10 Data Projection 数据投影

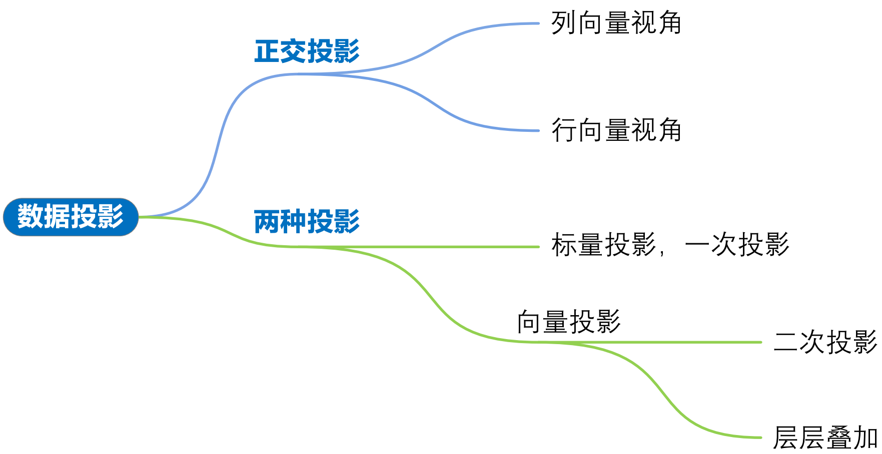
以鸢尾花数据集为例

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


数据投影
正交投影
列向量视角
行向量视角
两种投影
标量投影，一次投影
向量投影
二次投影
层层叠加

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，试图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：

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{V} \tag{1}$$

$\boldsymbol{X}$ 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，$\boldsymbol{X}$ 每一行代表一个数据点，每一列代表一个特征。

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

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

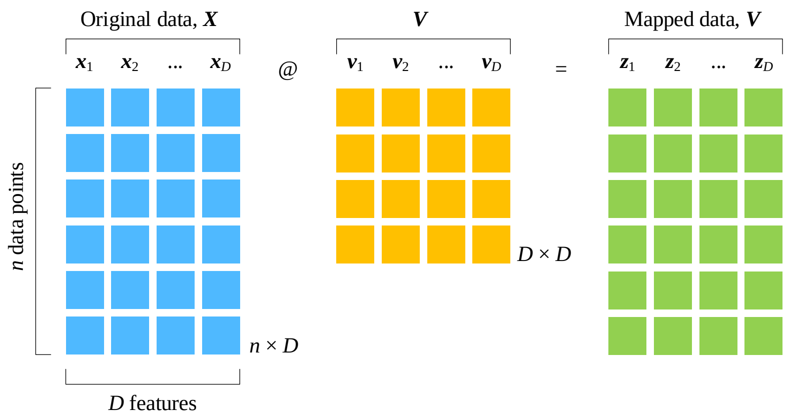


图 1. 数据矩阵 $\boldsymbol{X}$ 到 $\boldsymbol{Z}$ 线性变换

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

$$\begin{aligned}\begin{bmatrix}\boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_D\end{bmatrix} &= \boldsymbol{X}\begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_D\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{X}\boldsymbol{v}_1 & \boldsymbol{X}\boldsymbol{v}_2 & \cdots & \boldsymbol{X}\boldsymbol{v}_D\end{bmatrix}\end{aligned} \tag{2}$$

(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 $\boldsymbol{Z}$ 矩阵的第 j 列向量 $\boldsymbol{z}_j$ 的计算式：

$$\boldsymbol{z}_j = \boldsymbol{X}\boldsymbol{v}_j \tag{3}$$

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

$$\boldsymbol{z}_j = \underbrace{\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D\end{bmatrix}}_{\boldsymbol{X}}\underbrace{\begin{bmatrix}v_{1,j} \\ v_{2,j} \\ \vdots \\ v_{D,j}\end{bmatrix}}_{\boldsymbol{v}_j} = v_{1,j}\boldsymbol{x}_1 + v_{2,j}\boldsymbol{x}_2 + \cdots v_{D,j}\boldsymbol{x}_D \tag{4}$$

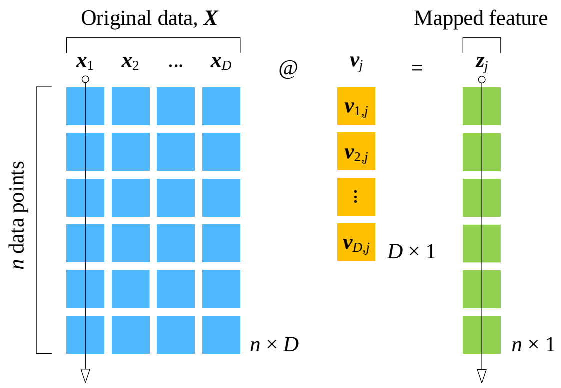


图 2. $\boldsymbol{Z}$ 第 j 列向量 $\boldsymbol{z}_j$ 的计算过程

行向量：点坐标

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

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

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

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 $\boldsymbol{x}^{(i)}$ 投影得到 $\boldsymbol{Z}$ 的第 i 行向量 $\boldsymbol{z}^{(i)}$：

$$\boldsymbol{z}^{(i)} = \boldsymbol{x}^{(i)}\boldsymbol{V} \tag{6}$$

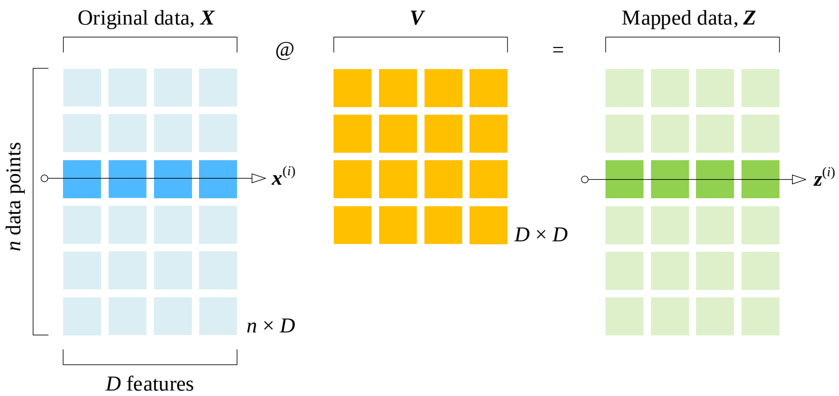


图 3. 每一行数据点之间的转换关系

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

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

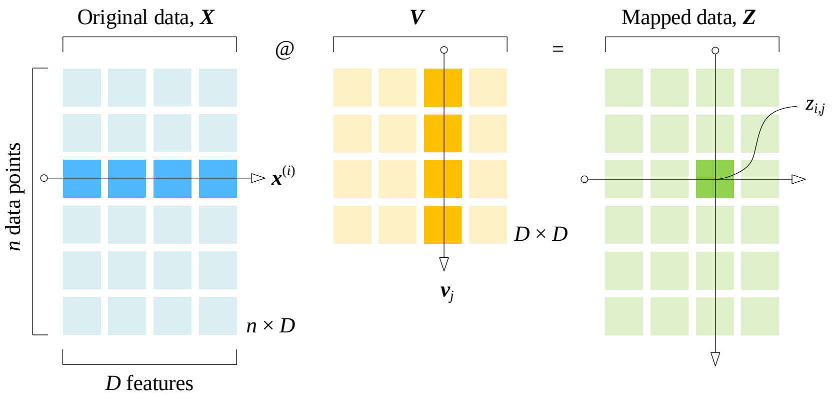


图 4. 每一行数据点向 $\boldsymbol{v}_j$ 投影

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

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

图 4 对应 (8) 运算。

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

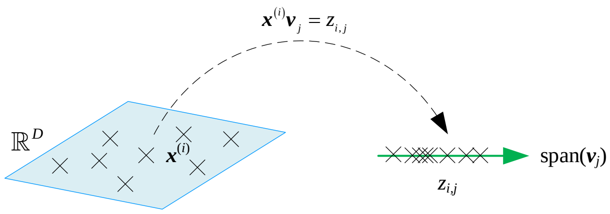


图 5. $\mathbb{R}^D$ 空间数据点投影到 $\operatorname{span}(\boldsymbol{v}_j)$

10.2 二次投影 + 层层叠加

本书前文给出下面这个看似莫明其妙的矩阵乘法：

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{I} = \boldsymbol{X}\underbrace{\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}}_{\boldsymbol{I}} = \boldsymbol{X} \tag{9}$$

也就是说，数据矩阵 $\boldsymbol{X}$ 乘以单位阵，结果为其本身！

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

$$\begin{aligned}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} &= \boldsymbol{X}\begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_D\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_1^{\mathrm{T}} \\ \boldsymbol{v}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_D^{\mathrm{T}}\end{bmatrix} \\ &= \underbrace{\boldsymbol{X}\boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{T}}}_{\boldsymbol{X}_1} + \underbrace{\boldsymbol{X}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{T}}}_{\boldsymbol{X}_2} + \cdots + \underbrace{\boldsymbol{X}\boldsymbol{v}_D\boldsymbol{v}_D^{\mathrm{T}}}_{\boldsymbol{X}_D}\end{aligned} \tag{10}$$

令，

$$\boldsymbol{X}_j = \boldsymbol{X}\boldsymbol{v}_j\boldsymbol{v}_j^{\mathrm{T}} \tag{11}$$

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

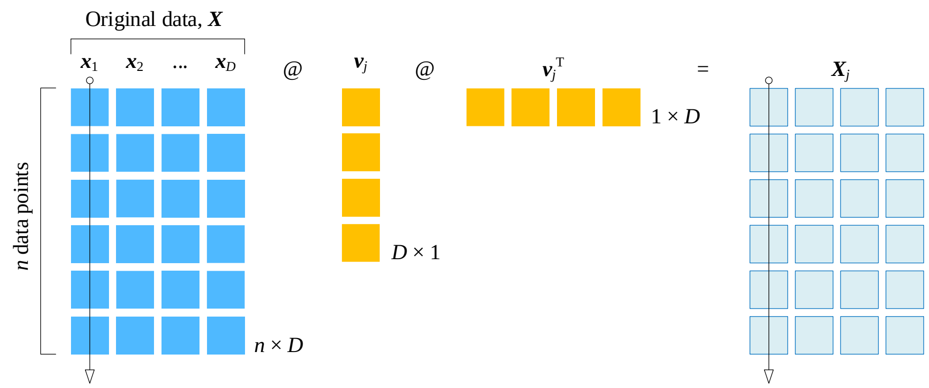


图 6. 二次投影

(10) 可以写成：

$$\boldsymbol{X} = \boldsymbol{X}_1 + \boldsymbol{X}_2 + \cdots + \boldsymbol{X}_D \tag{12}$$

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 $\boldsymbol{X}$。

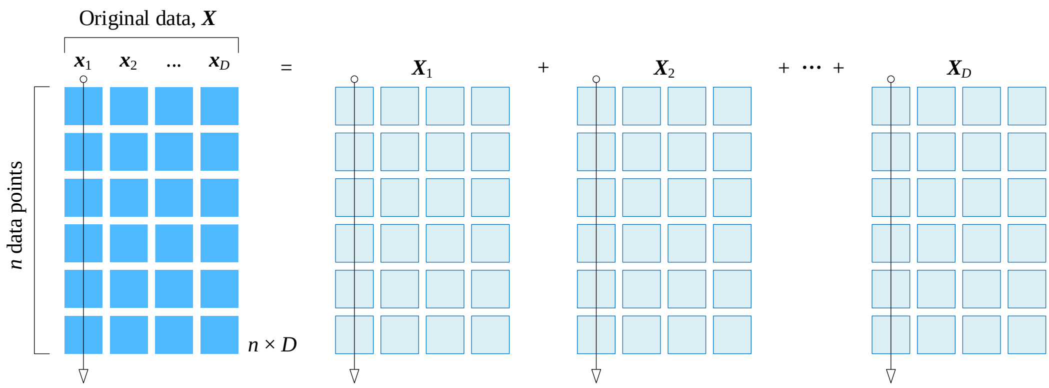


图 7. 层层叠加

二次投影

下面，我们首先聊聊"二次投影"。

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

$$\boldsymbol{x}_j^{(i)} = \underbrace{\boldsymbol{x}^{(i)}\boldsymbol{v}_j}_{z_{i,j}}\boldsymbol{v}_j^{\mathrm{T}} = z_{i,j}\boldsymbol{v}_j^{\mathrm{T}} \tag{13}$$

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

而 $z_{i,j}\boldsymbol{v}_j^{\mathrm{T}}$ 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，$\boldsymbol{x}^{(i)}$ 和 $z_{i,j}\boldsymbol{v}_j^{\mathrm{T}}$ 都用行向量表达坐标点。

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

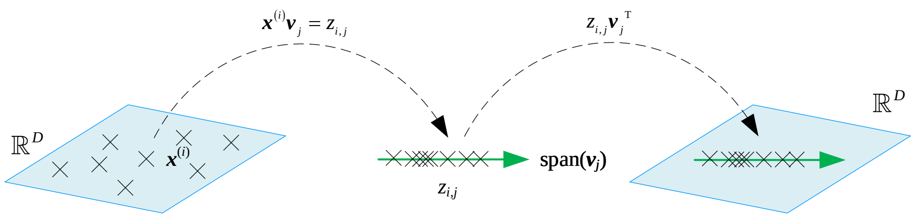


图 8. $\mathbb{R}^D$ 空间数据点先投影到 span($\boldsymbol{v}_j$)，再投影回到 $\mathbb{R}^D$

向量投影：张量积

将 (11) 写成张量积的形式：

$$\boldsymbol{X}_j = \boldsymbol{X}\boldsymbol{v}_j \otimes \boldsymbol{v}_j \tag{14}$$

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

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

很显然，

$$\operatorname{rank}\left(\boldsymbol{v}_j \otimes \boldsymbol{v}_j\right) = 1 \quad \Rightarrow \quad \operatorname{rank}\left(\boldsymbol{X}_j\right) = 1 \tag{15}$$

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

利用张量积，(10) 可以写成：

$$\boldsymbol{X} = \underbrace{\boldsymbol{X}\boldsymbol{v}_1 \otimes \boldsymbol{v}_1}_{\boldsymbol{X}_1} + \underbrace{\boldsymbol{X}\boldsymbol{v}_2 \otimes \boldsymbol{v}_2}_{\boldsymbol{X}_2} + \cdots + \underbrace{\boldsymbol{X}\boldsymbol{v}_D \otimes \boldsymbol{v}_D}_{\boldsymbol{X}_D} \tag{16}$$

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

标准正交基：便于理解

为了方便理解，我们用标准正交基 $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_D]$ 替换 (16) 中的 $[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$，得到：

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \boldsymbol{X}\boldsymbol{e}_2 \otimes \boldsymbol{e}_2 + \cdots + \boldsymbol{X}\boldsymbol{e}_D \otimes \boldsymbol{e}_D \tag{17}$$

展开上式中第一项得到：

$$\boldsymbol{X}_1 = \boldsymbol{X}\boldsymbol{e}_1 \otimes \boldsymbol{e}_1 = \boldsymbol{X}\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D\end{bmatrix}}_{\boldsymbol{X}}\begin{bmatrix}1 & & & \\ & 0 & & \\ & & 0 & \\ & & & 0\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}_1 & 0 & \cdots & 0\end{bmatrix} \tag{18}$$

$\boldsymbol{X}\boldsymbol{e}_1$ 得到的是 $\boldsymbol{X}$ 的每一行在 span($\boldsymbol{e}_1$) 这个子空间的坐标，即 $\boldsymbol{x}_1$。而 $\boldsymbol{X}\boldsymbol{e}_1 \otimes \boldsymbol{e}_1$ 告诉我们的是 $\boldsymbol{X}\boldsymbol{e}_1$ 在 D 维空间 $\mathbb{R}^D$ 中坐标值。

此后每一项 $\boldsymbol{X}_j$ 可以写成：

$$\begin{aligned}\boldsymbol{X}_2 &= \boldsymbol{X}\boldsymbol{e}_2 \otimes \boldsymbol{e}_2 = \begin{bmatrix}0 & \boldsymbol{x}_2 & \cdots & 0\end{bmatrix}\\ &\vdots \\ \boldsymbol{X}_D &= \boldsymbol{X}\boldsymbol{e}_D \otimes \boldsymbol{e}_D = \begin{bmatrix}0 & 0 & \cdots & \boldsymbol{x}_D\end{bmatrix}\end{aligned} \tag{19}$$

也就是说，这个每次计算 $\boldsymbol{X}\boldsymbol{e}_j \otimes \boldsymbol{e}_j$ 投影就是仅保留 $\boldsymbol{X}$ 的第 j 列 $\boldsymbol{x}_j$，其他元素置 0。

因此，(17) 可以写成：

$$\boldsymbol{X} = \underbrace{\begin{bmatrix}\boldsymbol{x}_1 & 0 & \cdots & 0\end{bmatrix}}_{\boldsymbol{X}_1} + \underbrace{\begin{bmatrix}0 & \boldsymbol{x}_2 & \cdots & 0\end{bmatrix}}_{\boldsymbol{X}_2} + \cdots + \underbrace{\begin{bmatrix}0 & 0 & \cdots & \boldsymbol{x}_D\end{bmatrix}}_{\boldsymbol{X}_D} \tag{20}$$

图 9 所示为上式二次投影与叠加过程。

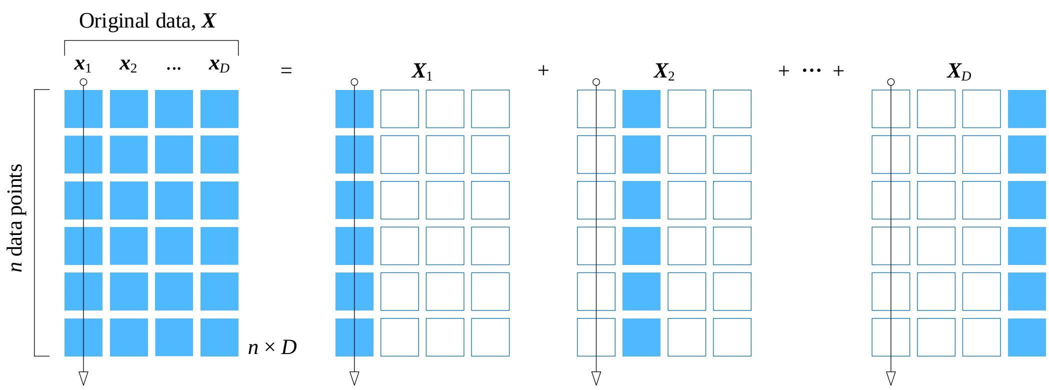


图 9. 标准正交基 [$\boldsymbol{e}_1$, $\boldsymbol{e}_2$, …, $\boldsymbol{e}_D$] 中二次投影与叠加

回过头再看 (9)，我们知道这个过程是先从标准正交基 [$\boldsymbol{e}_1$, $\boldsymbol{e}_2$, …, $\boldsymbol{e}_D$] 到规范正交基 [$\boldsymbol{v}_1$, $\boldsymbol{v}_2$, …, $\boldsymbol{v}_D$] 的投影，然后再投影回到标准正交基 [$\boldsymbol{e}_1$, $\boldsymbol{e}_2$, …, $\boldsymbol{e}_D$]：

$$\boldsymbol{X} \xrightarrow[\boldsymbol{V}]{} \boldsymbol{X}\boldsymbol{V} \xrightarrow[\boldsymbol{V}^{\mathrm{T}}]{} \boldsymbol{X} \tag{21}$$

看到这里，相信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 标准正交基

本节我们先从最简单的标准正交基 $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_D]$ 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 10 所示为鸢尾花数据集矩阵 $\boldsymbol{X}$ 在 $\boldsymbol{e}_1$ 方向上标量投影的运算热图。

从行向量角度来看，$\boldsymbol{x}^{(i)}\boldsymbol{e}_1 \to x_{i,1}$ 代表 $\mathbb{R}^D$ 空间坐标值 $\boldsymbol{x}^{(i)}$，投影到 $\text{span}(\boldsymbol{e}_1)$ 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 $\boldsymbol{x}^{(i)}$ 在 $\text{span}(\boldsymbol{e}_1)$ 的坐标值。

从列向量角度来看，$[\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4]\boldsymbol{e}_1 \to \boldsymbol{x}_1$，是一个线性组合过程。而 $\boldsymbol{e}_1 = [1, 0, 0, 0]^{\mathrm{T}}$，所以组合的结果只保留了鸢尾花数据集第一列 $\boldsymbol{x}_1$，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 11、图 12、图 13 三幅热图运算。

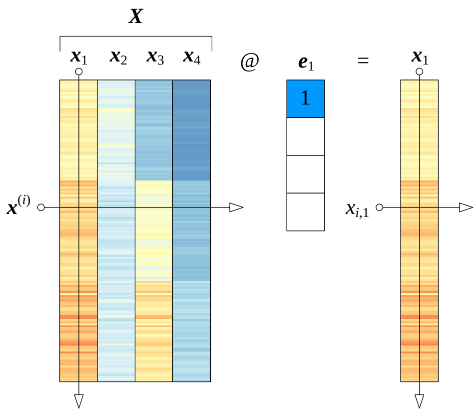


图 10. $\boldsymbol{X}$ 向 $\boldsymbol{e}_1$ 方向标量投影，一次投影

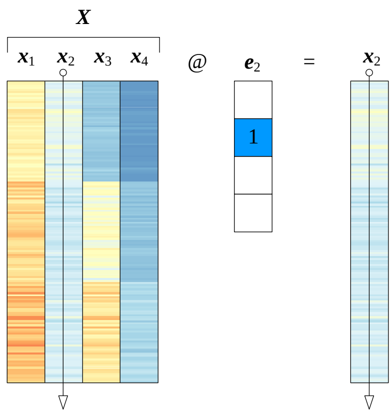


图 11. $\boldsymbol{X}$ 向 $\boldsymbol{e}_2$ 方向标量投影，一次投影

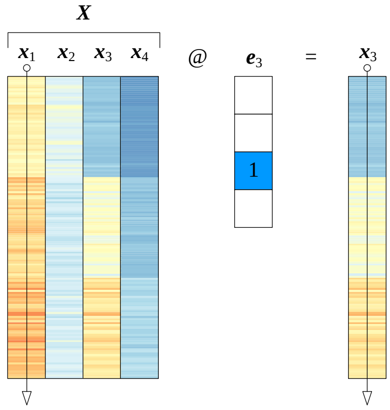


图 12. $\boldsymbol{X}$ 向 $\boldsymbol{e}_3$ 方向标量投影，一次投影

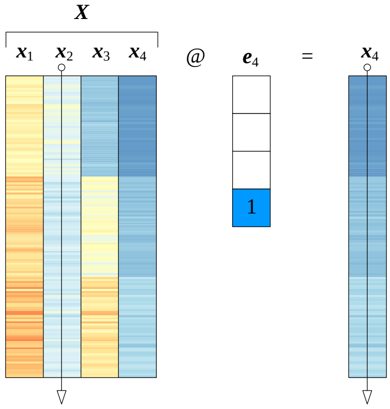


图 13. $\boldsymbol{X}$ 向 $\boldsymbol{e}_4$ 方向标量投影，一次投影

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 14 所示，$\boldsymbol{X}$ 向 $\boldsymbol{e}_1$ 方向向量投影结果就是 $\boldsymbol{X}$ 和 $\boldsymbol{e}_1 \otimes \boldsymbol{e}_1$ 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 15、图 16、图 17。

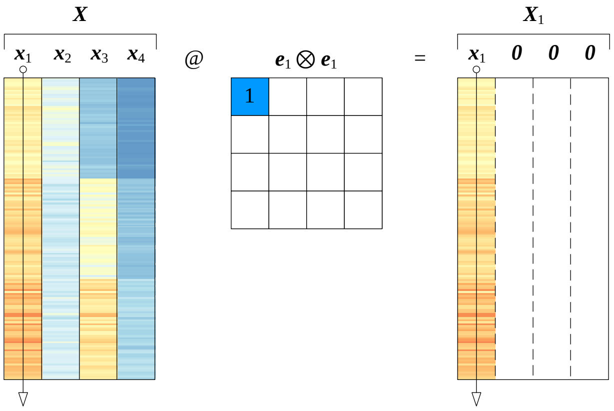


图 14. $\boldsymbol{X}$ 向 $\boldsymbol{e}_1$ 方向向量投影，二次投影

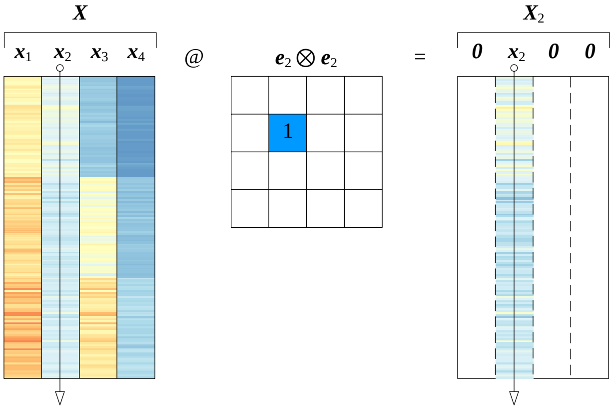


图 15. $\boldsymbol{X}$ 向 $\boldsymbol{e}_2$ 方向向量投影，二次投影

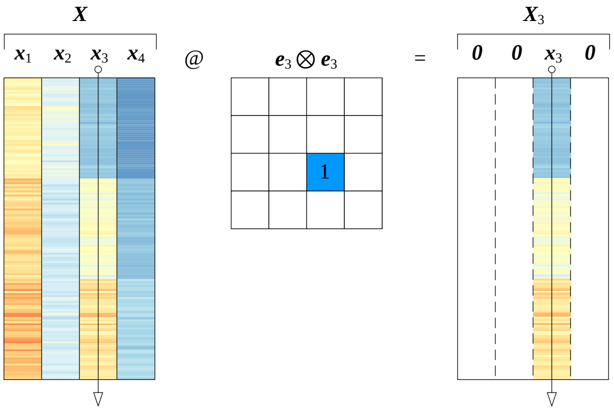


图 16. $\boldsymbol{X}$ 向 $\boldsymbol{e}_3$ 方向向量投影，二次投影

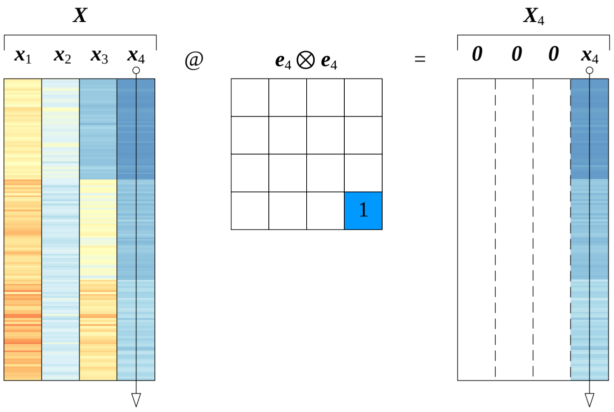


图 17. $\boldsymbol{X}$ 向 $\boldsymbol{e}_4$ 方向向量投影，二次投影

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 18 所示，$\boldsymbol{X}$ 向 $[\boldsymbol{e}_1, \boldsymbol{e}_2]$ 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。

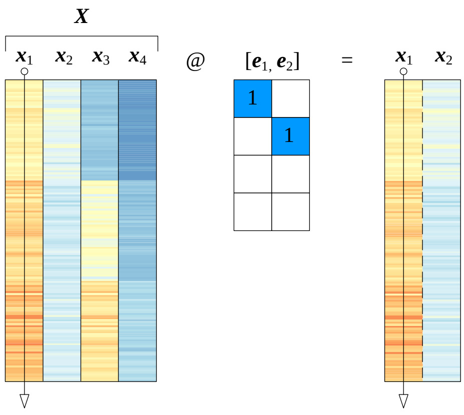


图 18. $\boldsymbol{X}$ 向 $[\boldsymbol{e}_1, \boldsymbol{e}_2]$ 方向标量投影

图 19 所示为 $\boldsymbol{X}$ 向 $[\boldsymbol{e}_1, \boldsymbol{e}_2]$ 方向向量投影，结果相当于图 14 和图 15 结果"叠加"，即 $\boldsymbol{X}_1 + \boldsymbol{X}_2$。很明显，$\boldsymbol{X}_1 + \boldsymbol{X}_2$ 并没有还原 $\boldsymbol{X}$。

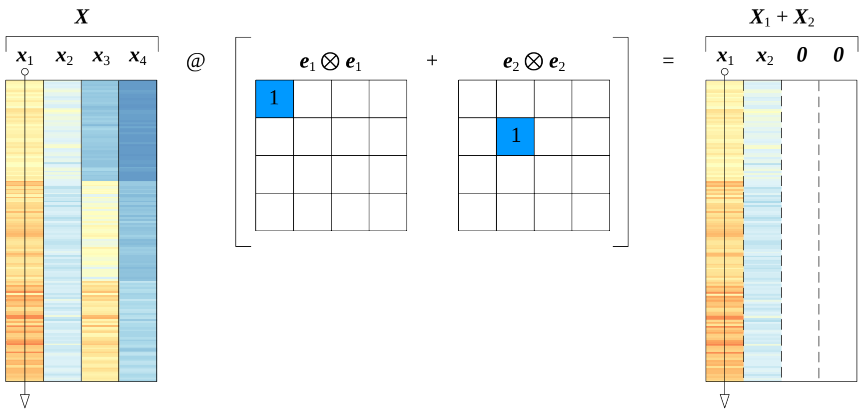


图 19. $\boldsymbol{X}$ 向 $[\boldsymbol{e}_1, \boldsymbol{e}_2]$ 方向向量投影

层层叠加：还原原始矩阵

上一节 (12) 告诉我们，数据矩阵 $\boldsymbol{X}$ 在规范正交基 $[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$ 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 $[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_D]$ 可以视作特殊的规范正交基。

图 20 告诉我们，要想完整还原 $\boldsymbol{X}$，需要图 14、图 15、图 16、图 17 四副热图叠加，即 $\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3 + \boldsymbol{X}_4$。显然，$\boldsymbol{X}_1$、$\boldsymbol{X}_2$、$\boldsymbol{X}_3$、$\boldsymbol{X}_4$ 这四个矩阵的秩都是 1。

图 21 是张量积的层层叠加，它是数据还原的另外一个侧面。如图 21 所示，这四个张量积相加得到单位矩阵，即：

$$\boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 + \boldsymbol{e}_3 \otimes \boldsymbol{e}_3 + \boldsymbol{e}_4 \otimes \boldsymbol{e}_4 = \boldsymbol{I} \tag{22}$$

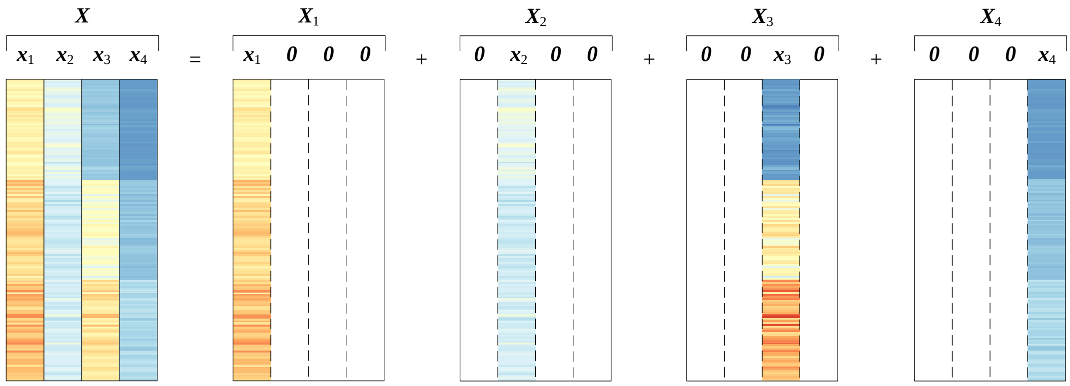


图 20. 投影数据矩阵的层层叠加

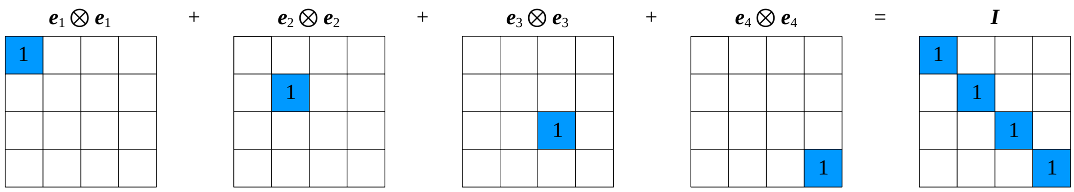


图 21. 张量积的层层叠加

10.4 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 $\boldsymbol{V}$，具体如下：

$$\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4 \end{bmatrix} = \begin{bmatrix} 0.751 & 0.284 & 0.502 & 0.321 \\ 0.380 & 0.547 & -0.675 & -0.317 \\ 0.513 & -0.709 & -0.059 & -0.481 \\ 0.168 & -0.344 & -0.537 & 0.752 \end{bmatrix} \tag{23}$$

图 22 所示为规范正交基 $\boldsymbol{V}$ 乘其转置 $\boldsymbol{V}^{\mathrm{T}}$ 得到单位矩阵。大家可以自己试着验算上式是否满足 $\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{I}$，即 $\boldsymbol{V}$ 每一列列向量都是单位向量，且 $\boldsymbol{V}$ 的列向量两两正交。

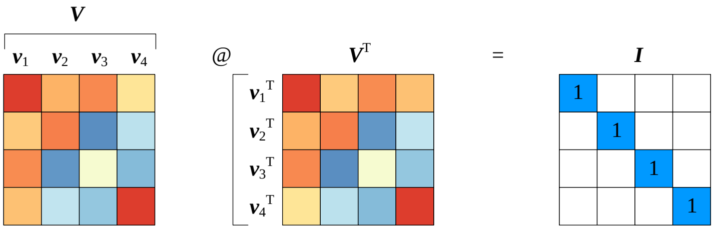


图 22. 规范正交基 $\boldsymbol{V}$ 乘其转置得到单位矩阵

$\boldsymbol{V}$ 中的像

如图 23 所示，直接将 $\boldsymbol{X}$ 投影到规范正交基 $\boldsymbol{V}$，得到 $\boldsymbol{Z}$。$\boldsymbol{Z}$ 就是 $\boldsymbol{X}$ 在 $\boldsymbol{V}$ 中的像，根据 $\boldsymbol{X}\boldsymbol{v}_j = \boldsymbol{z}_j$，下面我们逐一分析矩阵 $\boldsymbol{Z}$ 的列向量。

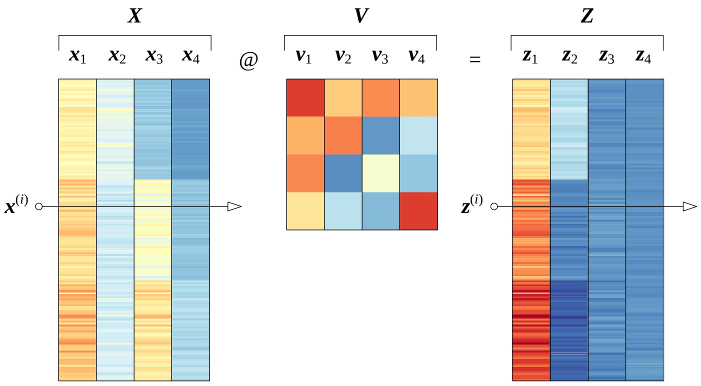


图 23. $\boldsymbol{X}$ 投影到规范正交基 $\boldsymbol{V}$ 得到 $\boldsymbol{Z}$

第 1 列向量 $\boldsymbol{v}_1$

图 24 所示为鸢尾花数据集矩阵 $\boldsymbol{X}$ 在 $\boldsymbol{v}_1$ 方向上标量投影的运算热图。

从行向量角度来看，$\boldsymbol{x}^{(i)}\boldsymbol{v}_1 \to z_{i,1}$ 代表 $\mathbb{R}^D$ 空间坐标值 $\boldsymbol{x}^{(i)}$，投影到 $\mathrm{span}(\boldsymbol{v}_1)$ 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 $\boldsymbol{x}^{(i)}$ 在 $\mathrm{span}(\boldsymbol{v}_1)$ 的坐标值。

从列向量角度来看，$[\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4]\boldsymbol{v}_1 \to \boldsymbol{z}_1$，是一个线性组合过程，即：

$$\boldsymbol{z}_1 = \boldsymbol{X}\boldsymbol{v}_1 = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \boldsymbol{x}_4 \end{bmatrix} \begin{bmatrix} 0.751 \\ 0.380 \\ 0.513 \\ 0.168 \end{bmatrix} = 0.751\boldsymbol{x}_1 + 0.380\boldsymbol{x}_2 + 0.513\boldsymbol{x}_3 + 0.168\boldsymbol{x}_4 \tag{24}$$

上式说明，0.7512 的 $\boldsymbol{x}_1$、0.380 倍的 $\boldsymbol{x}_2$、0.513 倍的 $\boldsymbol{x}_3$、0.168 倍的 $\boldsymbol{x}_4$ 合成得到了一个向量 $\boldsymbol{z}_1$。

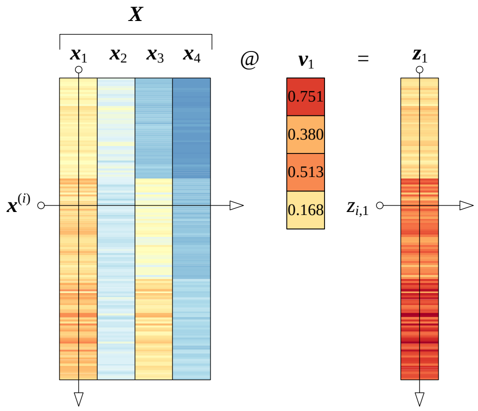


图 24. $\boldsymbol{X}$ 向 $\boldsymbol{v}_1$ 方向标量投影，一次投影

如图 25 所示，$\boldsymbol{z}_1$ 再乘 $\boldsymbol{v}_1^{\mathrm{T}}$，便得到 $\boldsymbol{X}_1$：

$$\boldsymbol{X}_1 = \boldsymbol{z}_1\boldsymbol{v}_1^{\mathrm{T}} = \boldsymbol{z}_1\begin{bmatrix}0.751 & 0.380 & 0.513 & 0.168\end{bmatrix} = \begin{bmatrix}0.751\boldsymbol{z}_1 & 0.380\boldsymbol{z}_1 & 0.513\boldsymbol{z}_1 & 0.168\boldsymbol{z}_1\end{bmatrix} \tag{25}$$

很容易发现，$\boldsymbol{X}_1$ 的每一列都是 $\boldsymbol{z}_1$ 乘一个标量系数。显然，$\boldsymbol{X}_1$ 的秩为 1，即 $\operatorname{rank}(\boldsymbol{X}_1) = 1$。

总结来说，图 24 和图 25 用了两步完成了“二次投影”，即向量投影。

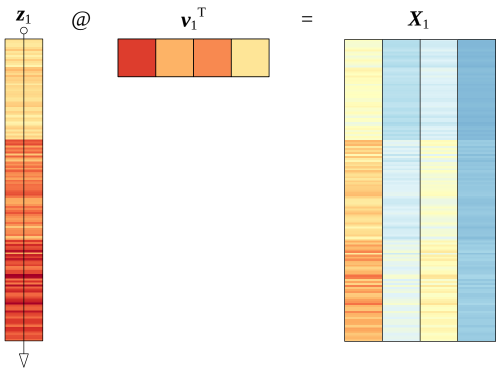


图 25. $\boldsymbol{z}_1$ 乘 $\boldsymbol{v}_1^{\mathrm{T}}$ 得到 $\boldsymbol{X}_1$

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 $\boldsymbol{v}_1 \otimes \boldsymbol{v}_1$：

$$\boldsymbol{v}_1 \otimes \boldsymbol{v}_1 = \boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{T}} = \begin{bmatrix}0.751\\0.380\\0.513\\0.168\end{bmatrix} @ \begin{bmatrix}0.751\\0.380\\0.513\\0.168\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}0.564 & 0.285 & 0.385 & 0.126\\0.285 & 0.144 & 0.194 & 0.063\\0.385 & 0.194 & 0.263 & 0.086\\0.126 & 0.063 & 0.086 & 0.028\end{bmatrix} \tag{26}$$

图 26 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。

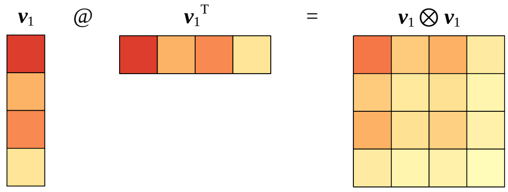


图 26. 计算张量积 $\boldsymbol{v}_1 \otimes \boldsymbol{v}_1$

图 27 所示为 $\boldsymbol{X}$ 和张量积 $\boldsymbol{v}_1 \otimes \boldsymbol{v}_1$ 乘积，几何视角，即 $\boldsymbol{X}$ 向 $\boldsymbol{v}_1$ 方向向量投影得到 $\boldsymbol{X}_1$，即所谓“二次投影”。

请大家特别注意一点，$\boldsymbol{X}$ 和 $\boldsymbol{X}_1$ 在热图上已经非常接近。这是因为我们在设定 $\boldsymbol{v}_1$ 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。

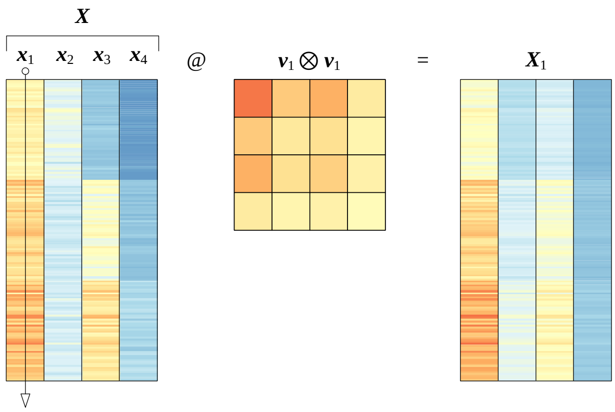


图 27. $\boldsymbol{X}$ 向 $\boldsymbol{v}_1$ 方向向量投影，二次投影

第 2 列向量 $\boldsymbol{v}_2$

图 28 和图 29 分别所示为获得 $\boldsymbol{z}_2$ 和 $\boldsymbol{X}_2$ 的过程。请大家根据之前分析 $\boldsymbol{v}_1$ 的思路自行分析这两图。

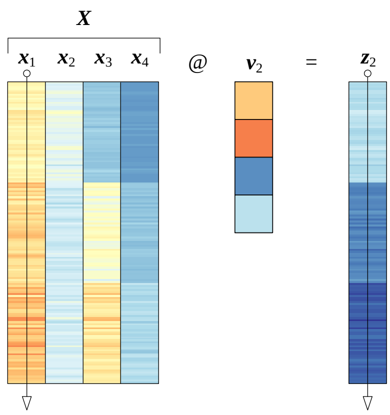


图 28. $\boldsymbol{X}$ 向 $\boldsymbol{v}_2$ 方向标量投影，一次投影

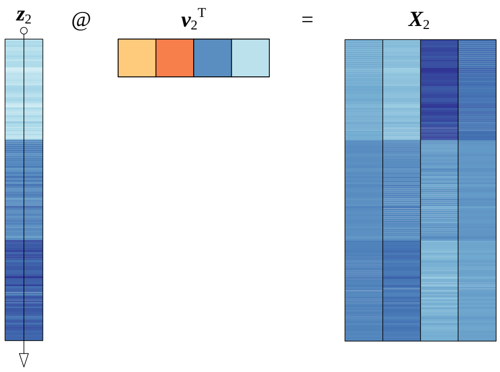


图 29. $\boldsymbol{z}_2$ 乘 $\boldsymbol{v}_2^T$ 得到 $\boldsymbol{X}_2$

大家自行计算张量积 $\boldsymbol{v}_2 \otimes \boldsymbol{v}_2$ 具体值，按照前文思路分析图 30 和图 31。有必要指出一点，相比 $\boldsymbol{X}_1$，$\boldsymbol{X}_2$ 热图和 $\boldsymbol{X}$ 相差很大。

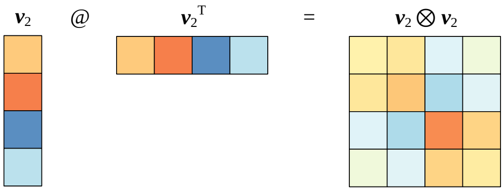


图 30. 计算张量积 $\boldsymbol{v}_2 \otimes \boldsymbol{v}_2$

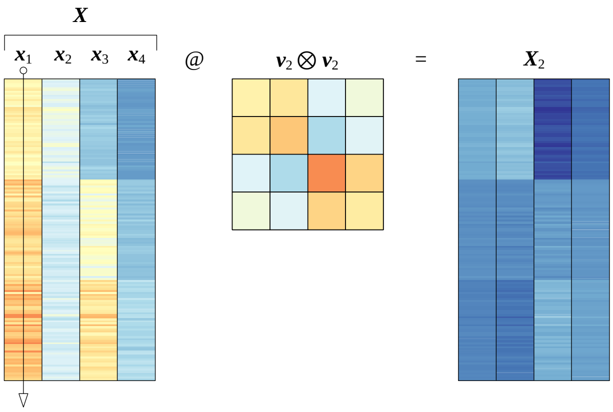


图 31. $\boldsymbol{X}$ 向 $\boldsymbol{v}_2$ 方向向量投影，二次投影

第 3 列向量 $\boldsymbol{v}_3$

大家自行分析图 32、图 33、图 34、图 35 这四幅图。

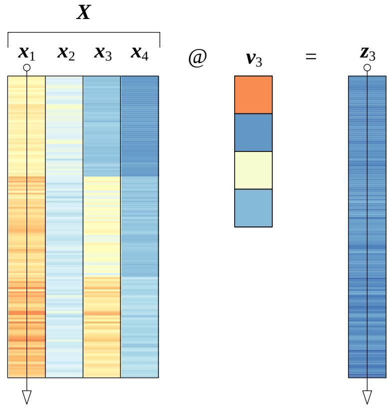


图 32. $\boldsymbol{X}$ 向 $\boldsymbol{v}_3$ 方向标量投影，一次投影

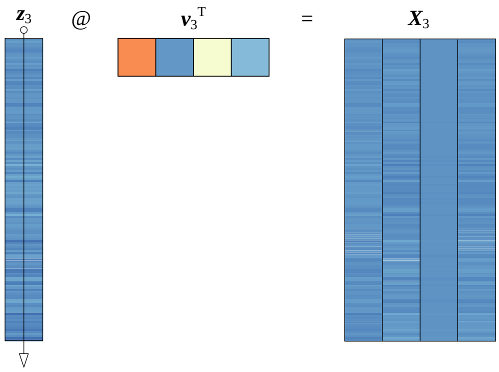


图 33. $\boldsymbol{z}_3$ 乘 $\boldsymbol{v}_3^{\mathrm{T}}$ 得到 $\boldsymbol{X}_3$

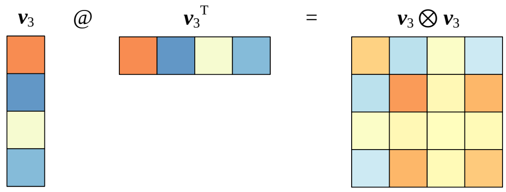


图 34. 计算张量积 $\boldsymbol{v}_3 \otimes \boldsymbol{v}_3$

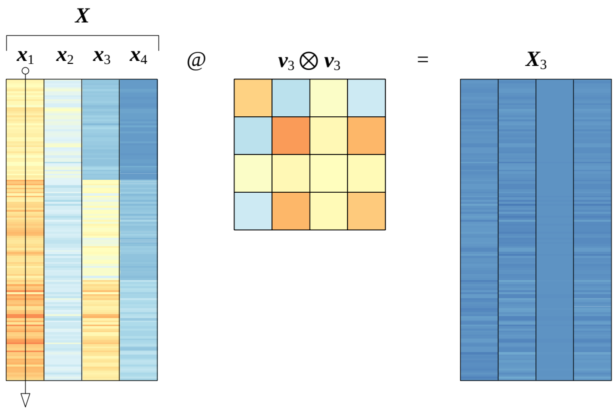


图 35. $\boldsymbol{X}$ 向 $\boldsymbol{v}_3$ 方向向量投影，二次投影

第 4 列向量 $\boldsymbol{v}_4$

大家自行分析图 36、图 37、图 38、图 39 这四幅图。

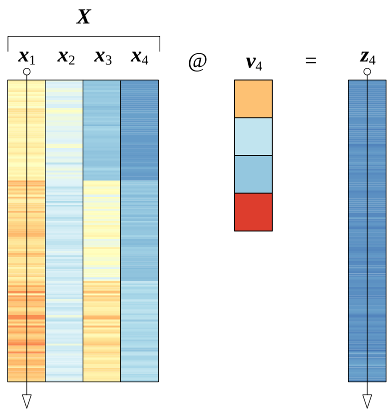


图 36. $\boldsymbol{X}$ 向 $\boldsymbol{v}_4$ 方向标量投影，一次投影

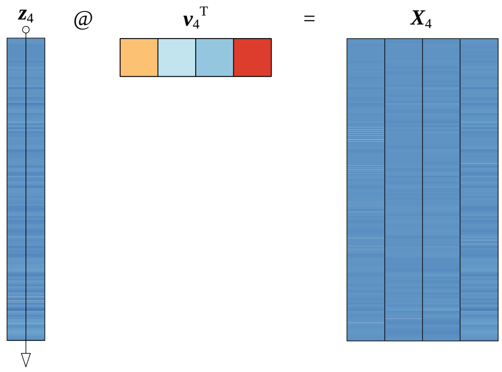


图 37. $\boldsymbol{z}_4$ 乘 $\boldsymbol{v}_4^{\mathrm{T}}$ 得到 $\boldsymbol{X}_4$

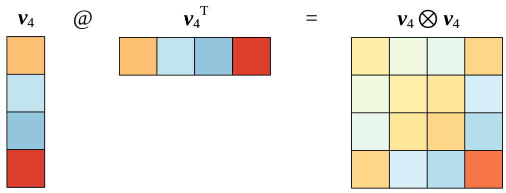


图 38. 计算张量积 $\boldsymbol{v}_4 \otimes \boldsymbol{v}_4$

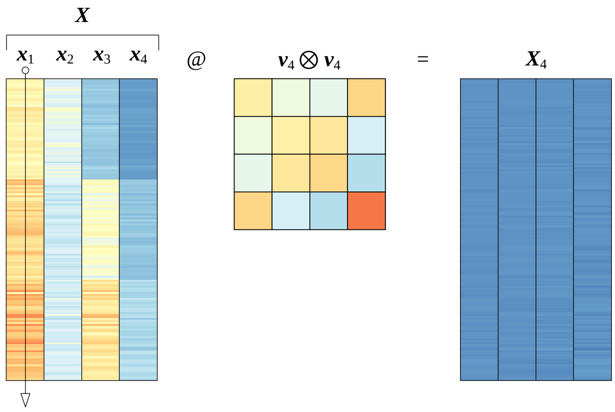


图 39. $\boldsymbol{X}$ 向 $\boldsymbol{v}_4$ 方向向量投影，二次投影

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 40 所示，数据矩阵 $\boldsymbol{X}$ 在规范正交基 $[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$ 中每个方向上向量投影层层叠加可以完全还原原始数据。

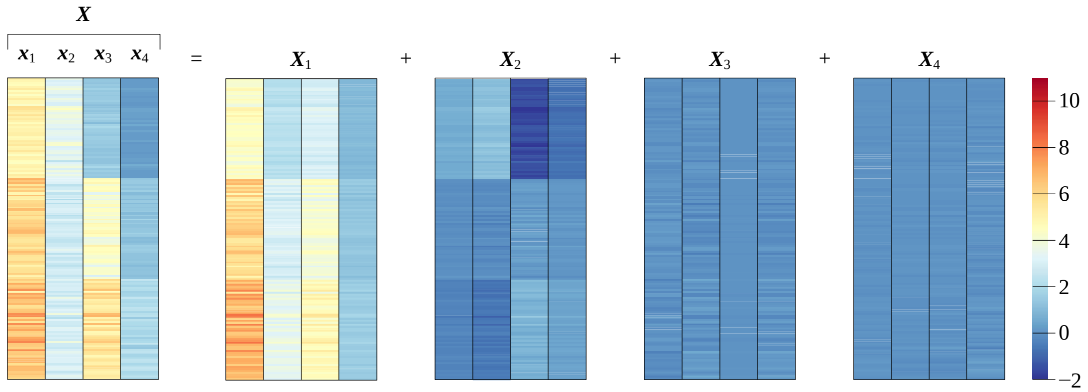


图 40. 层层叠加还原原始鸢尾花数据集矩阵

图 40 告诉我们，要想完整还原 $\boldsymbol{X}$，需要四副热图叠加，即 $\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3 + \boldsymbol{X}_4$。我们已经很清楚 $\boldsymbol{X}_1$、$\boldsymbol{X}_2$、$\boldsymbol{X}_3$、$\boldsymbol{X}_4$ 这四个矩阵的秩都是 1，而 $\boldsymbol{X}$ 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，$\boldsymbol{X}_1$ 已经非常接近 $\boldsymbol{X}$。也就是说，我们可以用 $\boldsymbol{X}_1$ 近似 $\boldsymbol{X}$。

特别考虑到 $\boldsymbol{X}_1$ 的秩为 1，即 $\text{rank}(\boldsymbol{X}) = 1$。也就是说 $\boldsymbol{X}_1$ 的四个列向量之间存在倍数关系，即，

$$\boldsymbol{X}_1 = \boldsymbol{z}_1\boldsymbol{v}_1^{\mathrm{T}} = \boldsymbol{z}_1\begin{bmatrix}0.751 & 0.380 & 0.513 & 0.168\end{bmatrix} = \begin{bmatrix}0.751\boldsymbol{z}_1 & 0.380\boldsymbol{z}_1 & 0.513\boldsymbol{z}_1 & 0.168\boldsymbol{z}_1\end{bmatrix} \tag{27}$$

$\boldsymbol{X}_2$、$\boldsymbol{X}_3$、$\boldsymbol{X}_4$ 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 40 中 $\boldsymbol{X}$、$\boldsymbol{X}_1$、$\boldsymbol{X}_2$、$\boldsymbol{X}_3$、$\boldsymbol{X}_4$ 这五幅热图，它们采用完全相同的色谱。

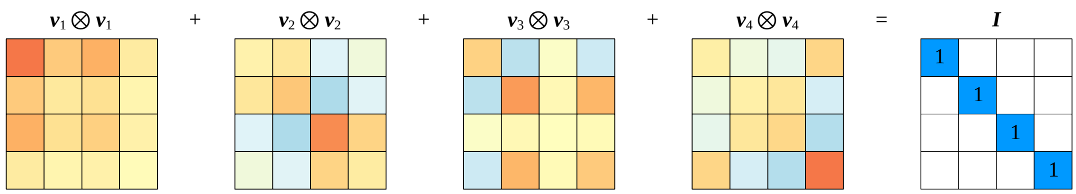


图 41. 张量积层层累加

图 41 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 41 所示，这四个张量积相加得到单位矩阵，即：

$$\boldsymbol{v}_1 \otimes \boldsymbol{v}_1 + \boldsymbol{v}_2 \otimes \boldsymbol{v}_2 + \boldsymbol{v}_3 \otimes \boldsymbol{v}_3 + \boldsymbol{v}_4 \otimes \boldsymbol{v}_4 = \boldsymbol{I} \tag{28}$$

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 $\boldsymbol{XI} = \boldsymbol{X}$。而单位矩阵 $\boldsymbol{I}$ 可以按 (22) 分解。这也就是说，张量积层层叠加得到了单位矩阵 $\boldsymbol{I}$，等价于还原原始数据。

Bk4_Ch10_01.py 绘制本章大部分热图。

```python
# 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)

#%%

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 = ['$X$',
              '$v_' + str(idx) + '$',
              '$z_' + str(idx) + '$']

    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)

#%% Third basis vector

v3 = V[:, 2].reshape((-1, 1))

plot_four_figs(X,v3)

#%% Fourth basis vector

v4 = V[:, 3].reshape((-1, 1))

plot_four_figs(X,v4)
```

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 $\boldsymbol{v}_1$ 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。