underbrace{\begin{bmatrix} x_1 & 0 & \dots & 0 \end{bmatrix}}_{x_1} + \underbrace{\begin{bmatrix} 0 & x_2 & \dots & 0 \end{bmatrix}}_{x_2} + \dots + \underbrace{\begin{bmatrix} 0 & 0 & \dots & x_D \end{bmatrix}}_{x_D} \quad (21)$$

图 9 所示为上式二次投影与叠加过程。

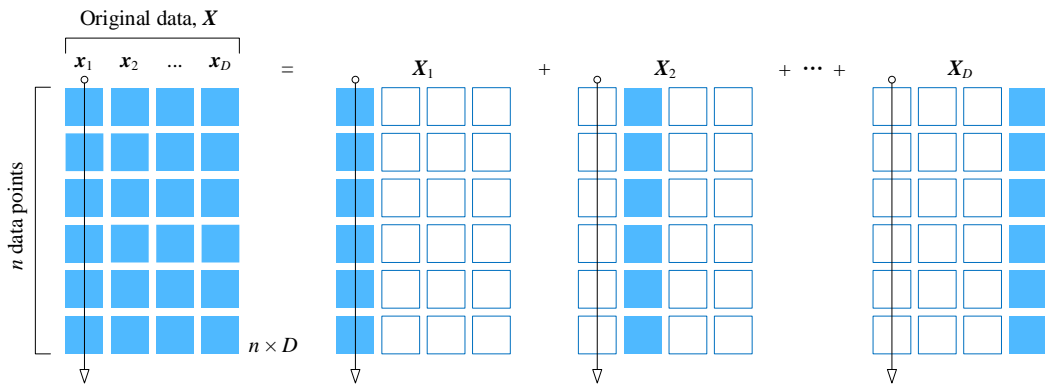


图 9. 标准正交基 $[e_1, e_2, \dots, e_D]$ 中二次投影与叠加

回过头再看 (9)，我们知道这个过程是先从标准正交基 $[e_1, e_2, \dots, e_D]$ 到规范正交基 $[v_1, v_2, \dots, v_D]$ 的投影，然后再投影回到标准正交基 $[e_1, e_2, \dots, e_D]$ ：

$$X \xrightarrow{V} XV \xrightarrow{V^T} X \quad (22)$$

看到这里，有些读者有可能已经晕头转向。下面利用鸢尾花数据集做例子，帮大家更直观理解本节内容。

10.3 二特征数据投影：标准正交基

本节以二特征矩阵为例讲解何谓“二次投影”和“层层叠加”。数据矩阵选取鸢尾花数据集前两列——花萼长度、花萼宽度，这样数据矩阵 $X_{150 \times 2}$ 的形状为 150×2 。

水平方向投影

如图 10 所示， $X_{150 \times 2}$ 向水平方向投影，即 $X_{150 \times 2}$ 向 e_1 投影。以图中红点 A 为例， A 的坐标为 $(5, 2)$ ，它在 e_1 方向上的投影坐标为：

$$\begin{bmatrix} 5 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 5 \quad (23)$$

e_1

注意，5 代表的是 A 在 $\text{span}(e_1)$ 空间中的坐标值，而 $\text{span}(e_1)$ 显然为一维空间。

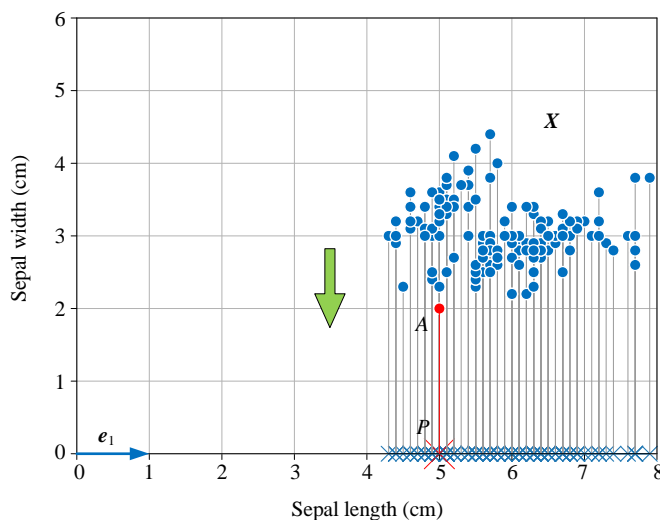
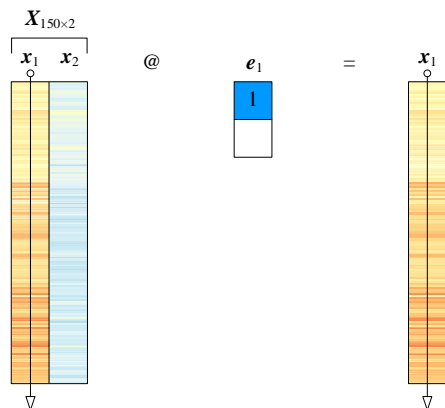


图 10. 二特征数据矩阵 $X_{150 \times 2}$ 向 e_1 投影，一次投影

如图 11 热图所示， $X_{150 \times 2}$ 向 e_1 投影结果相当于保留了 $X_{150 \times 2}$ 第一列数据：

$$z_1 = X e_1 = x_1 \quad (24)$$

图 11. 数据热图，二特征数据矩阵 $X_{150 \times 2}$ 向 e_1 投影，一次投影

大家可能会好奇，既然图 10 中 $X_{150 \times 2}$ 向水平方向投影结果都可以画在图 10 直角坐标系中，也就是 $\text{span}(e_1, e_2)$ 二维空间中，这些投影点一定有其二维坐标值。

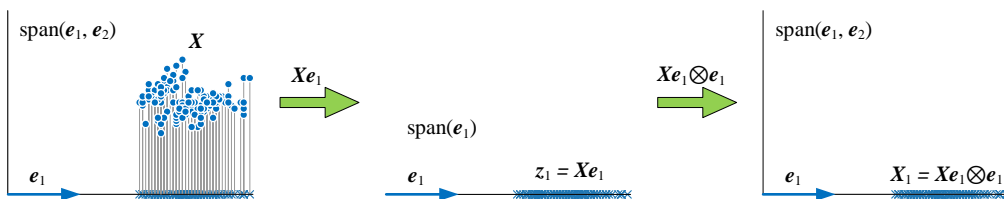
很明显，以 A 为例， A 在横轴投影点 P 在 $\text{span}(e_1, e_2)$ 的坐标值为 $(5, 0)$ 。这个结果是怎么得到的？

这就用到了本章前文讲到的“二次投影”，相当于在 (23) 基础上再次投影。第二次投影相当于“升维”，从一维升到二维。

以点 A 为例，“二次投影”对应的计算为：

$$[5 \ 2]e_1 \otimes e_1 = [5 \ 2]e_1 e_1^T = [5 \ 2] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = [5 \ 0] \quad (25)$$

上式对应的计算如图 12 所示。

图 12. 二特征数据矩阵 X 向 e_1 投影，二次投影

X 在 e_1 二次投影对应 $\text{span}(e_1, e_2)$ 坐标值为 X_1 ：

$$X_1 = Xe_1 \otimes e_1 = Xe_1 e_1^T = X \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = [x_1 \ 0] \quad (26)$$

图 13 所示为上述运算对应热图。值得注意的是，这种数据“升维”，不代表数据信息增多。显然，上式中 X_1 的秩仍为 1，即 $\text{rank}(X_1)$ 。

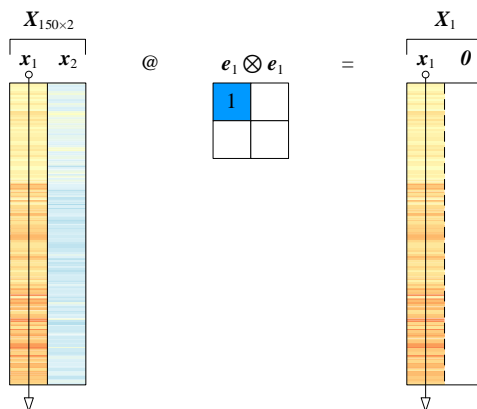


图 13. 数据热图，二特征数据矩阵 $X_{150 \times 2}$ 向 e_1 投影，二次投影

向竖直方向投影

如图 10 所示， $X_{150 \times 2}$ 向竖直方向投影，即 $X_{150 \times 2}$ 向 e_2 投影。还是以 A 点为例， $A(5, 2)$ 在 e_2 方向上的标量投影为：

$$\begin{bmatrix} 5 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 2 \quad (27)$$

e_2

2 代表的是 A 在 $\text{span}(e_2)$ 空间中的坐标值， $\text{span}(e_2)$ 同样为一维空间。

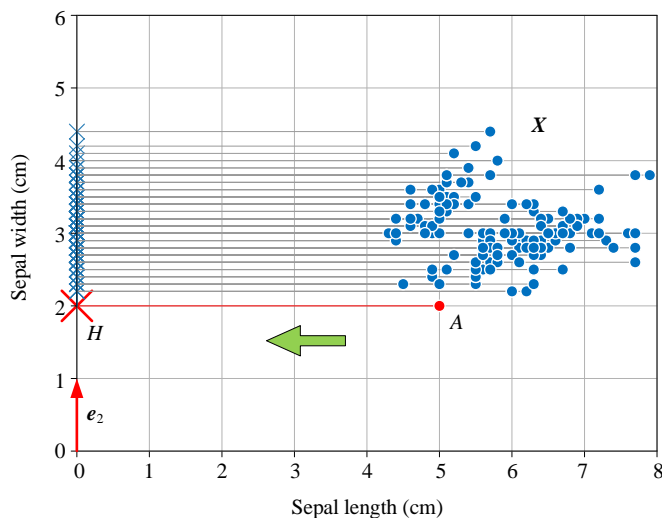
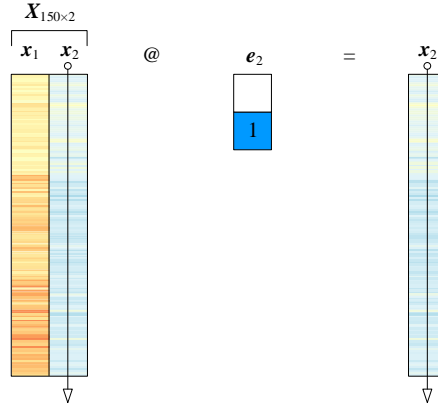
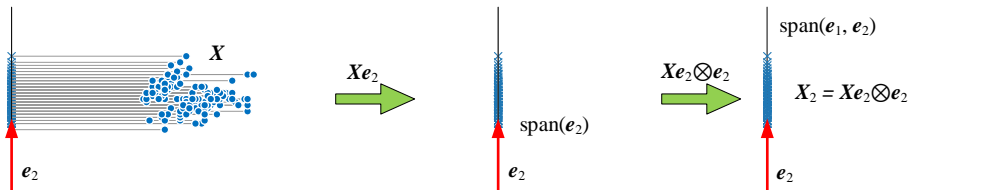


图 14. 二特征数据矩阵 $X_{150 \times 2}$ 向 e_2 方向标量投影，一次投影图 15. 数据热图，二特征数据矩阵 $X_{150 \times 2}$ 向 e_2 投影，一次投影

同样利用“二次投影”，得到 A 在竖直方向投影点 H 在 $\text{span}(e_1, e_2)$ 的坐标值为 $(0, 2)$ ：

$$[5 \ 2]e_2 \otimes e_2 = [5 \ 2]e_2 e_2^T = [5 \ 2] \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = [0 \ 2] \quad (28)$$

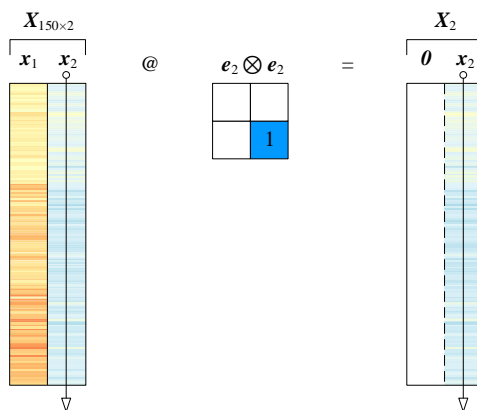
上式对应的计算如图 16 所示。

图 16. 二特征数据矩阵 $X_{150 \times 2}$ 向 e_2 方向标量投影，二次投影

$X_{150 \times 2}$ 在 e_2 二次投影对应应在 $\text{span}(e_1, e_2)$ 对的矩阵为 X_2 ：

$$X_2 = X e_2 \otimes e_2 = X e_2 e_2^T = X \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (29)$$

上式对应的热图运算为图 17。 X_2 第一列向量为 0 ，第二列向量为 x_2 。

图 17. 数据 heatmap，二特征数据矩阵 $X_{150 \times 2}$ 向 e_2 投影，二次投影

叠加

如图 18 所示，以 A 为例， $P(5, 0)$ 和 $H(0, 2)$ 叠加得到点 A 坐标 $(5, 2)$ 。这也相当于两个向量叠加得到一个向量，即：

$$\begin{bmatrix} 5 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix} \quad (30)$$

或，

$$\begin{bmatrix} 5 \\ 0 \end{bmatrix}^T + \begin{bmatrix} 0 \\ 2 \end{bmatrix}^T = \begin{bmatrix} 5 \\ 2 \end{bmatrix}^T \quad (31)$$

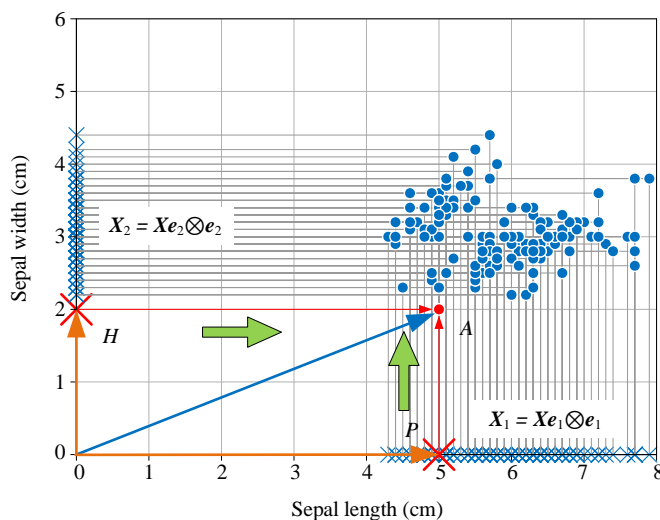


图 18. 数据叠加还原散点图

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

如图 19 所示, X_1 和 X_2 叠加还原 $X_{150 \times 2}$:

$$\begin{aligned}
 X_{150 \times 2} &= X_1 + X_2 \\
 &= X(e_1 \otimes e_1 + X e_2 \otimes e_2) \\
 &= X(e_1 e_1^T + e_2 e_2^T) \\
 &= X \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right) = XI
 \end{aligned} \tag{32}$$

图 20 所示为上述运算对应的热图。

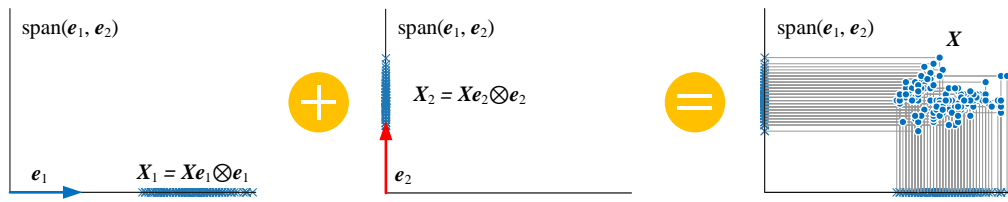


图 19. 数据叠加还原 $X_{150 \times 2}$

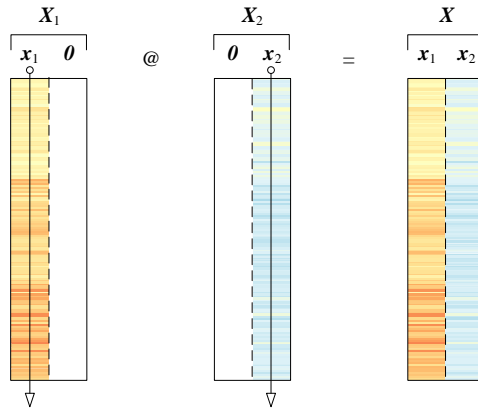


图 20. 数据热图, 叠加还原 $X_{150 \times 2}$

10.4 二特征数据投影：规范正交基

本节分析 $X_{150 \times 2}$ 在三个不同规范正交基投影情况。

第一个规范正交基

给定如下规范正交基 $V = [v_1, v_2]$:

$$\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2] = \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix} \quad (33)$$

从几何变换角度来上， \mathbf{V} 就是一个旋转矩阵。

如图 21 所示，同样以点 $A(5, 2)$ 为例， A 在 \mathbf{v}_1 方向标量投影为：

$$\begin{bmatrix} 5 & 2 \end{bmatrix} \underbrace{\begin{bmatrix} \sqrt{3}/2 \\ 1/2 \end{bmatrix}}_{\mathbf{v}_1} \approx 5.33 \quad (34)$$

也就是说， A 在 $\text{span}(\mathbf{v}_1)$ 投影 H 的坐标值为 5.33，对应向量可以写成 $5.33\mathbf{v}_1$ 。

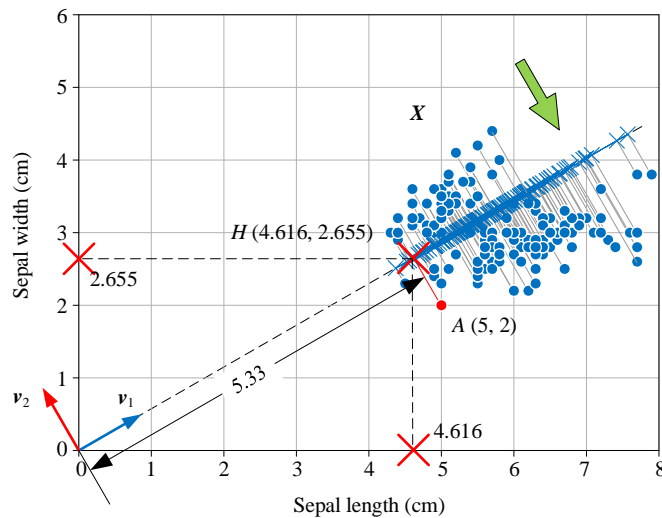


图 21. 二特征数据矩阵 $\mathbf{X}_{150 \times 2}$ 向 \mathbf{v}_1 投影

通过二次投影获得 H 在 $\text{span}(\mathbf{e}_1, \mathbf{e}_2)$ 坐标值，相当于把 \mathbf{v}_1 的具体值代入 $5.33\mathbf{v}_1$ ，即：

$$\begin{bmatrix} 5 & 2 \end{bmatrix} \mathbf{v}_1 \otimes \mathbf{v}_1 = \begin{bmatrix} 5 & 2 \end{bmatrix} \mathbf{v}_1 \mathbf{v}_1^T = \begin{bmatrix} 5 & 2 \end{bmatrix} \begin{bmatrix} 3/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 1/4 \end{bmatrix} \approx \begin{bmatrix} 4.616 & 2.665 \end{bmatrix} \quad (35)$$

这就是 H 在图 21 中坐标值。

$\mathbf{X}_{150 \times 2}$ 在 \mathbf{v}_1 投影 \mathbf{z}_1 为：

$$\mathbf{z}_1 = \mathbf{X} \mathbf{v}_1 = \underbrace{\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \sqrt{3}/2 \\ 1/2 \end{bmatrix}}_{\mathbf{v}_1} \approx 0.866\mathbf{x}_1 + 0.5\mathbf{x}_2 \quad (36)$$

即， \mathbf{z}_1 相当于 \mathbf{x}_1 和 \mathbf{x}_2 的线性组合。

$\mathbf{X}_{150 \times 2}$ 在 \mathbf{v}_1 二次投影 \mathbf{X}_1 为：

$$\mathbf{X}_1 = \mathbf{X}\mathbf{v}_1 \otimes \mathbf{v}_1 = \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T \approx \underbrace{\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}}_{\mathbf{x}} \begin{bmatrix} 0.750 & 0.433 \\ 0.433 & 0.250 \end{bmatrix} = \begin{bmatrix} 0.750\mathbf{x}_1 + 0.433\mathbf{x}_2 & 0.433\mathbf{x}_1 + 0.250\mathbf{x}_2 \end{bmatrix} \quad (37)$$

如图 22 所示，同样以点 A (5, 2) 为例，A 在 \mathbf{v}_2 方向标量为：

$$\begin{bmatrix} 5 & 2 \end{bmatrix} \underbrace{\begin{bmatrix} -1/2 \\ \sqrt{3}/2 \end{bmatrix}}_{\mathbf{v}_2} \approx -0.7679 \quad (38)$$

即 A 在 $\text{span}(\mathbf{v}_2)$ 投影点的坐标值为 -0.7679，对应向量可以写成 $-0.7679\mathbf{v}_2$ 。通过二次投影获得其坐标值：

$$\begin{bmatrix} 5 & 2 \end{bmatrix} \mathbf{v}_2 \otimes \mathbf{v}_2 = \begin{bmatrix} 5 & 2 \end{bmatrix} \mathbf{v}_2 \mathbf{v}_2^T = \begin{bmatrix} 5 & 2 \end{bmatrix} \begin{bmatrix} 1/4 & -\sqrt{3}/4 \\ -\sqrt{3}/4 & 3/4 \end{bmatrix} \approx \begin{bmatrix} 0.384 & -0.665 \end{bmatrix} \quad (39)$$

(35) 和 (39) 之和还原 A 坐标值 (5, 2)：

$$\begin{bmatrix} 5 & 2 \end{bmatrix} (\mathbf{v}_1 \otimes \mathbf{v}_1 + \mathbf{v}_2 \otimes \mathbf{v}_2) = \begin{bmatrix} 5 & 2 \end{bmatrix} \left\{ \begin{bmatrix} 3/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 1/4 \end{bmatrix} + \begin{bmatrix} 1/4 & -\sqrt{3}/4 \\ -\sqrt{3}/4 & 3/4 \end{bmatrix} \right\} = \begin{bmatrix} 5 & 2 \end{bmatrix} \quad (40)$$

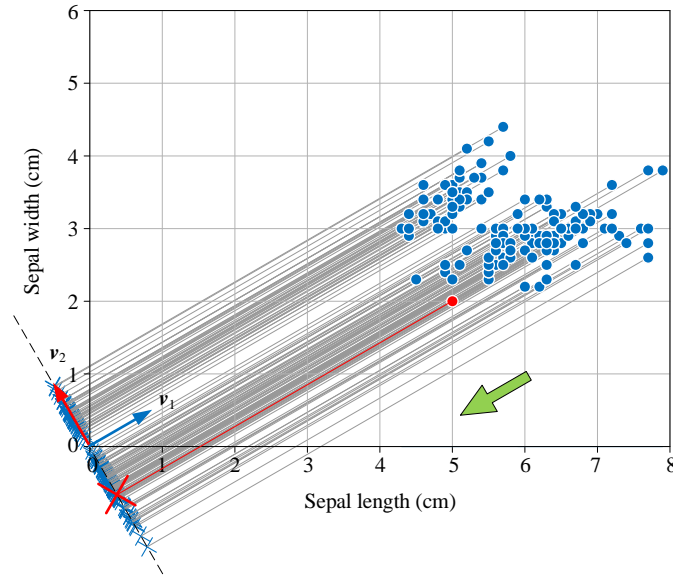


图 22. 二特征数据矩阵 $\mathbf{X}_{150 \times 2}$ 向 \mathbf{v}_2 投影

$\mathbf{X}_{150 \times 2}$ 在 \mathbf{v}_2 投影 \mathbf{z}_2 为：

$$\mathbf{z}_2 = \mathbf{X}\mathbf{v}_2 = \underbrace{\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}}_{\mathbf{x}} \underbrace{\begin{bmatrix} -1/2 \\ \sqrt{3}/2 \end{bmatrix}}_{\mathbf{v}_2} \approx -0.5\mathbf{x}_1 + 0.866\mathbf{x}_2 \quad (41)$$

z_2 也是 x_1 和 x_2 的线性组合。

$X_{150 \times 2}$ 在 v_2 二次投影 X_2 为：

$$X_2 = Xv_2 \otimes v_2 = Xv_2 v_2^T \approx \underbrace{\begin{bmatrix} x_1 & x_2 \end{bmatrix}}_X \begin{bmatrix} 0.250 & -0.433 \\ -0.433 & 0.750 \end{bmatrix} = \begin{bmatrix} 0.250x_1 - 0.433x_2 & -0.433x_1 + 0.750x_2 \end{bmatrix} \quad (42)$$

(37) 和 (42) 叠加还原 X ：

$$X_1 + X_2 = Xv_1 \otimes v_1 + Xv_2 \otimes v_2 = X \left\{ \begin{bmatrix} 0.750 & 0.433 \\ 0.433 & 0.250 \end{bmatrix} + \begin{bmatrix} 0.250 & -0.433 \\ -0.433 & 0.750 \end{bmatrix} \right\} = X \quad (43)$$

第二个规范正交基

给定如下规范正交基 $W = [w_1, w_2]$ ：

$$W = [w_1 \quad w_2] = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \quad (44)$$

图 23 和图 24 所示为二特征数据矩阵 $X_{150 \times 2}$ 向 w_1 和 w_2 投影。请按照本节之前分析 V 的逻辑，自行分析数据在 W 中的投影。

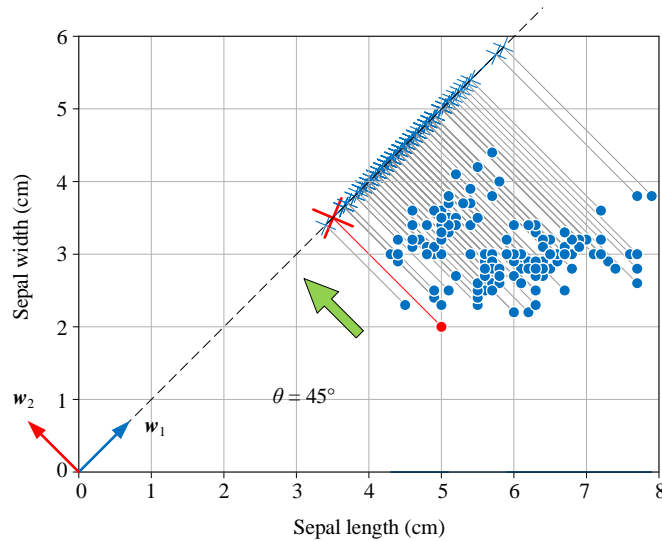
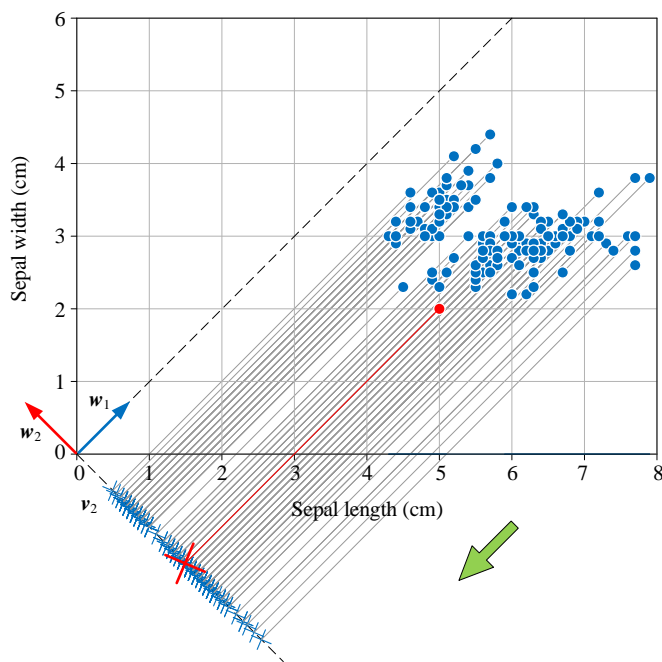


图 23. 二特征数据矩阵 $X_{150 \times 2}$ 向 w_1 投影

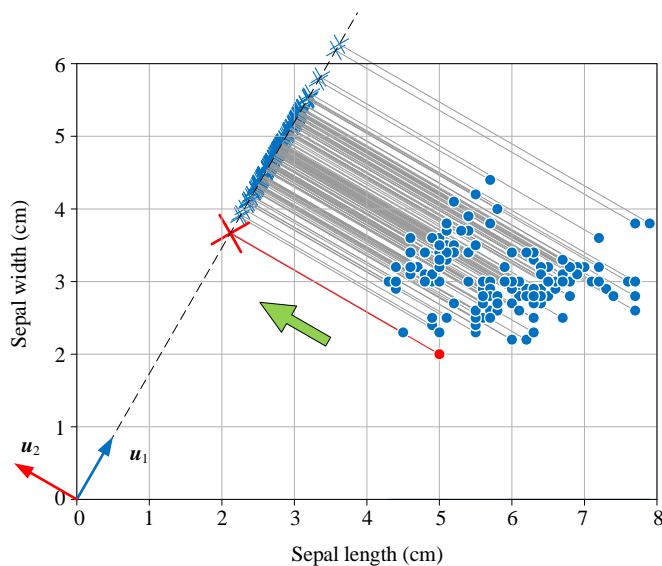
图 24. 二特征数据矩阵 $X_{150 \times 2}$ 向 w_2 投影

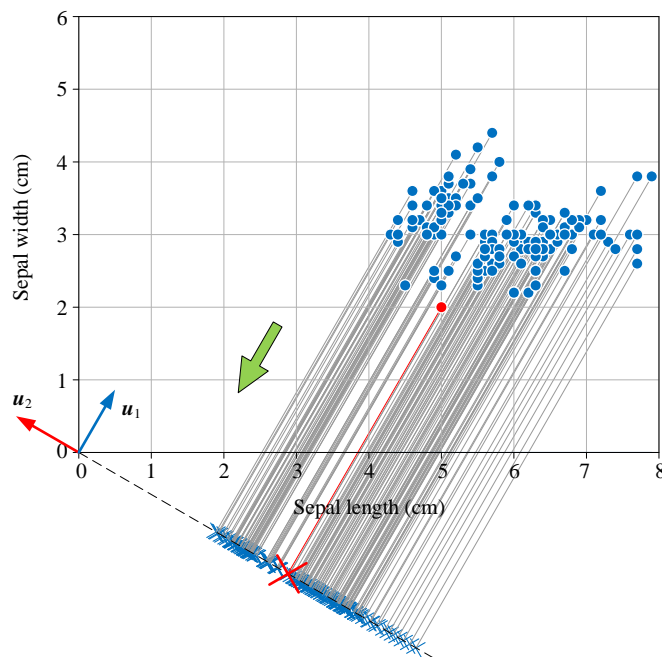
第三个规范正交基

给定如下规范正交基 $U = [u_1, u_2]$:

$$U = [u_1 \quad u_2] = \begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix} \quad (45)$$

图 25 和图 26 所示为二特征数据矩阵 $X_{150 \times 2}$ 向 w_1 和 w_2 投影。请自行分析数据在 U 中的投影。

图 25. 二特征数据矩阵 $X_{150 \times 2}$ 向 u_1 投影

图 26. 二特征数据矩阵 $X_{150 \times 2}$ 向 u_2 投影

10.5 四特征数据投影：标准正交基

本章最后两节以四特征数据矩阵为例，推广前文分析思路。

本节先从最简单的标准正交基 $[e_1, e_2, \dots, e_D]$ 入手。

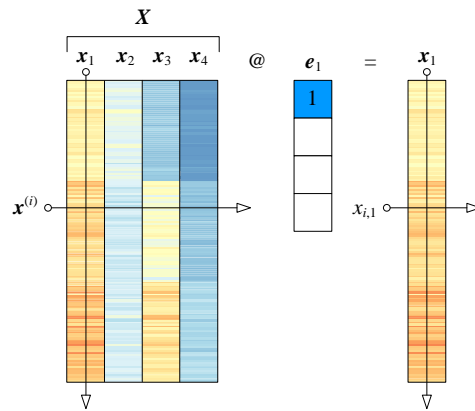
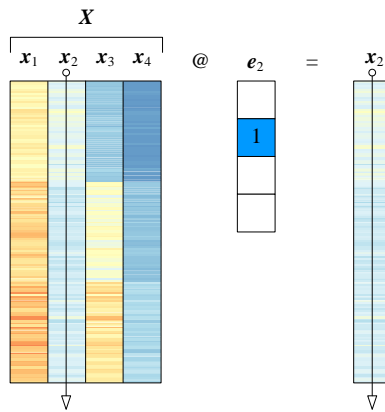
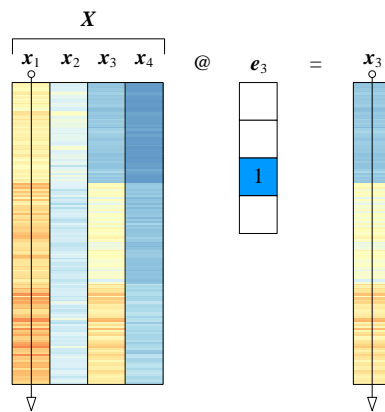
一次投影：标量投影

前文提到过，一次投影实际上就是“标量投影”。图 27 所示为鸢尾花数据集矩阵 X 在 e_1 方向上标量投影的运算热图。

从行向量角度来看， $\mathbf{x}^{(i)} e_1 \rightarrow x_{i,1}$ 代表 \mathbb{R}^D 空间坐标值 $\mathbf{x}^{(i)}$ ，投影到 $\text{span}(e_1)$ 这个子空间后，得到的坐标值变成 $x_{i,1}$ 。再次强调， $x_{i,1}$ 是 $\mathbf{x}^{(i)}$ 在 $\text{span}(e_1)$ 的坐标值。

从列向量角度来看， $[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4] e_1 \rightarrow \mathbf{x}_1$ ，是一个线性组合过程。而 $e_1 = [1, 0, 0, 0]^T$ ，所以组合的结果只保留了鸢尾花数据集第一列 \mathbf{x}_1 ，即花萼长度，这个特征的所有样本数据。

请大家按照这个思路分析图 28、图 29、图 30 三幅热图运算。

图 27. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_1 投影，一次投影图 28. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_2 投影，一次投影图 29. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_3 投影，一次投影

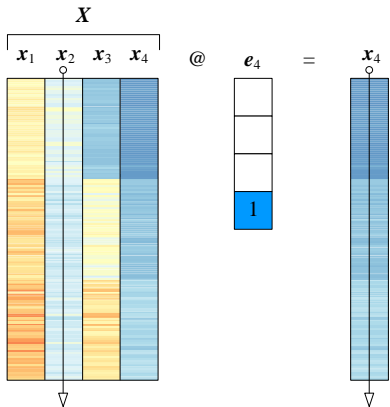


图 30. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_4 投影，一次投影

二次投影

如前文所述，本章所谓的“二次投影”实际上就是向量投影。如图 31 所示， X 向 e_1 方向向量投影结果就是 X 和 $e_1 \otimes e_1$ 的矩阵乘积。乘积结果是，只保留鸢尾花数据集第一列——花萼长度，其他数据均置 0。请大家按照这个思路自行分析图 32、图 33、图 34。

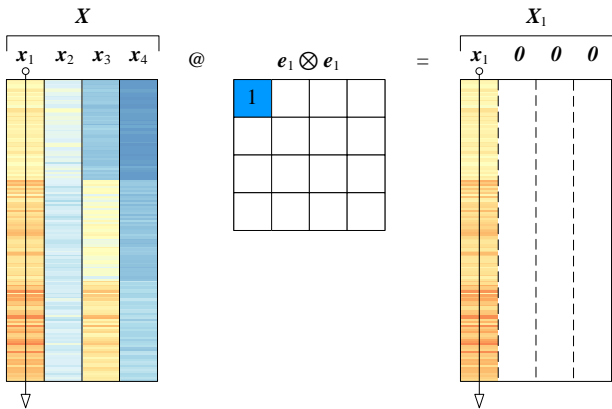


图 31. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_1 方向向量投影，二次投影

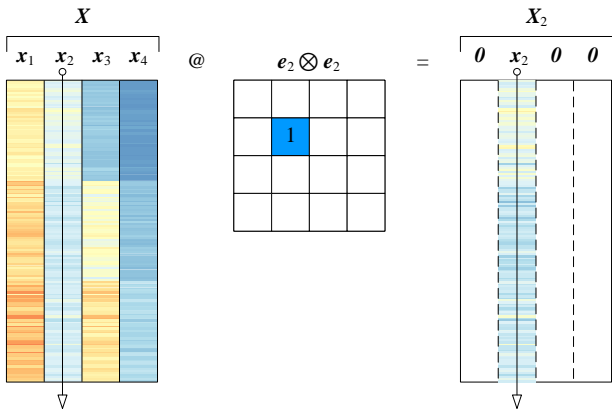
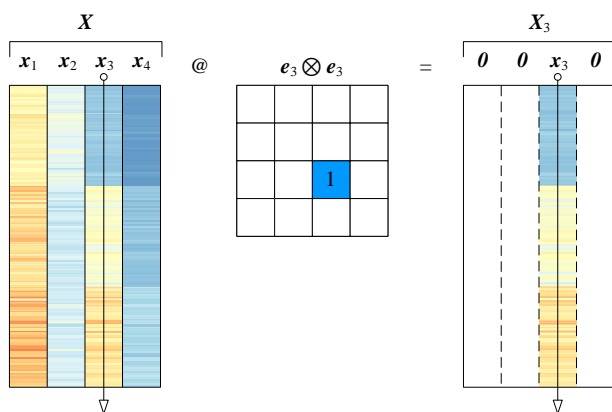
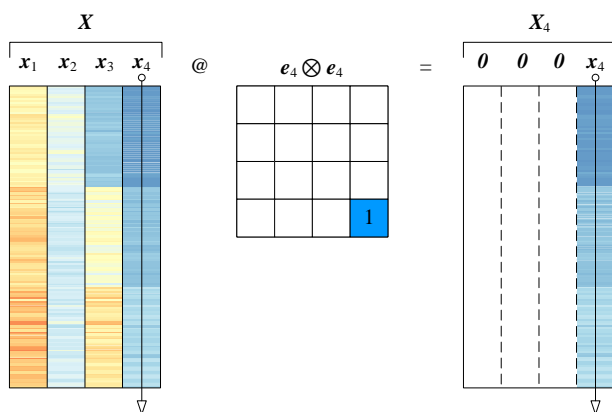


图 32. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_2 方向向量投影，二次投影

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。
版权归清华大学出版社所有，请勿商用，引用请注明出处。
代码及 PDF 文件下载：<https://github.com/Visualize-ML>
本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>
欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

图 33. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_3 方向向量投影，二次投影图 34. 四特征数据矩阵 $X_{150 \times 4}$ 向 e_4 方向向量投影，二次投影

两个方向投影

本节之前提到的都是向单一方向投影。下面，我们用一个例子说明向两个方向投影。

如图 35 所示， X 向 $[e_1, e_2]$ 方向标量投影，这个过程也相当于降维，从 4 维降到 2 维，只保留了鸢尾花花萼长度、花萼宽度两个特征。

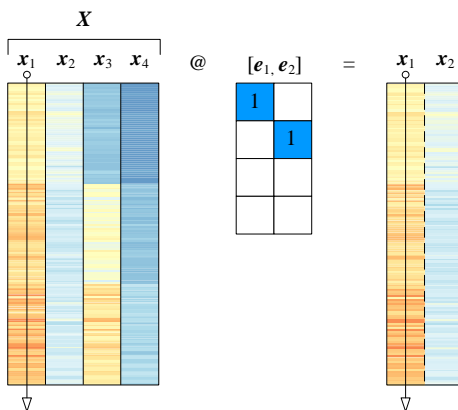
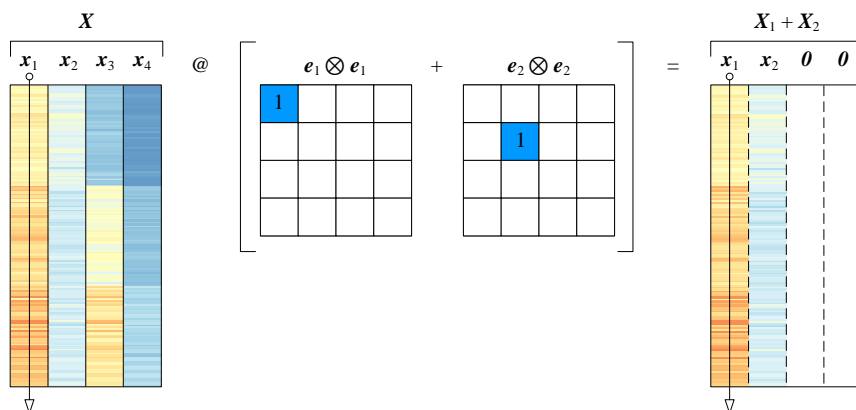
图 35. 四特征数据矩阵 $X_{150 \times 4}$ 向 $[e_1, e_2]$ 方向标量投影

图 36 所示为 X 向 $[e_1, e_2]$ 方向向量投影，结果相当于图 31 和图 32 结果“叠加”，即 $X_1 + X_2$ 。很明显， $X_1 + X_2$ 并没有还原 X 。

图 36. 四特征数据矩阵 $X_{150 \times 4}$ 向 $[e_1, e_2]$ 方向向量投影

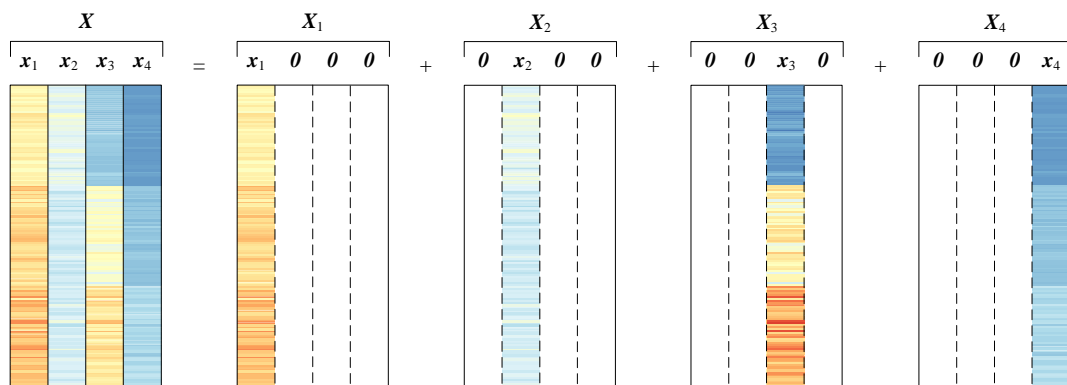
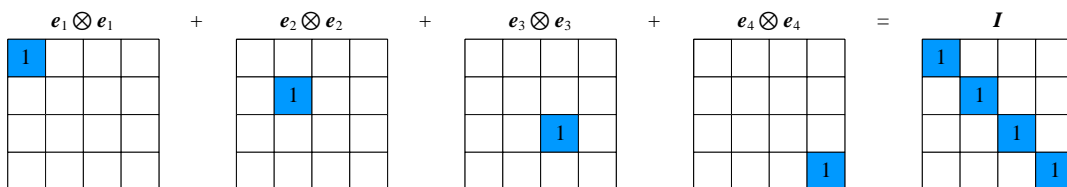
层层叠加：还原原始矩阵

上一节 (12) 告诉我们，数据矩阵 X 在规范正交基 $[v_1, v_2, \dots, v_D]$ 中每个方向上向量投影层层叠加可以完全还原原始数据。而标准正交基 $[e_1, e_2, \dots, e_D]$ 可以视作特殊的规范正交基。

图 37 告诉我们，要想完整还原 X ，需要图 31、图 32、图 33、图 34 四副热图叠加，即 $X_1 + X_2 + X_3 + X_4$ 。显然， X_1 、 X_2 、 X_3 、 X_4 这四个矩阵的秩都是 1。

图 38 是张量积的层层叠加，它是数据还原的另外一个侧面。如图 38 所示，这四个张量积相加得到单位矩阵，即：

$$e_1 \otimes e_1 + e_2 \otimes e_2 + e_3 \otimes e_3 + e_4 \otimes e_4 = I \quad (46)$$

图 37. 投影数据矩阵的层层叠加还原数据矩阵 $X_{150 \times 4}$ 图 38. 张量积的层层叠加还原 4×4 单位矩阵

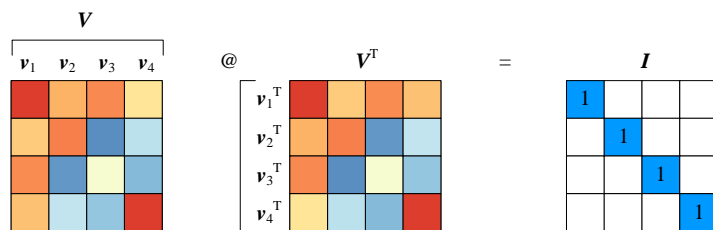
10.6 四维数据投影：规范正交基

有了上一节内容作为基础，这一节提高难度，我们用一个规范正交基重复上一节所有计算。

我们恰好找到了一个 4×4 规范正交基 V ，具体如下：

$$V = [v_1 \quad v_2 \quad v_3 \quad v_4] = \begin{bmatrix} 0.751 & 0.284 & 0.502 & 0.321 \\ 0.380 & 0.547 & -0.675 & -0.317 \\ 0.513 & -0.709 & -0.059 & -0.481 \\ 0.168 & -0.344 & -0.537 & 0.752 \end{bmatrix} \quad (47)$$

图 39 所示为规范正交基 V 乘其转置 V^T 得到单位矩阵。大家可以自己试着验算上式是否满足 $VV^T = I$ ，即 V 每一列列向量都是单位向量，且 V 的列向量两两正交。

图 39. 规范正交基 V 乘其转置得到 4×4 单位矩阵

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

V 中的像

如图 40 所示，直接将 X 投影到规范正交基 V ，得到 Z 。 Z 就是 X 在 V 中的像，根据 $Xv_j = z_j$ ，下面我们逐一分析矩阵 Z 的列向量。

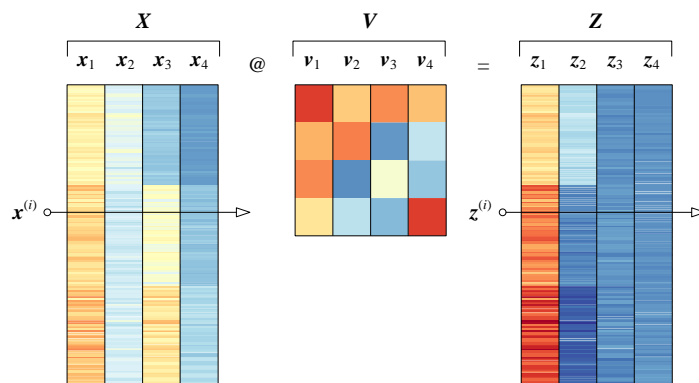


图 40. 四特征数据矩阵 $X_{150 \times 4}$ 投影到规范正交基 V 得到 Z

第 1 列向量 v_1

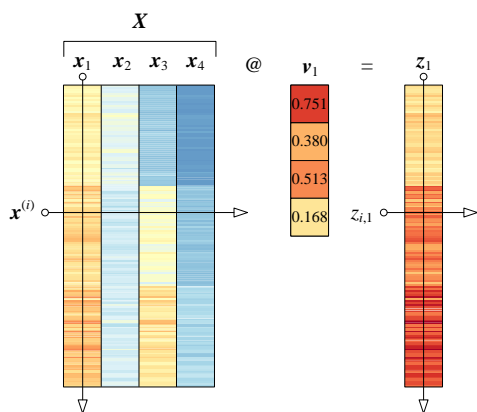
图 41 所示为鸢尾花数据集矩阵 X 在 v_1 方向上标量投影的运算热图。

从行向量角度来看， $x^{(i)}v_1 \rightarrow z_{i,1}$ 代表 \mathbb{R}^D 空间坐标值 $x^{(i)}$ ，投影到 $\text{span}(v_1)$ 这个子空间后，得到的坐标值变成 $z_{i,1}$ 。再次强调， $z_{i,1}$ 是 $x^{(i)}$ 在 $\text{span}(v_1)$ 的坐标值。

从列向量角度来看， $[x_1, x_2, x_3, x_4]v_1 \rightarrow z_1$ ，是一个线性组合过程，即：

$$z_1 = Xv_1 = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} \begin{bmatrix} 0.751 \\ 0.380 \\ 0.513 \\ 0.168 \end{bmatrix} = 0.751x_1 + 0.380x_2 + 0.513x_3 + 0.168x_4 \quad (48)$$

上式说明，0.7512 的 x_1 、0.380 倍的 x_2 、0.513 倍的 x_3 、0.168 倍的 x_4 合成得到了一个向量 z_1 。

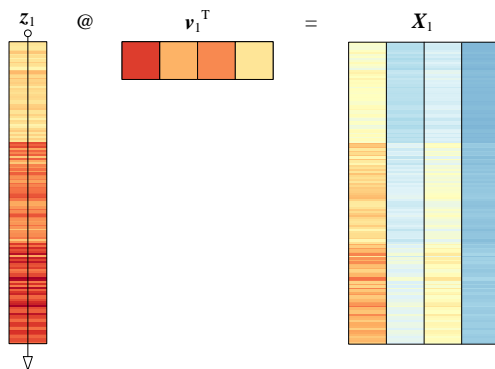
图 41. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_1 方向标量投影，一次投影

如图 42 所示， z_1 再乘 v_1^T ，便得到 X_1 ：

$$X_1 = z_1 v_1^T = z_1 [0.751 \quad 0.380 \quad 0.513 \quad 0.168] = [0.751z_1 \quad 0.380z_1 \quad 0.513z_1 \quad 0.168z_1] \quad (49)$$

很容易发现， X_1 的每一列都是 z_1 乘一个标量系数。显然， X_1 的秩为 1，即 $\text{rank}(X_1) = 1$ 。

总结来说，图 41 和图 42 用了两步完成了“二次投影”，即向量投影。

图 42. 四特征数据矩阵 $X_{150 \times 4}$ 乘 v_1^T 得到 X_1

下面，我们用向量张量积的方法，完成同样的计算。

首先计算张量积 $v_1 \otimes v_1$ ：

$$v_1 \otimes v_1 = v_1 v_1^T = \begin{bmatrix} 0.751 \\ 0.380 \\ 0.513 \\ 0.168 \end{bmatrix} @ \begin{bmatrix} 0.751 & 0.380 & 0.513 & 0.168 \end{bmatrix}^T = \begin{bmatrix} 0.564 & 0.285 & 0.385 & 0.126 \\ 0.285 & 0.144 & 0.194 & 0.063 \\ 0.385 & 0.194 & 0.263 & 0.086 \\ 0.126 & 0.063 & 0.086 & 0.028 \end{bmatrix} \quad (50)$$

图 43 所示为上述运算热图。注意，上式仅仅保留小数点后 3 位数值。很容易发现，张量积为对称矩阵。请大家自行计算，张量积的秩是否为 1。

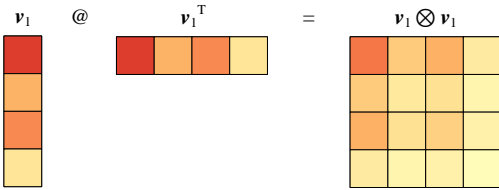


图 43. 计算张量积 $v_1 \otimes v_1$

图 44 所示为 X 和张量积 $v_1 \otimes v_1$ 乘积，几何视角，即 X 向 v_1 方向向量投影得到 X_1 ，即所谓“二次投影”。

请大家特别注意一点， X 和 X_1 在热图上已经非常接近。这是因为我们在设定 v_1 时，有特殊的“讲究”。我们将会在本书下一个板块——矩阵分解，和大家深入探讨。

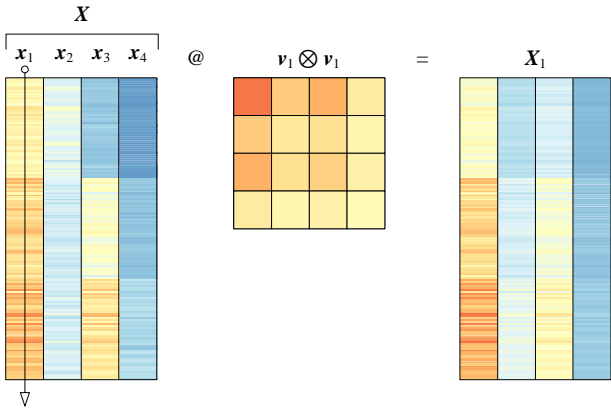


图 44. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_1 方向向量投影，二次投影

第 2 列向量 v_2

图 45 和图 46 分别所示为获得 z_2 和 X_2 的过程。请大家根据之前分析 v_1 的思路自行分析这两图。

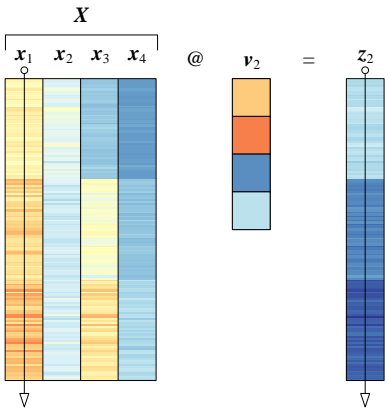


图 45. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_2 投影，一次投影

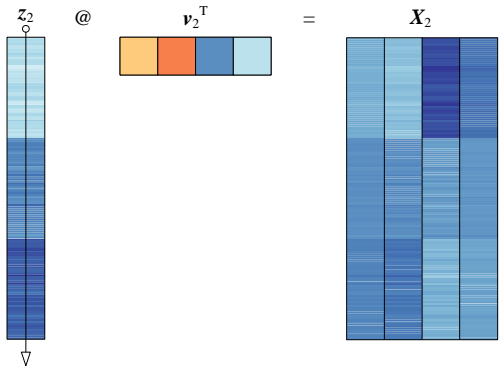


图 46. z_2 乘 v_2^T 得到 X_2

大家自行计算张量积 $v_2 \otimes v_2$ 具体值，按照前文思路分析图 47 和图 48。有必要指出一点，相比 X_1 ， X_2 热图和 X 相差很大。

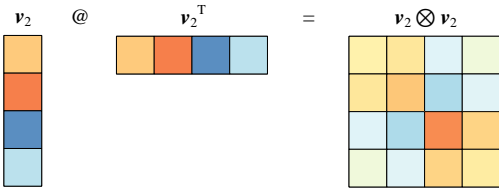


图 47. 计算张量积 $v_2 \otimes v_2$

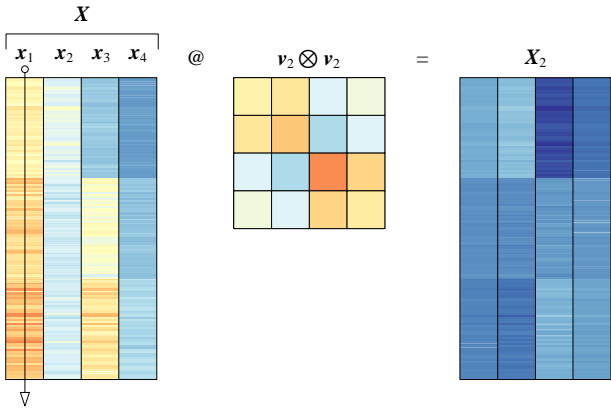


图 48. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_2 投影，二次投影

第 3 列向量 v_3

大家自行分析图 49、图 50、图 51、图 52 这四幅图。

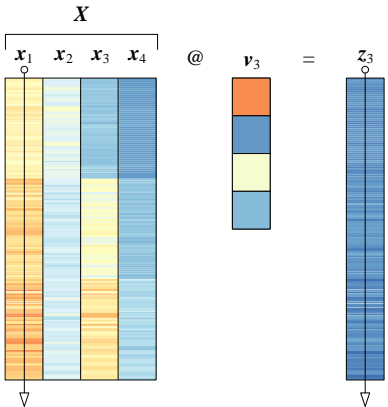


图 49. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_3 投影，一次投影

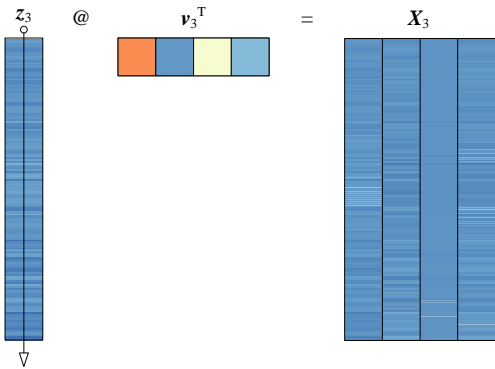


图 50. z_3 乘 v_3^T 得到 X_3

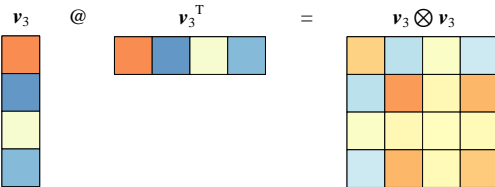


图 51. 计算张量积 $v_3 \otimes v_3$

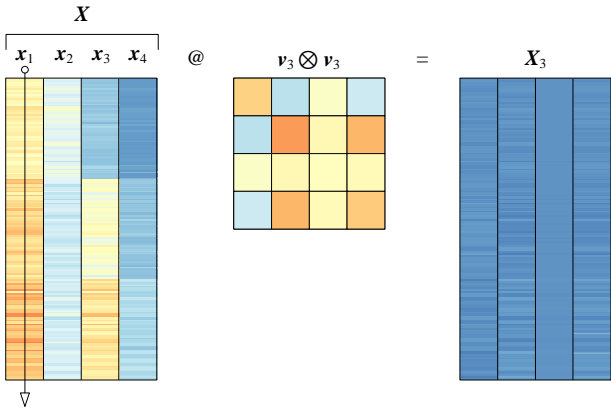


图 52. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_3 投影，二次投影

第 4 列向量 v_4

大家自行分析图 53、图 54、图 55、图 56 这四幅图。

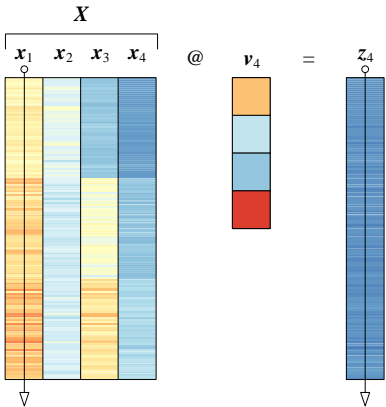


图 53. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_4 投影，一次投影

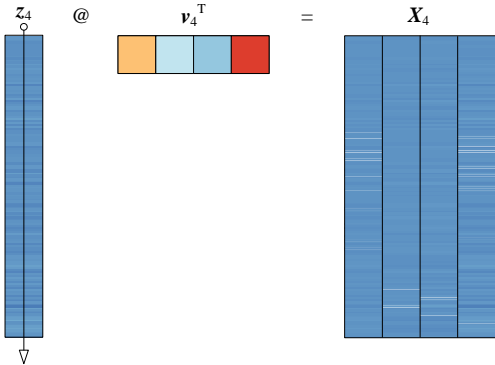


图 54. z_4 乘 v_4^T 得到 X_4

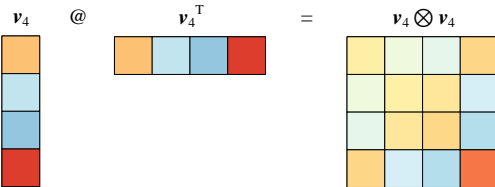


图 55. 计算张量积 $v_4 @ v_4$

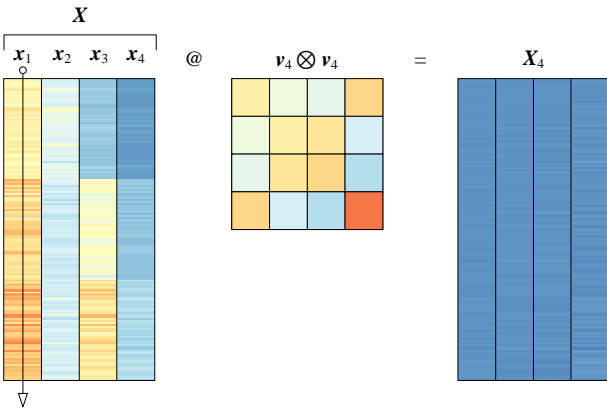


图 56. 四特征数据矩阵 $X_{150 \times 4}$ 向 v_4 投影，二次投影

层层叠加

类似前文，我们也从两个视角讨论层层叠加还原原矩阵。

如图 57 所示，数据矩阵 X 在规范正交基 $[v_1, v_2, \dots, v_D]$ 中每个方向上向量投影层层叠加可以完全还原原始数据。

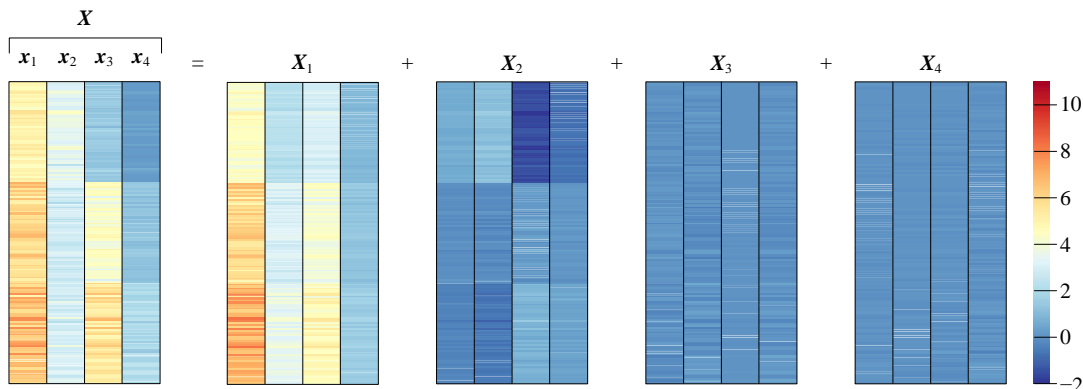


图 57. 层层叠加还原四特征数据矩阵 $X_{150 \times 4}$

图 57 告诉我们，要想完整还原 X ，需要四副热图叠加，即 $X_1 + X_2 + X_3 + X_4$ 。我们已经很清楚 X_1 、 X_2 、 X_3 、 X_4 这四个矩阵的秩都是 1，而 X 就是这四个秩为 1 的不同矩阵层层叠加之和。

但是，如前文已经指出的， X_1 已经非常接近 X 。也就是说，我们可以用 X_1 近似 X 。

特别考虑到 X_1 的秩为 1，即 $\text{rank}(X) = 1$ 。也就是说 X_1 的四个列向量之间存在倍数关系，即，

$$X_1 = z_1 v_1^T = z_1 \begin{bmatrix} 0.751 & 0.380 & 0.513 & 0.168 \end{bmatrix} = \begin{bmatrix} 0.751z_1 & 0.380z_1 & 0.513z_1 & 0.168z_1 \end{bmatrix} \quad (51)$$

X_2 、 X_3 、 X_4 各自的列向量也存在一样的关系。

此外，建议大家仔细对比图 57 中 X 、 X_1 、 X_2 、 X_3 、 X_4 这五幅热图，它们采用完全相同的色谱。

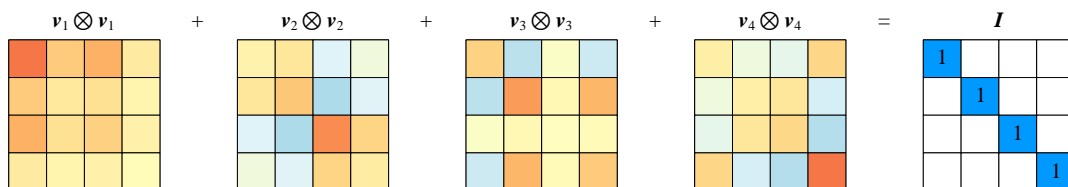


图 58. 张量积层层累加获得 4×4 单位矩阵

图 58 是张量积的层层叠加，如前文所述，它是数据还原的另外一个侧面。如图 58 所示，这四个张量积相加得到单位矩阵，即：

$$\mathbf{v}_1 \otimes \mathbf{v}_1 + \mathbf{v}_2 \otimes \mathbf{v}_2 + \mathbf{v}_3 \otimes \mathbf{v}_3 + \mathbf{v}_4 \otimes \mathbf{v}_4 = \mathbf{I} \quad (52)$$

本章前文就提到 (9)，也就是一个矩阵乘单位矩阵，结果为其本身，即 $\mathbf{X}\mathbf{I} = \mathbf{X}$ 。而单位矩阵 \mathbf{I} 可以按 (46) 分解。这也就是说，张量积层层叠加得到了单位矩阵 \mathbf{I} ，等价于还原原始数据。

数据正交化

本节最后再回过头来分析图 40 中数据矩阵 \mathbf{Z} 。如图 59 所示，矩阵 \mathbf{Z} 转置乘自身得到 \mathbf{Z} 的格拉姆矩阵：

$$\mathbf{Z}^T \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_D^T \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_D \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \mathbf{z}_1 & \mathbf{z}_1^T \mathbf{z}_2 & \cdots & \mathbf{z}_1^T \mathbf{z}_D \\ \mathbf{z}_2^T \mathbf{z}_1 & \mathbf{z}_2^T \mathbf{z}_2 & \cdots & \mathbf{z}_2^T \mathbf{z}_D \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_D^T \mathbf{z}_1 & \mathbf{z}_D^T \mathbf{z}_2 & \cdots & \mathbf{z}_D^T \mathbf{z}_D \end{bmatrix} \quad (53)$$

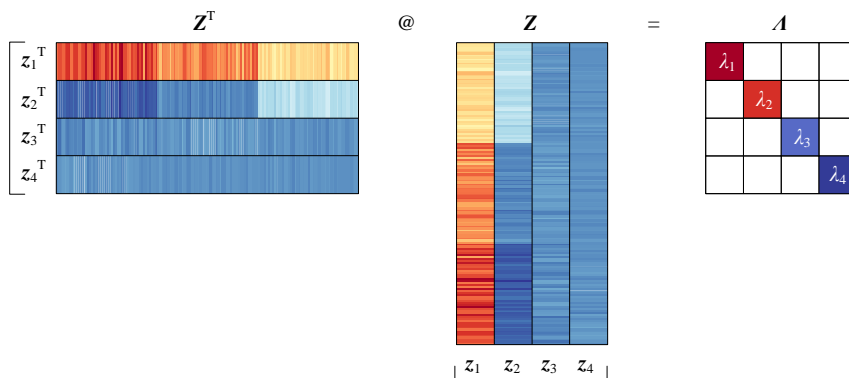


图 59. 矩阵 \mathbf{Z} 的格拉姆矩阵

(53) 写成向量内积形式：

$$\mathbf{Z}^T \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 \cdot \mathbf{z}_1 & \mathbf{z}_1 \cdot \mathbf{z}_2 & \cdots & \mathbf{z}_1 \cdot \mathbf{z}_D \\ \mathbf{z}_2 \cdot \mathbf{z}_1 & \mathbf{z}_2 \cdot \mathbf{z}_2 & \cdots & \mathbf{z}_2 \cdot \mathbf{z}_D \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_D \cdot \mathbf{z}_1 & \mathbf{z}_D \cdot \mathbf{z}_2 & \cdots & \mathbf{z}_D \cdot \mathbf{z}_D \end{bmatrix} = \begin{bmatrix} \langle \mathbf{z}_1, \mathbf{z}_1 \rangle & \langle \mathbf{z}_1, \mathbf{z}_2 \rangle & \cdots & \langle \mathbf{z}_1, \mathbf{z}_D \rangle \\ \langle \mathbf{z}_2, \mathbf{z}_1 \rangle & \langle \mathbf{z}_2, \mathbf{z}_2 \rangle & \cdots & \langle \mathbf{z}_2, \mathbf{z}_D \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{z}_D, \mathbf{z}_1 \rangle & \langle \mathbf{z}_D, \mathbf{z}_2 \rangle & \cdots & \langle \mathbf{z}_D, \mathbf{z}_D \rangle \end{bmatrix} \quad (54)$$

观察图 59，发现 $\mathbf{Z}^T \mathbf{Z}$ 的结果而恰好对角方阵，即：

$$\mathbf{Z}^T \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 \cdot \mathbf{z}_1 & \mathbf{z}_1 \cdot \mathbf{z}_2 & \cdots & \mathbf{z}_1 \cdot \mathbf{z}_D \\ \mathbf{z}_2 \cdot \mathbf{z}_1 & \mathbf{z}_2 \cdot \mathbf{z}_2 & \cdots & \mathbf{z}_2 \cdot \mathbf{z}_D \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_D \cdot \mathbf{z}_1 & \mathbf{z}_D \cdot \mathbf{z}_2 & \cdots & \mathbf{z}_D \cdot \mathbf{z}_D \end{bmatrix} = \begin{bmatrix} \langle \mathbf{z}_1, \mathbf{z}_1 \rangle & \langle \mathbf{z}_1, \mathbf{z}_2 \rangle & \cdots & \langle \mathbf{z}_1, \mathbf{z}_D \rangle \\ \langle \mathbf{z}_2, \mathbf{z}_1 \rangle & \langle \mathbf{z}_2, \mathbf{z}_2 \rangle & \cdots & \langle \mathbf{z}_2, \mathbf{z}_D \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{z}_D, \mathbf{z}_1 \rangle & \langle \mathbf{z}_D, \mathbf{z}_2 \rangle & \cdots & \langle \mathbf{z}_D, \mathbf{z}_D \rangle \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_D \end{bmatrix} = \mathbf{A} \quad (55)$$

这说明, \mathbf{Z} 的列向量两两正交, 即:

$$\mathbf{z}_i^T \mathbf{z}_j = \mathbf{z}_j^T \mathbf{z}_i = \mathbf{z}_i \cdot \mathbf{z}_j = \mathbf{z}_j \cdot \mathbf{z}_i = \langle \mathbf{z}_i, \mathbf{z}_j \rangle = \langle \mathbf{z}_j, \mathbf{z}_i \rangle = 0, \quad i \neq j \quad (56)$$

像 \mathbf{Z} 这样具有这种正交性 (orthogonality) 的数据应用场合很多, 因此我们再深究一步。

将 $\mathbf{Z} = \mathbf{XV}$ 其代入 (55) 得到:

$$\mathbf{Z}^T \mathbf{Z} = (\mathbf{XV})^T \mathbf{XV} = \mathbf{V}^T \mathbf{X}^T \mathbf{XV} = \mathbf{V}^T \mathbf{GV} = \mathbf{A} \quad (57)$$

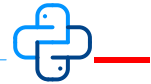
也就是说, 对于鸢尾花数据矩阵 \mathbf{X} 来说, (47) 中给出的这个 \mathbf{V} 可谓“万里挑一”, 不是一个寻常的规范正交基!

再进一步, 由于 \mathbf{V} 为规范正交基, 因此 $\mathbf{V}^T \mathbf{V} = \mathbf{I}$, 根据 (57) 等式关系, \mathbf{G} 可以写成:

$$\mathbf{G} = \mathbf{VAV}^T \quad (58)$$

这就是说 \mathbf{X} 的格拉姆矩阵 \mathbf{G} 可以分解成三个矩阵的乘积。其中, \mathbf{V} 为正交矩阵, \mathbf{A} 为对角方阵。为了获得 (58) 等式, 就需要本书下一个板块要介绍的重要线性代数工具——矩阵分解。

Bk4_Ch10_01.py 绘制本章大部分热图。



```
# Bk4_Ch10_01.py

import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import load_iris

# A copy from Seaborn
iris = load_iris()

X = iris.data
y = iris.target

feature_names = ['Sepal length, x1', 'Sepal width, x2',
                  'Petal length, x3', 'Petal width, x4']

# Convert X array to dataframe
X_df = pd.DataFrame(X, columns=feature_names)

%% Original data, X

X = X_df.to_numpy();

# Gram matrix, G and orthogonal basis V

G = X.T@X
D, V = np.linalg.eig(G)
```

本 PDF 文件为作者草稿, 发布目的为方便读者在移动终端学习, 终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有, 请勿商用, 引用请注明出处。

代码及 PDF 文件下载: <https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教, 本书专属邮箱: jiang.visualize.ml@gmail.com

```

#%%
def heatmap(Matrices, Titles, Ranges, Equal_tags):

    M1 = Matrices[0]
    M2 = Matrices[1]
    M3 = Matrices[2]

    Title_1 = Titles[0]
    Title_2 = Titles[1]
    Title_3 = Titles[2]

    fig, axs = plt.subplots(1, 5, figsize=(12, 3))

    plt.sca(axs[0])
    ax = sns.heatmap(M1, cmap='RdYlBu_r',
                     vmin = Ranges[0][0],
                     vmax = Ranges[0][1],
                     cbar=False,
                     xticklabels=False,
                     yticklabels=False)

    if Equal_tags[0] == True:
        ax.set_aspect("equal")

    plt.title(Title_1)

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

    plt.sca(axs[2])
    ax = sns.heatmap(V, cmap='RdYlBu_r',
                     vmin = Ranges[1][0],
                     vmax = Ranges[1][1],
                     cbar=False,
                     xticklabels=False,
                     yticklabels=False)

    if Equal_tags[1] == True:
        ax.set_aspect("equal")
    plt.title(Title_2)

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

    plt.sca(axs[4])
    ax = sns.heatmap(V.T, cmap='RdYlBu_r',
                     vmin = Ranges[2][0],
                     vmax = Ranges[2][1],
                     cbar=False,
                     xticklabels=False,
                     yticklabels=False)

    if Equal_tags[2] == True:
        ax.set_aspect("equal")
    plt.title(Title_3)

#%%
def plot_four_figs(X, v_j, idx):

    # Fig 1:  $X @ v_j = z_j$ 

    z_j = X @ v_j
    Titles = [' $X X^T$ ',
              ' $v_j^T$  + str(idx) + ' $v_j$ ',
              ' $z_j$  + str(idx) + ' $z_j$ ']

    Ranges = [[-2, 11],

```

```

        [-1,1],
        [-2,11]]

Equal_tags = [False,True,False]
heatmap([X,v_j,z_j],Titles,Ranges,Equal_tags)

# Fig 2:  $z@v_j.T = X_j$ 
X_j = z_j@v_j.T
Titles = ['$z_{'+str(idx)+'}',
          '$v_{'+str(idx)+'}^T$',
          '$X_{'+str(idx)+'}']

Ranges = [[-2,11],
          [-1,1],
          [-2,11]]

Equal_tags = [False,True,False]

heatmap([z_j,v_j.T,X_j],Titles,Ranges,Equal_tags)

# Fig 3:  $T_j = v_j@v_j.T$ 
T_j = v_j@v_j.T

Titles = ['$v_{'+str(idx)+'}',
          '$v_{'+str(idx)+'}^T$',
          '$T_{'+str(idx)+'}']

Ranges = [[-1,1],
          [-1,1],
          [-1,1]]

Equal_tags = [True,True,True]

heatmap([v_j,v_j.T,T_j],Titles,Ranges,Equal_tags)

# Fig 4:  $X@T_j = X_j$ 
T_j = X@T_j

Titles = ['$X$',
          '$T_{'+str(idx)+'}',
          '$X_{'+str(idx)+'}']

Ranges = [[-2,11],
          [-1,1],
          [-2,11]]

Equal_tags = [False,True,False]

heatmap([X,T_j,X_j],Titles,Ranges,Equal_tags)

### First basis vector
v1 = V[:, 0].reshape((-1, 1))
plot_four_figs(X,v1,1)

### Second basis vector
v2 = V[:, 1].reshape((-1, 1))
plot_four_figs(X,v2,2)

### Third basis vector
v3 = V[:, 2].reshape((-1, 1))
plot_four_figs(X,v3,3)

### Fourth basis vector

```

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

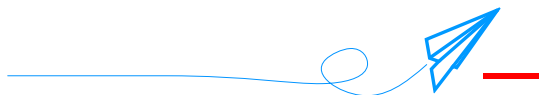
版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

```
v4 = V[:, 3].reshape((-1, 1))  
plot_four_figs(X, v4, 4)
```



本章内容是个分水岭。如果本章内容，特别是前两节内容，你读起来毫无压力，恭喜你，你可以顺利进入本书下一个板块——矩阵分解——的学习。

如果你对本章内容感觉很陌生，请回头重读前 9 章内容。

大家可能会好奇，本章中可以很好还原原始数据矩阵的 v_1 到底是怎么算出来的？其实本章代码文件已经给出了答案——特征值分解。这是本书下一个板块要讲的内容之一。

再次，强调有数据的地方，就有向量；有向量的地方，就有几何！

再加一句，有向量的地方，也会有空间。