# 5 矩阵乘法

Dive into Matrix Multiplication

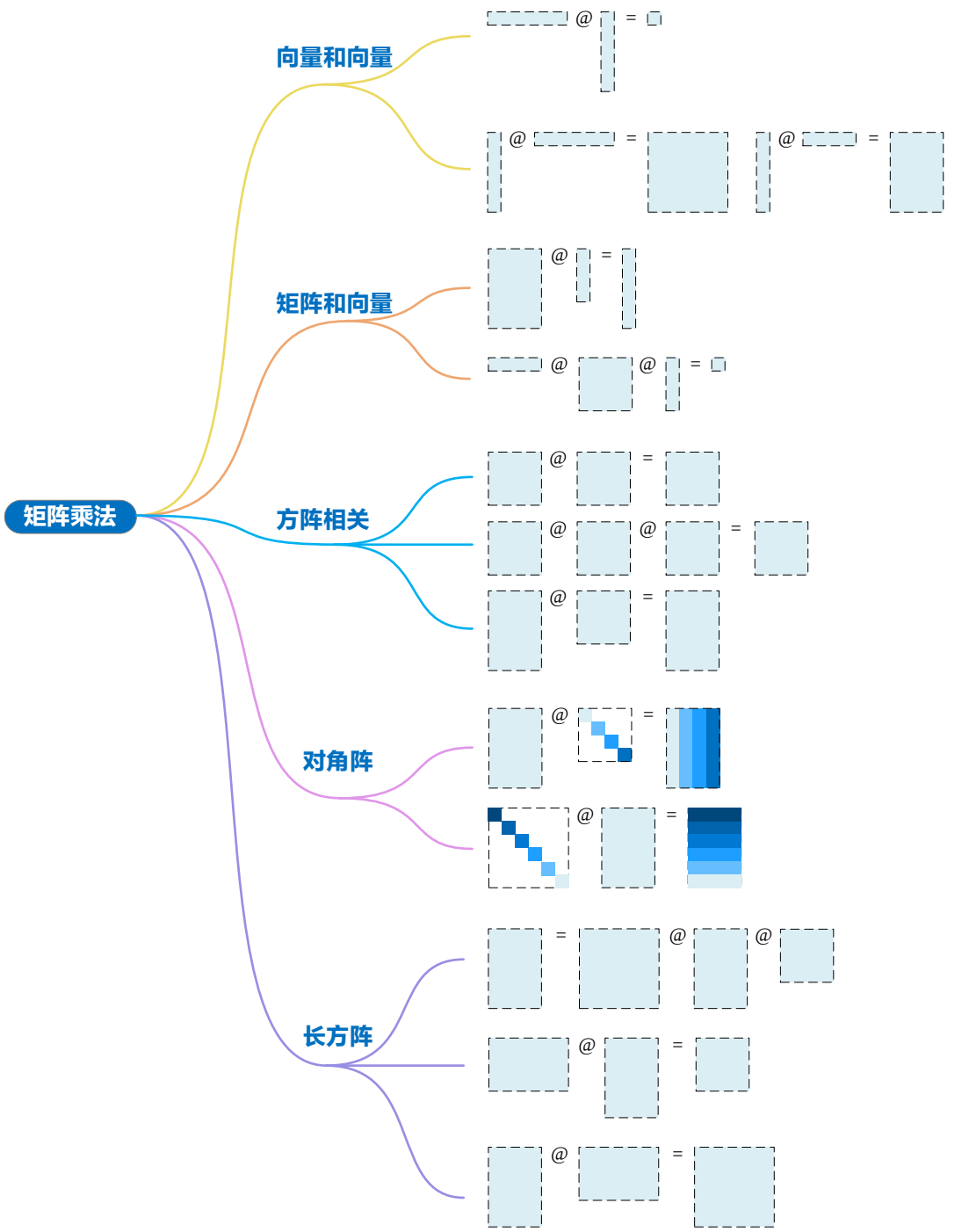
代数、几何、统计、数据交融的盛宴

只要持续进步，千万别泼冷水，哪怕蜗行牛步。

***Never discourage anyone who continually makes progress, no matter how slow.***

—— 柏拉图 (Plato) | 古希腊哲学家 | 424/423 ~ 348/347 BC

- `numpy.array()` 构造多维矩阵/数组
- `numpy.einsum()` 爱因斯坦求和约定
- `numpy.linalg.inv()` 矩阵逆运算
- `numpy.matrix()` 构造二维矩阵
- `numpy.multiply()` 矩阵逐项积
- `numpy.random.random_integers()` 生成随机整数
- `seaborn.heatmap()` 绘制数据热图



# 5.1 矩阵乘法：形态丰富多样

矩阵乘法可以说是矩阵运算中最重要的规则，没有之一！

矩阵乘法的规则本身并不难理解；但是，横在我们面前最大的困难是——矩阵乘法的灵活性。这种灵活性主要体现在矩阵乘法不同视角、矩阵乘法形态的多样性这两方面。

本书前文和大家讨论了矩阵乘法的两个视角，本书后续还将在分块矩阵中继续探讨矩阵乘法更多视角。而本章将介绍常见矩阵乘法形态。

⚠ 注意，学习本章时，请大家多从代数、几何、数据、统计几个角度理解不同矩阵乘法形态，特别是几何和数据这两个角度。

本章的作用就是鸟瞰全景，让大家开开眼界，不需要大家关注运算细节。如果你之前曾经系统学过线性代数，这一章会让你有寻他千百度、蓦然回首的感觉！作者在学习线性代数的时候，就特别希望能找到一本书能够把常见的矩阵乘法形态和应用场景都娓娓道来。

如果你刚刚接触线性代数相关内容，不要被本章大量术语吓到，大家现在不需要记住它们。本章可以视作全书重要知识点的总结。希望大家在本书不同学习阶段时，能够不断回头翻阅本章，让自己对矩阵乘法的认识一步步加深。

下面，我们就开始“鸟瞰”各种形态的矩阵乘法。

# 5.2 向量和向量

给定两个等长列向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$：

$$\boldsymbol{x}=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \boldsymbol{y}=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{1}$$

$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 向量内积可以写作 $\boldsymbol{x}$ 转置乘 $\boldsymbol{y}$，或者 $\boldsymbol{y}$ 转置乘 $\boldsymbol{x}$：

$$\boldsymbol{x}\cdot\boldsymbol{y}=\boldsymbol{y}\cdot\boldsymbol{x}=\langle\boldsymbol{x},\boldsymbol{y}\rangle=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}=\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}\right)^{\mathrm{T}}=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}=x_1y_1+x_2y_2+\cdots x_ny_n=\sum_{i=1}^{n}x_iy_i \tag{2}$$

(2) 告诉我们，$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$ 和 $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$ 相当于向量元素分别相乘，再求和，结果为标量。这和向量内积的运算结果完全一致，因此我们常用矩阵乘法替代向量内积。

观察图 1，$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$ 和 $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$ 结果均为标量，相当于 $1\times1$ 矩阵；这就是为什么 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}=(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})^{\mathrm{T}}$。

如果 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 正交 (orthogonal)，则两者向量内积为 0：

$$\boldsymbol{x}\cdot\boldsymbol{y}=\boldsymbol{y}\cdot\boldsymbol{x}=\langle\boldsymbol{x},\boldsymbol{y}\rangle=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}=\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}\right)^{\mathrm{T}}=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}=0 \tag{3}$$

正交相当于“垂直”的推广。本书中出现“正交”最多的场合就是“正交投影(orthogonal projection)”。本书第 9、10 两章专门讲解“正交投影”。此外，本书将在第 19 章深入总结正交这一概念。

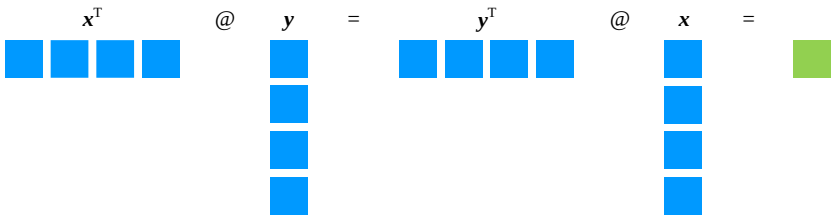



图 1. 标量积

## 全 1 列向量

全 1 列向量 $\boldsymbol{1}$ 是非常神奇的存在，多元统计离不开全 1 列向量。下面举几个例子。

如图 2 所示，全 1 列向量 $\boldsymbol{1}$ 乘行向量 $\boldsymbol{a}$，相当于对行向量 $\boldsymbol{a}$ 进行复制、向下叠放。$\boldsymbol{1}@\boldsymbol{a}$ 结果如下：

$$\boldsymbol{1}@\boldsymbol{a} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{1\times n} @\boldsymbol{a}_{m\times 1} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{a} \\ \vdots \\ \boldsymbol{a} \end{bmatrix}_{m\times n} \tag{4}$$

复制的份数取决于全 1 列向量 $\boldsymbol{1}$ 的元素个数。再次强调，上式中 $\boldsymbol{a}$ 为行向量。

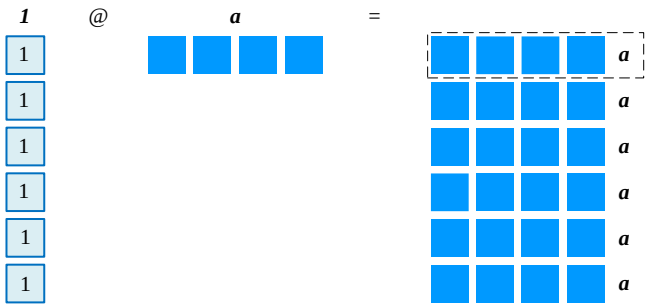



图 2. 复制行向量 $\boldsymbol{a}$

类似地，如图 3 所示，列向量 $\boldsymbol{b}$ 乘全 1 列向量 $\boldsymbol{1}$ 转置，相当于对列向量 $\boldsymbol{b}$ 复制、左右排列：

$$\boldsymbol{b}@\boldsymbol{1}^{\mathrm{T}}=\boldsymbol{b}@\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}^{\mathrm{T}}=\begin{bmatrix}\boldsymbol{b} & \boldsymbol{b} & \cdots & \boldsymbol{b}\end{bmatrix} \tag{5}$$

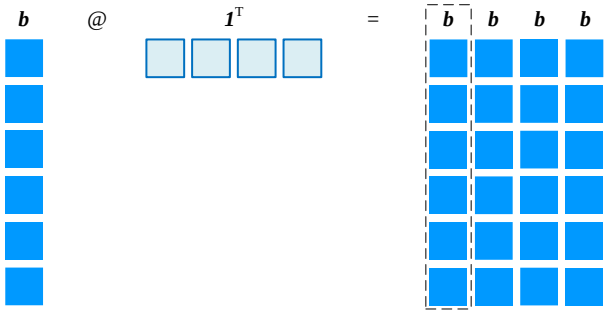



图 3. 复制列向量 $\boldsymbol{b}$

## 统计视角

下式为利用 $\boldsymbol{1}$ 对列向量 $\boldsymbol{x}$ 元素求和：

$$\boldsymbol{1}\cdot\boldsymbol{x}=\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}=x_1+x_2+\cdots x_n=\sum_{i=1}^{n}x_i \tag{6}$$

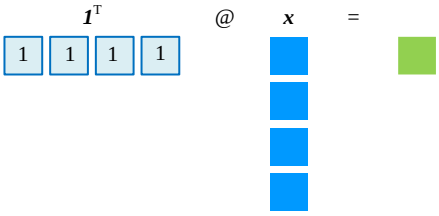



图 4. 求和运算

(6) 除以 $n$ 便是向量 $\boldsymbol{x}$ 元素平均值：

$$\mathrm{E}(\boldsymbol{x})=\frac{x_1+x_2+\cdots x_n}{n}=\frac{1}{n}\sum_{i=1}^{n}x_i=\frac{\boldsymbol{1}\cdot\boldsymbol{x}}{n}=\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}}{n}=\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}}{n} \tag{7}$$

下式为向量 $\boldsymbol{x}$ 元素各自平方后再求和：

$$\boldsymbol{x}\cdot\boldsymbol{x}=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}=x_1^2+x_2^2+\cdots x_n^2=\sum_{i=1}^{n}x_i^2 \tag{8}$$

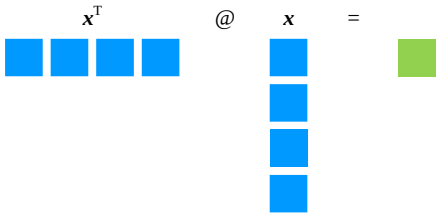



图 5. 平方和运算

计算样本方差时也用到类似 (8) 计算：

$$\operatorname{var}(X)=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2 \tag{9}$$

上式中，随机数 $X$ 的样本点构成列向量 $\boldsymbol{x}$，$\boldsymbol{x}$ 方差则为：

$$\operatorname{var}(\boldsymbol{x})=\frac{1}{n-1}\left(\boldsymbol{x}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}}{n}\right)\cdot\left(\boldsymbol{x}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}}{n}\right)=\frac{1}{n-1}\left(\boldsymbol{x}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}}{n}\right)^{\mathrm{T}}\left(\boldsymbol{x}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}}{n}\right) \tag{10}$$

本书第 22 章将讲解如何展开上式。

前文介绍过，在计算样本协方差时，我们用过类似 (2) 运算：

$$\operatorname{cov}(X,Y)=\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\mathrm{E}(X)\right)\left(y_i-\mathrm{E}(Y)\right) \tag{11}$$

⚠ 注意，如果计算总体方差、协方差的话，(9) 和 (11) 分母的 $n-1$ 则应该改为 $n$。当 $n$ 足够大，可以不区分 $n-1$ 或 $n$。

上式中，随机数 $X$ 和 $Y$ 的样本点写成列向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$，也就是说，(11) 可以写成：

$$\operatorname{cov}(\boldsymbol{x},\boldsymbol{y})=\frac{1}{n-1}\left(\boldsymbol{x}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}}{n}\right)\cdot\left(\boldsymbol{y}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{y}}{n}\right)=\frac{1}{n-1}\left(\boldsymbol{x}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x}}{n}\right)^{\mathrm{T}}\left(\boldsymbol{y}-\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{y}}{n}\right) \tag{12}$$

统计和线性代数之前有着千丝万缕的联系，本书第 22 章还会继续这一话题。

### 几何视角

如果 $\boldsymbol{x}$ 为 $n$ 维单位向量，则下两式成立：

$$\boldsymbol{x}\cdot\boldsymbol{x}=\langle\boldsymbol{x},\boldsymbol{x}\rangle=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}=\|\boldsymbol{x}\|_2^2=1,\quad\sqrt{\boldsymbol{x}\cdot\boldsymbol{x}}=\sqrt{\langle\boldsymbol{x},\boldsymbol{x}\rangle}=\sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}=\|\boldsymbol{x}\|_2=1 \tag{13}$$

整理以上不同等式都得到同一等式：

$$x_1^2+x_2^2+\cdots x_n^2=1 \tag{14}$$

几何角度，如图 6 (a) 所示，若 $n = 2$，(13) 代表平面上的单位圆 (unit circle)。如图 6 (b) 所示，若 $n = 3$，(13) 代表三维空间的单位球体 (unit sphere)。当 $n > 3$ 时，在多维空间中，(13) 代表 $n$ 维单位球面 (unit $n$-sphere) 或单位超球面或 (unit hyper-sphere)。

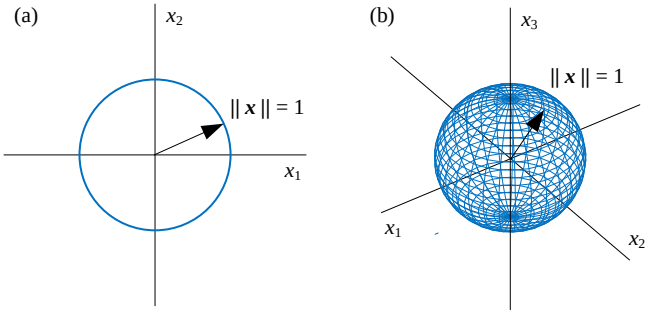



图 6. 单位圆和单位球体

单位圆、单位球、单位超球面内部的点满足：

$$\boldsymbol{x}\cdot\boldsymbol{x} = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = \|\boldsymbol{x}\|_2^2 < 1, \quad \sqrt{\boldsymbol{x}\cdot\boldsymbol{x}} = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \|\boldsymbol{x}\|_2 < 1 \tag{15}$$

即，

$$x_1^2 + x_2^2 + \cdots x_n^2 < 1 \tag{16}$$

单位圆、单位球、单位超球面内部的点满足：

$$x_1^2 + x_2^2 + \cdots x_n^2 > 1 \tag{17}$$

## 张量积

列向量 $\boldsymbol{x}$ 和自身的张量积结果为方阵，相当于 $\boldsymbol{x}$ 和 $\boldsymbol{x}^{\mathrm{T}}$ 的乘积：

$$\boldsymbol{x} \otimes \boldsymbol{x} = \boldsymbol{x} \mathbin{@} \boldsymbol{x}^{\mathrm{T}} = \begin{bmatrix} x_1x_1 & x_1x_2 & \cdots & x_1x_n \\ x_2x_1 & x_2x_2 & \cdots & x_2x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_nx_1 & x_nx_2 & \cdots & x_nx_n \end{bmatrix} \tag{18}$$

图 7 所示为 (18) 计算过程。


图 7. 张量积运算

用两种方式展开 (18)，可以得到：

$$\boldsymbol{x} \otimes \boldsymbol{x} = \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \boldsymbol{x}^{\mathrm{T}} = \begin{bmatrix} x_1\boldsymbol{x}^{\mathrm{T}} \\ x_2\boldsymbol{x}^{\mathrm{T}} \\ \vdots \\ x_n\boldsymbol{x}^{\mathrm{T}} \end{bmatrix} \tag{19}$$

$$= \boldsymbol{x}\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} = \begin{bmatrix} x_1\boldsymbol{x} & x_2\boldsymbol{x} & \cdots & x_n\boldsymbol{x} \end{bmatrix}$$

本书前文提过，向量张量积的行向量、列向量都存在“倍数关系”。这实际上解释了为什么非 $\boldsymbol{0}$ 向量张量积的秩 (rank) 为 1。本书第 7 章将介绍“秩”这个概念。另外，请大家注意如图 8 所示的两种形状的张量积和矩阵乘法关系，并注意区分结果形状。

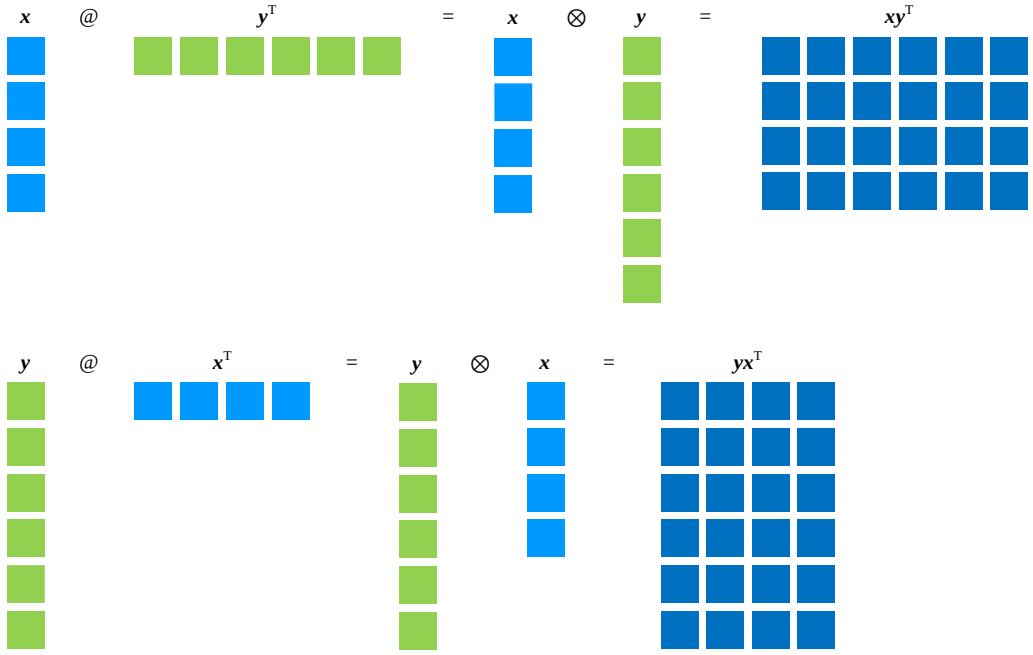



图 8. 另外两种形状的张量积

# 5.3 再聊全 1 列向量

本节主要介绍全 1 列向量 $\boldsymbol{1}$ 在求和方面的作用。

有关 Σ 求和，本系列丛书《数学要素》第 14 章中讲过。本节主要从矩阵乘法角度再深入探讨。

## 每列元素求和

如图 9 所示，全 1 列向量 $\boldsymbol{1}$ 转置左乘数据矩阵 $\boldsymbol{X}$，相当于对 $\boldsymbol{X}$ 每一列元素求和。计算结果为行向量，行向量的每个元素是 $\boldsymbol{X}$ 对应列元素之和：

$$\left(\boldsymbol{1}_{n\times 1}\right)^{\mathrm{T}}\boldsymbol{X}=\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}_{1\times n}\begin{bmatrix}x_{1,1} & x_{1,2} & \cdots & x_{1,D}\\ x_{2,1} & x_{2,2} & \cdots & x_{2,D}\\ \vdots & \vdots & \ddots & \vdots\\ x_{n,1} & x_{n,2} & \cdots & x_{n,D}\end{bmatrix}_{n\times D}=\begin{bmatrix}\sum_{i=1}^{n}x_{i,1} & \sum_{i=1}^{n}x_{i,2} & \cdots & \sum_{i=1}^{n}x_{i,D}\end{bmatrix}_{1\times D} \tag{20}$$

请大家格外注意矩阵形状。全 1 列向量 $\boldsymbol{1}$ 的形状为 $n \times 1$，转置之后 $\boldsymbol{1}^{\mathrm{T}}$ 的形状为 $1 \times n$。数据矩阵 $\boldsymbol{X}$ 的形状为 $n \times D$。矩阵乘积 $\boldsymbol{1}^{\mathrm{T}}\boldsymbol{X}$ 的结果形状为 $1 \times D$。

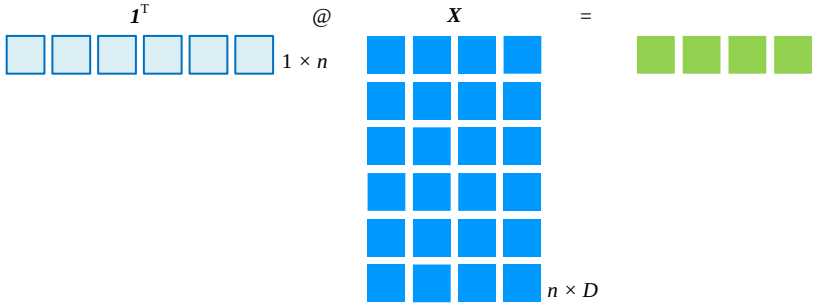



图 9. 列方向求和

(20) 左右除以 $n$，便得到每一列元素均值构成的行向量 E($\boldsymbol{X}$)：

$$\mathrm{E}\left(\boldsymbol{X}\right)=\frac{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{X}}{n}=\begin{bmatrix}\frac{\sum_{i=1}^{n}x_{i,1}}{n} & \frac{\sum_{i=1}^{n}x_{i,2}}{n} & \cdots & \frac{\sum_{i=1}^{n}x_{i,D}}{n}\end{bmatrix}=\begin{bmatrix}\mu_1 & \mu_2 & \cdots & \mu_D\end{bmatrix} \tag{21}$$

E($\boldsymbol{X}$) 常被称作数据矩阵 $\boldsymbol{X}$ 的**质心** (centroid)。我们也常用 $\boldsymbol{\mu}_{\boldsymbol{X}}$ 表达质心。$\boldsymbol{\mu}_{\boldsymbol{X}}$ 为列向量，是行向量 E($\boldsymbol{X}$) 的转置：

$$\boldsymbol{\mu}_{\boldsymbol{X}} = \mathrm{E}\left(\boldsymbol{X}\right)^{\mathrm{T}} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_D \end{bmatrix} = \frac{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{1}}{n} \tag{22}$$

⚠ 注意，本系列丛书定义 $\mathrm{E}(\boldsymbol{X})$ 为行向量。而 $\boldsymbol{\mu}_{\boldsymbol{X}}$ 为列向量，$\boldsymbol{\mu}_{\boldsymbol{X}}$ 和 $\mathrm{E}(\boldsymbol{X})$ 就差在转置上。$\mathrm{E}(\boldsymbol{X})$ 一般常配合原始数据矩阵 $\boldsymbol{X}$ 一起出现，比如利用广播原则去均值。而 $\boldsymbol{\mu}_{\boldsymbol{X}}$ 多用在分布相关运算中，比如多元高斯分布。

### 去均值

上一节提到，全 1 列向量有复制的功能。很多应用场合需要将 (21) 复制 $n$ 份，得到一个和原矩阵形状相同的矩阵。下式可以完成这个计算：

$$\boldsymbol{1}_{n\times 1} @ \mathrm{E}\left(\boldsymbol{X}\right)_{1\times D} = \frac{\boldsymbol{1}_{n\times 1}\boldsymbol{1}_{n\times 1}^{\mathrm{T}}\boldsymbol{X}}{n} = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_D \\ \mu_1 & \mu_2 & \cdots & \mu_D \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1 & \mu_2 & \cdots & \mu_D \end{bmatrix}_{n\times D} \tag{23}$$

上式结果和数据矩阵 $\boldsymbol{X}$ 形状一致，都是 $n \times D$。其中，$\boldsymbol{1}_{n\times 1}\boldsymbol{1}_{n\times 1}^{\mathrm{T}}$ 相当于向量张量积 $\boldsymbol{1}_{n\times 1} \otimes \boldsymbol{1}_{n\times 1}$，结果为 $n \times n$ 全 1 方阵。利用张量积，(23) 可以写成：

$$\boldsymbol{1}_{n\times 1} @ \mathrm{E}\left(\boldsymbol{X}\right)_{1\times D} = \frac{\boldsymbol{1}_{n\times 1} \otimes \boldsymbol{1}_{n\times 1}}{n}\boldsymbol{X} \tag{24}$$

上式相当于是 $\boldsymbol{X}$ 向 $\boldsymbol{1}$ 正交投影，这是本书第 10 章要探讨的内容。

对 $\boldsymbol{X}$ **去均值** (demean 或 centralize) 就是 $\boldsymbol{X}$ 的每个元素减去 $\boldsymbol{X}$ 对应列方向数据均值，即 $\boldsymbol{X}$ 减去 (23) 得到去均值数据矩阵 $\boldsymbol{X}_c$：

$$\boldsymbol{X}_c = \boldsymbol{X} - \frac{\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}\boldsymbol{X}}{n} = \begin{bmatrix} x_{1,1} - \mu_1 & x_{1,2} - \mu_2 & \cdots & x_{1,D} - \mu_D \\ x_{2,1} - \mu_1 & x_{2,2} - \mu_2 & \cdots & x_{2,D} - \mu_D \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} - \mu_1 & x_{n,2} - \mu_2 & \cdots & x_{n,D} - \mu_D \end{bmatrix}_{n\times D} \tag{25}$$

上式可以整理为：

$$\boldsymbol{X} - \frac{\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}\boldsymbol{X}}{n} = \boldsymbol{I}\boldsymbol{X} - \frac{\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}\boldsymbol{X}}{n} = \left(\boldsymbol{I} - \frac{\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}}{n}\right)\boldsymbol{X} \tag{26}$$

其中，$\boldsymbol{I}$ 是单位矩阵，对角线元素都是 1，其余为 0。上式 $\boldsymbol{I}$ 形状为 $n \times n$。

➡ 有关去均值运算，本书第 22 章还要深入这一话题。

如图 10 所示，从几何视角来看，去均值相当于将数据的质心平移到原点。

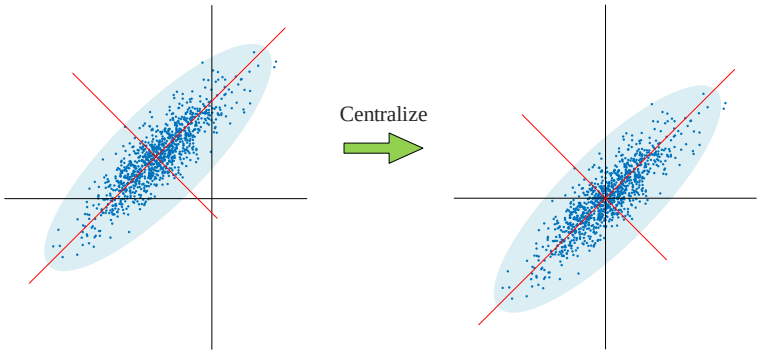



图 10. 去均值的几何视角

用张量积 $\boldsymbol{1} \otimes \boldsymbol{1}$，(25) 可以写成：

$$\boldsymbol{X}_c = \boldsymbol{X} - \frac{\boldsymbol{1} \otimes \boldsymbol{1}}{n} \boldsymbol{X} \tag{27}$$

前文提到，张量积 $\boldsymbol{1} \otimes \boldsymbol{1}$ 是个 $n \times n$ 方阵，矩阵的元素都是 1。张量积 $\boldsymbol{1} \otimes \boldsymbol{1}$ 再除以 $n$ 得到的方阵每个元素都是 $1/n$。

### 每行元素求和

如图 11 所示，据矩阵 $\boldsymbol{X}$ 乘全 1 列向量 $\boldsymbol{1}$，相当于对 $\boldsymbol{X}$ 每一行元素求和，结果为列向量：

$$\boldsymbol{X}_{n\times D}\boldsymbol{1}_{D\times 1} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,D} \end{bmatrix}_{n\times D} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{D\times 1} = \begin{bmatrix} \sum_{j=1}^{D} x_{1,j} \\ \sum_{j=1}^{D} x_{2,j} \\ \vdots \\ \sum_{j=1}^{D} x_{n,j} \end{bmatrix}_{n\times 1} \tag{28}$$

⚠ 注意，(21) 和 (28) 两式中的全 $\boldsymbol{1}$ 向量长度不同。(28) 中全 1 列向量 $\boldsymbol{1}$ 形状为 $D \times 1$。

而 (28) 除以 $D$ 结果是 $\boldsymbol{X}$ 每行元素平均值，即：

$$\frac{\boldsymbol{X}_{n\times D}\boldsymbol{1}_{D\times 1}}{D} = \begin{bmatrix} \sum_{j=1}^{D} x_{1,j} \Big/ D \\ \sum_{j=1}^{D} x_{2,j} \Big/ D \\ \vdots \\ \sum_{j=1}^{D} x_{n,j} \Big/ D \end{bmatrix}_{n\times 1} \tag{29}$$

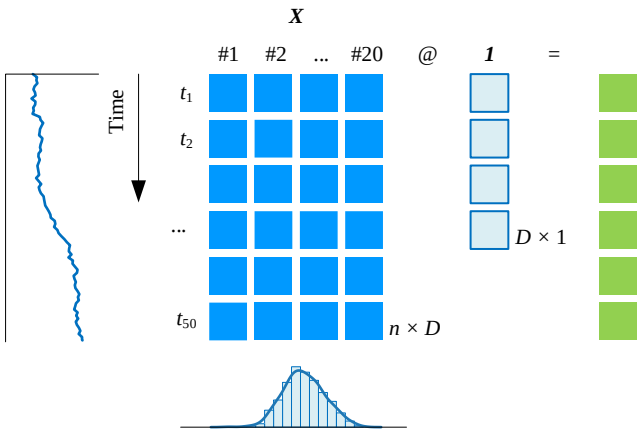



图 11. 行方向求和

大家可能会好奇，数据矩阵的列均值、行均值有怎样的应用场景？

举个例子，假设图 11 中数据矩阵 $\boldsymbol{X}$ 为某个班级 20 名学生不同时间 $t$ 连续 50 次数学测验成绩。每一列的均值代表的是某个学生的平均成绩，每一行的均值则代表一个班级在某次数学测验的整体表现。采用直方图分析列均值，我们可以得到学生平均成绩的分布。采用线图分析行均值，我们可以得到班级学生平均成绩随时间变化趋势。

## 所有元素的和

图 12 所示，数据矩阵 $\boldsymbol{X}$ 分别左乘 $\boldsymbol{1}^{\mathrm{T}}$、右乘全 $\boldsymbol{1}$ 向量，结果为 $\boldsymbol{X}$ 所有元素求和：

$$\boldsymbol{1}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{1} = \begin{bmatrix} \sum_{i=1}^{n} x_{i,1} & \sum_{i=1}^{n} x_{i,2} & \cdots & \sum_{i=1}^{n} x_{i,D} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \sum_{j=1}^{D}\sum_{i=1}^{n} x_{i,j} \tag{30}$$

上式结果除以 $n \times D$，得到的是整个数据矩阵 $\boldsymbol{X}$ 所有元素的均值。

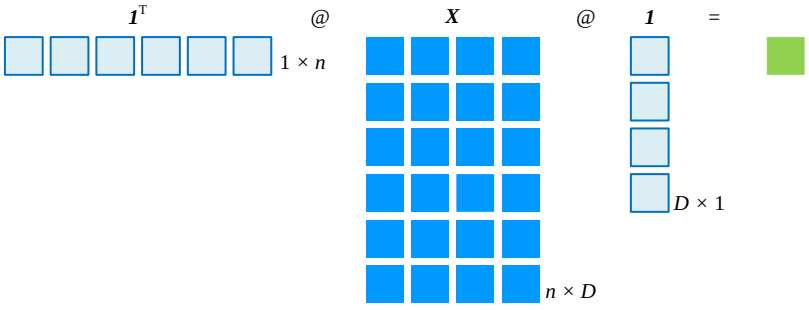



图 12. 矩阵所有元素求和

⚠ 注意，上式中两个全 $\boldsymbol{1}$ 列向量长度也不同，具体形状如图 12 所示。再强调一点，希望读者在看到代数式时，要联想可能的线性代数运算式。本章后续还会继续给出更多示例，以便强化代数和线性代数的联系。

# 5.4 矩阵乘向量：线性方程组

矩阵 $\boldsymbol{A}$ 为 $n$ 行、$D$ 列：

$$\boldsymbol{A}_{n\times D}=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,D} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,D} \end{bmatrix} \tag{31}$$

$\boldsymbol{x}$ 为 $D$ 个未知量 $x_1$、$x_1$ … $x_D$ 构成的列向量，$\boldsymbol{b}$ 为常数 $b_1$、$b_1$ … $b_n$ 构成的列向量：

$$\boldsymbol{x}_{D\times 1}=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix},\quad \boldsymbol{b}_{n\times 1}=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{32}$$

如图 13 所示，$\boldsymbol{Ax}=\boldsymbol{b}$ 可以写成：

$$\underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,D} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,D} \end{bmatrix}}_{\boldsymbol{A}_{n\times D}}\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}}_{\boldsymbol{x}_{D\times 1}}=\underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\boldsymbol{b}_{n\times 1}} \tag{33}$$

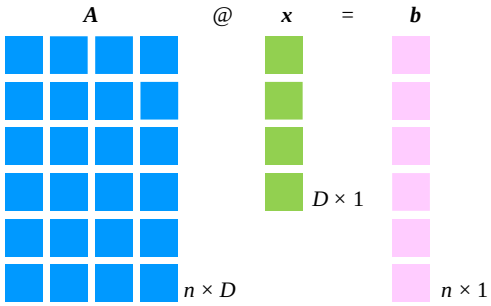



图 13. 长方阵乘列向量

(33) 展开得到**线性方程组** (system of linear equations)：

$$\begin{cases} a_{1,1}x_1+a_{1,2}x_2+\cdots+a_{1,D}x_D=b_1 \\ a_{2,1}x_1+a_{2,2}x_2+\cdots+a_{2,D}x_D=b_2 \\ \vdots \\ a_{n,1}x_1+a_{n,2}x_2+\cdots+a_{n,D}x_D=b_n \end{cases} \tag{34}$$

举个例子，本系列丛书《数学要素》一册中鸡兔同笼问题，就可以写成：

$$\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 35 \\ 94 \end{bmatrix} \quad\Rightarrow\quad \begin{cases} x_1+x_2=35 \\ 2x_1+4x_2=94 \end{cases} \tag{35}$$

### 解的个数

若 (33) 有唯一一组解，矩阵 $\boldsymbol{A}$ 可逆，即：

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b} \tag{36}$$

此时称 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 为恰定方程组。

有无穷多解的**欠定方程组** (underdetermined system)。

解不存在的方程组被称作**超定方程组** (overdetermined system)。

特别地，如果 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 可逆，$\boldsymbol{x}$ 可以通过下式求解：

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{x}=\underbrace{\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}}_{\boldsymbol{A}^{+}}\boldsymbol{b} \tag{37}$$

$\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}$ 常被称作**广义逆** (generalized inverse)，或**伪逆** (pseudoinverse)。

本系列丛书《数学要素》一册介绍过最小二乘法 (ordinary least squares, OLS) 和广义逆之间的关系。本系列丛书《概率统计》和《数据科学》两册还会深入讲解。

线性代数本身具有“代数”属性，这也就是为什么很多教材以求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 为起点讲解线性代数。而本书则试图跳出“代数”的桎梏，从向量、几何、空间、数据等视角理解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$。

### 线性组合视角

下面用另外一个视角看 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$。

本书前文反复提到，矩阵 $\boldsymbol{A}$ 可以看做由一组列向量构造而成：

$$\boldsymbol{A}_{n\times D}=\begin{bmatrix}\boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_D\end{bmatrix} \tag{38}$$

如图 14 所示，(33) 可以写成：

$$\begin{bmatrix}\boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_D\end{bmatrix}_{1\times D}\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_D\end{bmatrix}_{D\times 1}=\boldsymbol{b}_{n\times 1} \tag{39}$$

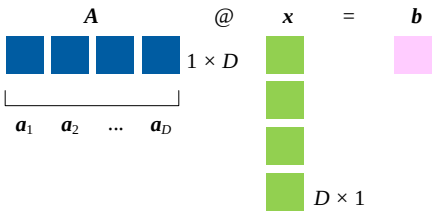


图 14. 线性组合视角看线性方程组

展开 (39) 得到：

$$x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 + \cdots + x_D\boldsymbol{a}_D = \boldsymbol{b}_{n\times 1} \tag{40}$$

即，

$$x_1\underbrace{\begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \vdots \\ a_{n,1} \end{bmatrix}}_{\boldsymbol{a}_1} + x_2\underbrace{\begin{bmatrix} a_{1,2} \\ a_{2,2} \\ \vdots \\ a_{n,2} \end{bmatrix}}_{\boldsymbol{a}_2} + \cdots + x_D\underbrace{\begin{bmatrix} a_{1,D} \\ a_{2,D} \\ \vdots \\ a_{n,D} \end{bmatrix}}_{\boldsymbol{a}_D} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{41}$$

当 $x_1$、$x_1 \ldots x_D$ 取具体值时，上式代表**线性组合** (linear combination)。用腊八粥举个例子，上式相当于不同比例的原料混合，$x_i$ 就是比例，$\boldsymbol{a}_i$ 就是不同的原料。而 $\boldsymbol{b}$ 就是混合得到的八宝粥。

线性组合这个概念非常重要，本书第 7 章将专门介绍。

### 映射视角

如图 15 所示，从线性映射 (linear mapping) 角度来看，(33) 代表从 $\mathbb{R}^n$ 空间到 $\mathbb{R}^D$ 空间的映射。当且仅当矩阵 $\boldsymbol{A}$ 可逆，可以完成从 $\mathbb{R}^D$ 空间到 $\mathbb{R}^n$ 空间的映射。

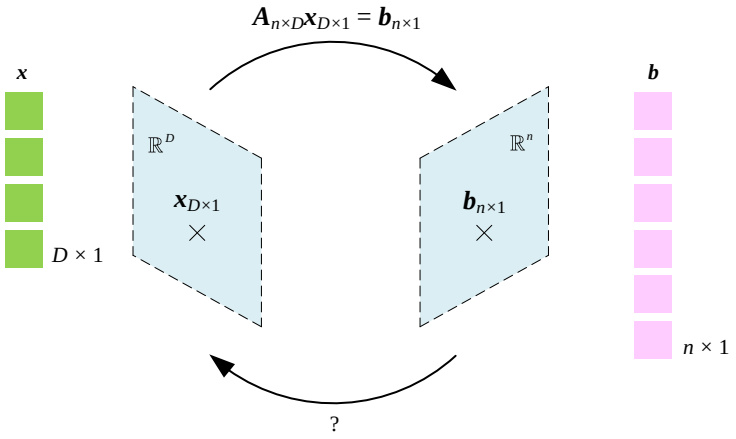



图 15. 线性映射

### 几何视角

如果 2 维向量 $\boldsymbol{x} = [x_1, x_2]^{\mathrm{T}}$ 的模为 1，$\boldsymbol{x}$ 的起点位于原点，终点则位于单位圆上。给定如下矩阵 $\boldsymbol{S}$ 和 $\boldsymbol{R}$：

$$\boldsymbol{S} = \begin{bmatrix} 2 & \\ & 1 \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \tag{42}$$

利用矩阵乘法，$\boldsymbol{x}$ 分别经过 $\boldsymbol{S}$ 和 $\boldsymbol{R}$ 映射得到 $\boldsymbol{y}$：

$$\boldsymbol{y} = \underbrace{\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}}_{\boldsymbol{R}} \underbrace{\begin{bmatrix} 2 & \\ & 1 \end{bmatrix}}_{\boldsymbol{S}} \boldsymbol{x} \tag{43}$$

如图 16 所示，(43) 代表“缩放 → 旋转”。请大家注意几何变换的先后顺序，缩放 ($\boldsymbol{S}$) 先作用于 x，对应矩阵乘法 $\boldsymbol{Sx}$；然后，旋转 ($\boldsymbol{R}$)再作用于 $\boldsymbol{Sx}$，得到 $\boldsymbol{RSx}$。

图 16 中几何变换叫做线性变换 (linear transformation)，这是本书第 8 章要探讨的话题。

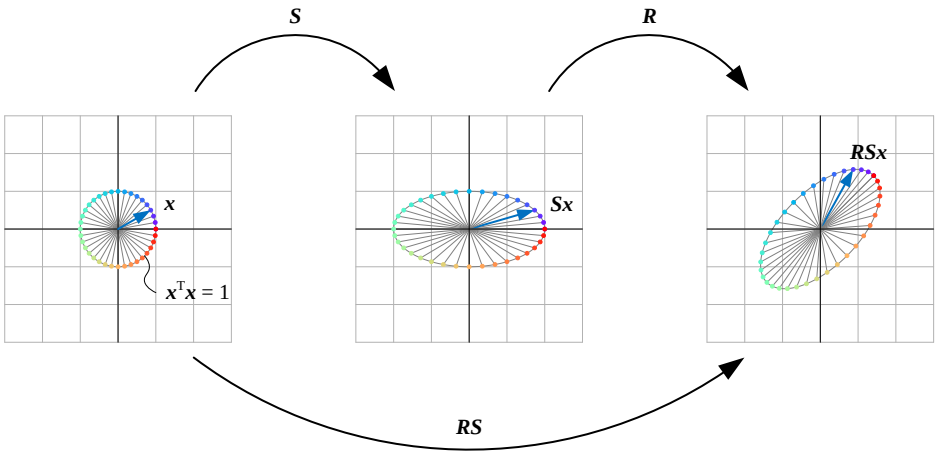



图 16. 几何变换视角

# 5.5 向量乘矩阵乘向量：二次型

**二次型** (quadratic form) 的矩阵算式为：

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} = q \tag{44}$$

其中，$\boldsymbol{Q}$ 为对称阵，$q$ 为实数。$\boldsymbol{Q}$ 和 $\boldsymbol{x}$ 分别为：

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,D} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ q_{D,1} & q_{D,2} & \cdots & q_{D,D} \end{bmatrix} \tag{45}$$

(44) 对应的矩阵运算过程如图 17 所示。$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$ 像极了 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$，也就是说 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$ 类似 $\|\boldsymbol{x}\|_2^2$，结果都是“标量”。几何角度，$\|\boldsymbol{x}\|_2^2$ 代表向量 $\boldsymbol{x}$ 长度的平方，$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$ 似乎也代表着某种“距离的平方”，本书后续将会专门介绍。

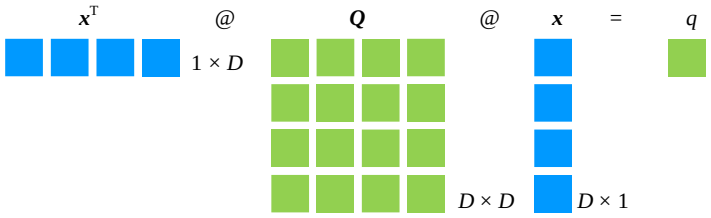



图 17. $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} = q$ 矩阵运算

将 (45) 代入 (44)，展开得到：

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} = \sum_{i=1}^{D} q_{i,i} x_i^2 + \sum_{i=1}^{D}\sum_{j=1}^{D} q_{i,j} x_i x_j = q \tag{46}$$

观察上式，发现单项式变量的最高次数为 2，这就是为什么 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$ 叫二次型的原因。

### 举个例子

比如 $\boldsymbol{x}$ 和 $\boldsymbol{Q}$ 分别为：

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{47}$$

代入 (44) 得到：

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = a x_1^2 + (b+c) x_1 x_2 + d x_2^2 = q \tag{48}$$

可以发现，(48) 对应本系列丛书中《数学要素》介绍过各种二次曲线，比如正圆、椭圆、抛物线或双曲线，如图 18 所示。本书第 20 章还要用线性代数工具深入探讨这些圆锥曲线。

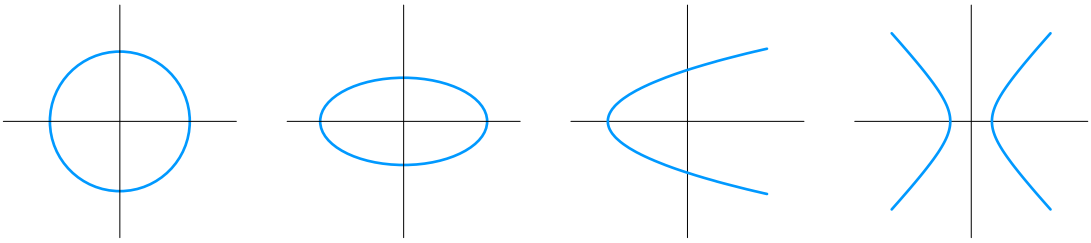


图 18. 四种二次曲线

将 (48) 写成二元函数形式 $f(x_1, x_2)$：

$$f\left(x_1, x_2\right)=\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=ax_1^2+\left(b+c\right)x_1x_2+dx_2^2 \tag{49}$$

(49) 对应着如图 19 所示的几种曲面。而 $f(x_1, x_2) = q$，相当于曲面某个高度的等高线。

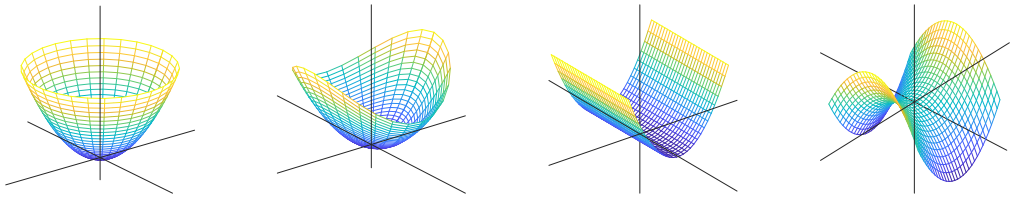


图 19. 常见二次型曲面

本书第 21 章将探讨图 19 这些曲面和正定性、极值之间的联系。

二次型的应用无处不在。举个例子，二元正态分布的概率密度函数解析式如下：

$$f_{X1,X2}\left(x_1, x_2\right)=\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho_{1,2}^2}}\times\exp\left(\frac{-1}{2\left(1-\rho_{1,2}^2\right)}\left(\overbrace{\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2-2\rho_{1,2}\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right)+\left(\frac{x_2-\mu_2}{\sigma_2}\right)^2}^{\text{Ellipse}}\right)\right) \tag{50}$$

相信大家已经看到了其中的椭圆。而多元正态分布的概率密度函数为：

$$f_{\boldsymbol{x}}\left(\boldsymbol{x}\right)=\frac{\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right)}{\left(2\pi\right)^{\frac{D}{2}}\left|\boldsymbol{\Sigma}\right|^{\frac{1}{2}}} \tag{51}$$

分母中已经明显看到类似 (44) 的矩阵乘法。

## 三个方阵连乘

我们再看另外一种二次型，具体如下：

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{V} \tag{52}$$

其中，$\boldsymbol{V}$ 和 $\boldsymbol{\Sigma}$ 都是 $D \times D$ 方阵，上式结果也是 $D \times D$ 方阵。特别地，实际应用中 $\boldsymbol{V}$ 多为正交矩阵，即为 $\boldsymbol{V}$ 方阵且满足 $\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{I}$。

将 $\boldsymbol{V}$ 写成 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, .., \boldsymbol{v}_D]$，展开 (52) 得到：

$$\begin{bmatrix} \boldsymbol{v}_1^{\mathrm{T}} \\ \boldsymbol{v}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_D^{\mathrm{T}} \end{bmatrix}\boldsymbol{\Sigma}\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_D \end{bmatrix}=\begin{bmatrix} \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_1 & \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_2 & \cdots & \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_D \\ \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_1 & \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_2 & \cdots & \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_D \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{v}_D^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_1 & \boldsymbol{v}_D^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_2 & \cdots & \boldsymbol{v}_D^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_D \end{bmatrix} \tag{53}$$

上式中结果矩阵 $(i, j)$ 元素为 $\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_j$，$\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_j$ 对应的运算示意图如图 20 所示。

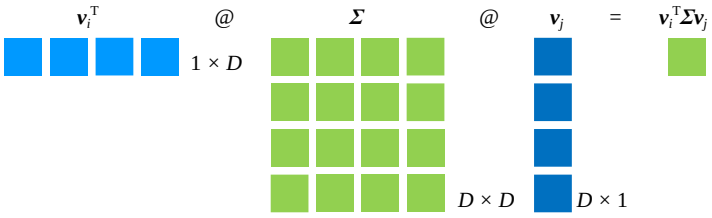



图 20. $\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{v}_j$ 矩阵运算

二次型在多元微积分、正定性、多元正态分布、协方差矩阵、数据映射和优化方法中都有举足轻重的分量。本书后续将会深入探讨。

## 5.6 方阵乘方阵：矩阵分解

和方阵有关的矩阵乘法中，方阵乘方阵最为简单。图 21 所示两种方阵乘法常见于 LU 分解、Cholesky 分解、特征值分解等场合。

本节不展开讲解矩阵分解，本书第 11 ~ 16 章将专门介绍不同类别矩阵分解。

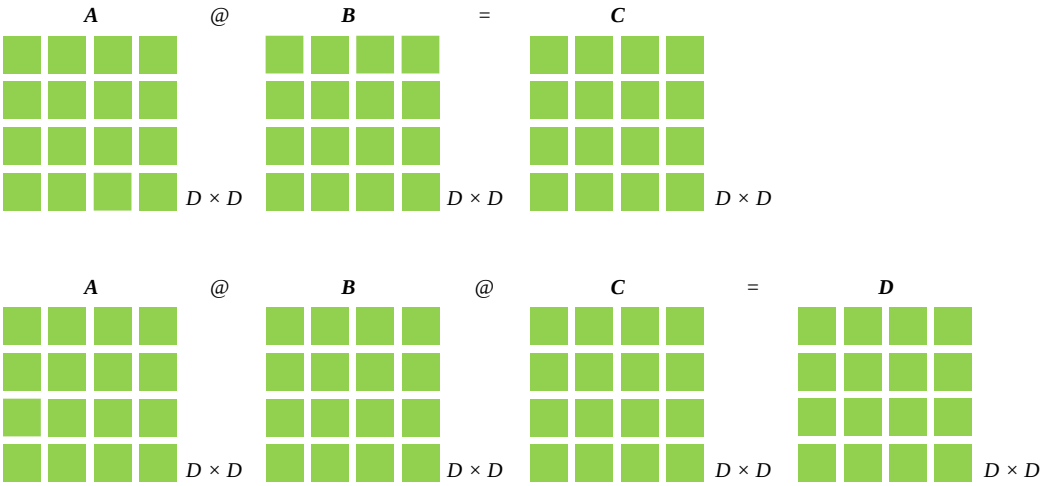



图 21. 方阵乘方阵

特别地，方阵 $\boldsymbol{A}$ 如果满足：

$$\boldsymbol{A}^2 = \boldsymbol{A} \tag{54}$$

则称 $\boldsymbol{A}$ 为**幂等矩阵** (idempotent matrix)。

我们会在本书统计部分和最小二乘法线性回归中再谈及幂等矩阵。丛书每册均有涉及线性回归这个话题，本书采用的是线性代数和向量几何视角，《概率统计》则利用统计视角理解线性回归，而《数据科学》则是从数据分析视角介绍如何应用这个模型。

# 5.7 对角阵：批量缩放

如果形状相同的方阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 都为对角阵，两者乘积还是一个对角阵：

$$\boldsymbol{A}_{D\times D}\boldsymbol{B}_{D\times D}=\begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_D \end{bmatrix}\begin{bmatrix} b_1 & & & \\ & b_2 & & \\ & & \ddots & \\ & & & b_D \end{bmatrix}=\begin{bmatrix} a_1b_1 & & & \\ & a_2b_2 & & \\ & & \ddots & \\ & & & a_Db_D \end{bmatrix} \tag{55}$$

对角阵 $\boldsymbol{\Lambda}$ 的逆也是一个对角阵：

$$\boldsymbol{\Lambda}_{D\times D}\left(\boldsymbol{\Lambda}_{D\times D}\right)^{-1}=\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix}\begin{bmatrix} 1/\lambda_1 & & & \\ & 1/\lambda_2 & & \\ & & \ddots & \\ & & & 1/\lambda_D \end{bmatrix}=\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}=\boldsymbol{I}_{D\times D} \tag{56}$$

其中，$\lambda_j \neq 0$。注意，本书中经常采用 $\boldsymbol{\Lambda}$ (capital lambda) 和 $\boldsymbol{S}$ 代表对角阵。

### 右乘

矩阵 $\boldsymbol{X}$ 乘 $D \times D$ 对角方阵 $\boldsymbol{\Lambda}$：

$$\begin{aligned}\boldsymbol{X}_{n\times D}\boldsymbol{\Lambda}_{D\times D}&=\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_D\end{bmatrix}\\&=\begin{bmatrix}\lambda_1\boldsymbol{x}_1 & \lambda_2\boldsymbol{x}_2 & \cdots & \lambda_D\boldsymbol{x}_D\end{bmatrix}\end{aligned} \tag{57}$$

观察 (57) 发现，$\boldsymbol{\Lambda}$ 的对角线元素相当于缩放系数，分别对矩阵 $\boldsymbol{X}$ 的每一列数值进行不同比例缩放。

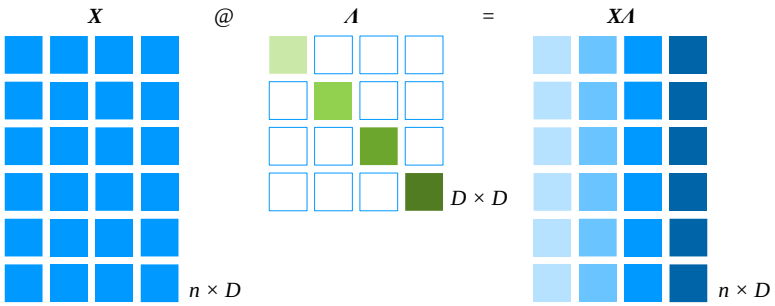



图 22. $\boldsymbol{X}$ 乘对角方阵 $\boldsymbol{\Lambda}$

## 左乘

$n \times n$ 对角阵 $\boldsymbol{\Lambda}$ 左乘矩阵 $\boldsymbol{X}$:

$$\boldsymbol{\Lambda}_{n\times n}\boldsymbol{X}_{n\times D}=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_n\end{bmatrix}_{n\times n}\begin{bmatrix}\boldsymbol{x}^{(1)}\\ \boldsymbol{x}^{(2)}\\ \vdots\\ \boldsymbol{x}^{(n)}\end{bmatrix}_{n\times 1}=\begin{bmatrix}\lambda_1\boldsymbol{x}^{(1)}\\ \lambda_2\boldsymbol{x}^{(2)}\\ \vdots\\ \lambda_n\boldsymbol{x}^{(n)}\end{bmatrix}_{n\times 1} \tag{58}$$

观察 (58)，可以发现 $\boldsymbol{\Lambda}$ 的对角线元素分别对矩阵 $\boldsymbol{X}$ 的每一行数值进行批量缩放。

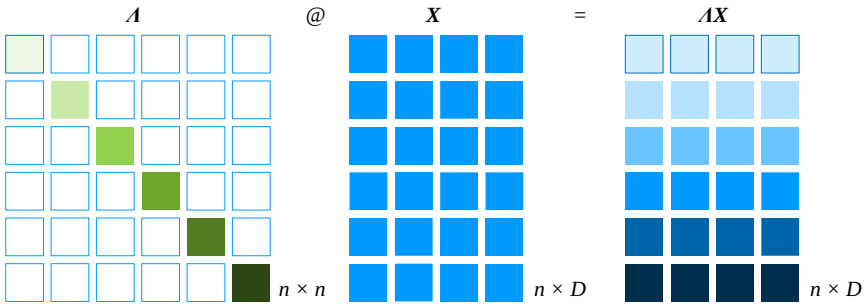



图 23. 对角阵 $\boldsymbol{\Lambda}$ 乘方阵 $\boldsymbol{X}$

## 乘行向量

特别地，行向量 $\boldsymbol{x}^{(1)}$ 乘 $D \times D$ 对角阵 $\boldsymbol{\Lambda}$，相当于对行向量每个元素以不同比例分别缩放:

$$\boldsymbol{x}^{(1)}\boldsymbol{\Lambda}_{D\times D}=\begin{bmatrix}x_{1,1} & x_{1,2} & \cdots & x_{1,D}\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_D\end{bmatrix}_{D\times D}=\begin{bmatrix}\lambda_1 x_{1,1} & \lambda_2 x_{1,2} & \cdots & \lambda_D x_{1,D}\end{bmatrix} \tag{59}$$

## 乘列向量

类似地，$n \times n$ 对角阵 $\boldsymbol{\Lambda}$ 乘列向量 $\boldsymbol{x}$，相当于对列向量每个元素以不同比例分别缩放：

$$\boldsymbol{\Lambda}_{n\times n}\boldsymbol{x}_{n\times 1}=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_n\end{bmatrix}_{n\times n}\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}_{n\times 1}=\begin{bmatrix}\lambda_1 x_1\\ \lambda_2 x_2\\ \vdots\\ \lambda_n x_n\end{bmatrix}_{n\times 1} \tag{60}$$

## 左右都乘

再看下例，$D \times D$ 对角方阵 $\boldsymbol{\Lambda}$ 分别左乘、右乘 $D \times D$ 方阵 $\boldsymbol{B}$：

$$\begin{aligned}\boldsymbol{\Lambda B \Lambda}&=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_D\end{bmatrix}\begin{bmatrix}b_{1,1} & b_{1,2} & \cdots & b_{1,D}\\ b_{2,1} & b_{2,2} & \cdots & b_{2,D}\\ \vdots & \vdots & \ddots & \vdots\\ b_{D,1} & b_{D,2} & \cdots & b_{D,D}\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_D\end{bmatrix}\\ &=\begin{bmatrix}\lambda_1\lambda_1 b_{1,1} & \lambda_1\lambda_2 b_{1,2} & \cdots & \lambda_1\lambda_D b_{1,D}\\ \lambda_2\lambda_1 b_{2,1} & \lambda_2\lambda_2 b_{2,2} & \cdots & \lambda_2\lambda_D b_{2,D}\\ \vdots & \vdots & \ddots & \vdots\\ \lambda_D\lambda_1 b_{D,1} & \lambda_D\lambda_2 b_{D,2} & \cdots & \lambda_D\lambda_D b_{D,D}\end{bmatrix}\end{aligned} \tag{61}$$

看到 (61) 结果形式，大家是否想到了协方差矩阵。$\lambda_i$ 相当于均方差，$b_{i,j}$ 相当于相关性系数。

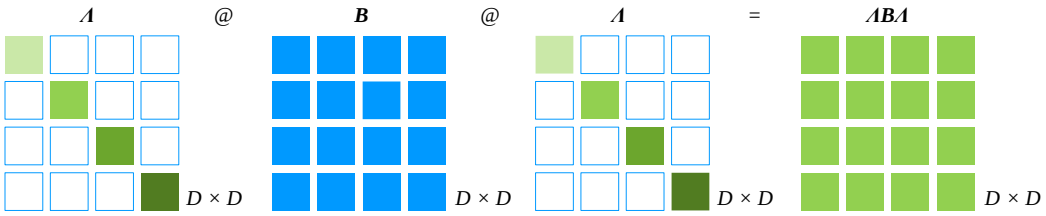



图 24. 对角阵 $\boldsymbol{\Lambda}$ 分别左乘、右乘方阵 $\boldsymbol{B}$

## 二次型特例

再看一个二次型的特例：

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Lambda}_{D\times D}\boldsymbol{x}=\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_D\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_D\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_D\end{bmatrix}=\lambda_1 x_1^2+\lambda_2 x_2^2+\cdots+\lambda_D x_D^2=\sum_{j=1}^{D}\lambda_j x_j^2 \tag{62}$$

图 25 所示为上述运算的示意图。

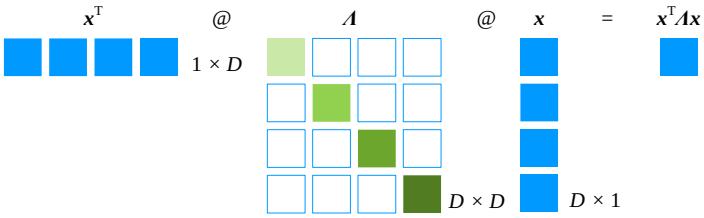



图 25. $x^T\Lambda x$ 对应的矩阵运算

### 几何视角

看到类似 (62) 形式运算，希望大家能联想到正椭圆、正椭球、正椭圆抛物面。比如，如果 $\lambda_1 > \lambda_2 > 0$，且 $k > 0$，下式对应正椭圆：

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = k \tag{63}$$

这个椭圆的半长轴长度为 $\sqrt{k/\lambda_2}$，半短轴长度为 $\sqrt{k/\lambda_1}$。

举个例子，下式对应的正椭圆半长轴长度为 2，半短轴长度为 1：

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 1/4 & \\ & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{4}x_1^2 + x_2^2 = 1 \tag{64}$$

再次强调，如果在矩阵运算时遇到对角阵，请试着从几何体缩放角度来看。

# 5.8 置换矩阵：调换元素顺序

行向量 $\boldsymbol{a}$ 乘副对角矩阵，如果副对角线上元素都为 1，得到左右翻转的行向量：

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_D \end{bmatrix}_{1\times D} \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{bmatrix}_{D\times D} = \begin{bmatrix} a_D & a_{D-1} & \cdots & a_1 \end{bmatrix} \tag{65}$$

实际上，(65) 中完成左右翻转的方阵是**置换矩阵** (permutation matrix) 的一种特殊形式。

置换矩阵是由 0 和 1 组成的方阵。置换矩阵的每一行、每一列都恰好只有一个 1，其余元素均为 0。置换矩阵的作用是调换元素顺序。

举个例子：

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} \begin{bmatrix} & 1 & & \\ & & & 1 \\ 1 & & & \\ & & 1 & \end{bmatrix} = \begin{bmatrix} a_3 & a_1 & a_4 & a_2 \end{bmatrix} \tag{66}$$

### 调整列向量顺序

置换矩阵同样可以作用于矩阵，将 (66) 中行向量元素替换成列向量，即，

$$\boldsymbol{a}_1 = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \\ a_{4,1} \end{bmatrix}, \quad \boldsymbol{a}_2 = \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ a_{3,2} \\ a_{4,2} \end{bmatrix}, \quad \boldsymbol{a}_3 = \begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{bmatrix}, \quad \boldsymbol{a}_4 = \begin{bmatrix} a_{1,4} \\ a_{2,4} \\ a_{3,4} \\ a_{4,4} \end{bmatrix} \tag{67}$$

可以得到：

$$\begin{bmatrix} \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \\ a_{4,1} \end{bmatrix} & \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ a_{3,2} \\ a_{4,2} \end{bmatrix} & \begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{bmatrix} & \begin{bmatrix} a_{1,4} \\ a_{2,4} \\ a_{3,4} \\ a_{4,4} \end{bmatrix} \end{bmatrix} \begin{bmatrix} & 1 & & \\ & & & 1 \\ 1 & & & \\ & & 1 & \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{4,3} \end{bmatrix} & \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \\ a_{4,1} \end{bmatrix} & \begin{bmatrix} a_{1,4} \\ a_{2,4} \\ a_{3,4} \\ a_{4,4} \end{bmatrix} & \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ a_{3,2} \\ a_{4,2} \end{bmatrix} \end{bmatrix} \tag{68}$$

大家看到置换矩阵右乘矩阵 $\boldsymbol{A}$，让 $\boldsymbol{A}$ 的列向量顺序改变。

### 调整行向量顺序

这个置换矩阵左乘矩阵 $\boldsymbol{A}$，可以改变 $\boldsymbol{A}$ 的行向量的排序：

$$\begin{bmatrix} & 1 & & \\ & & & 1 \\ 1 & & & \\ & & 1 & \end{bmatrix} \begin{bmatrix} \boldsymbol{a}^{(1)} \\ \boldsymbol{a}^{(2)} \\ \boldsymbol{a}^{(3)} \\ \boldsymbol{a}^{(4)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}^{(2)} \\ \boldsymbol{a}^{(4)} \\ \boldsymbol{a}^{(1)} \\ \boldsymbol{a}^{(3)} \end{bmatrix} \tag{69}$$

置换矩阵可以用来简化一些矩阵运算。

## 5.9 矩阵乘向量：映射到一维

任何矩阵乘法都可以从**线性映射** (linear mapping) 角度理解。形状为 $n \times D$ 矩阵 $\boldsymbol{X}$ 乘 $D \times 1$ 列向量 $\boldsymbol{v}$ 得到 $n \times 1$ 列向量 $\boldsymbol{z}$：

$$\boldsymbol{X}_{n \times D} \boldsymbol{v}_{D \times 1} = \boldsymbol{z}_{n \times 1} \tag{70}$$

如图 26 所示，矩阵 $\boldsymbol{X}$ 有 $D$ 列，对应 $D$ 个特征。而结果 $\boldsymbol{z}$ 只有一列，也就是一个特征。类似 (41)，(70) 也可以写成“线性组合”：

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D \end{bmatrix}}_{\boldsymbol{X}} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_D \end{bmatrix}}_{\boldsymbol{v}} = v_1\boldsymbol{x}_1 + v_2\boldsymbol{x}_2 + \cdots + v_D\boldsymbol{x}_D = \boldsymbol{z} \tag{71}$$

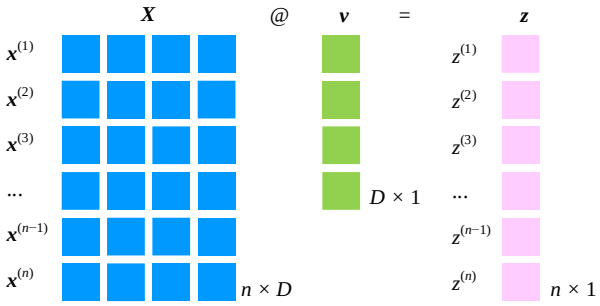



图 26. 矩阵乘法 $\boldsymbol{Xv} = \boldsymbol{z}$

此外，$\boldsymbol{Xv} = \boldsymbol{z}$ 可以展开写成：

$$\boldsymbol{Xv} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \vdots \\ \boldsymbol{x}^{(n)} \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} \boldsymbol{x}^{(1)}\boldsymbol{v} \\ \boldsymbol{x}^{(2)}\boldsymbol{v} \\ \vdots \\ \boldsymbol{x}^{(n)}\boldsymbol{v} \end{bmatrix} = \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \vdots \\ z^{(n)} \end{bmatrix} \tag{72}$$

把矩阵 $\boldsymbol{X}$ 中任意一行 $\boldsymbol{x}^{(i)}$ 看做是多维坐标系的一个点，运算 $\boldsymbol{x}^{(i)}\boldsymbol{v}$ 则是点 $\boldsymbol{x}^{(i)}$ 在 $\boldsymbol{v}$ 方向映射，$z^{(i)}$ 则是结果在 $\boldsymbol{v}$ 上的坐标。如图 27 所示，(70) 这个矩阵乘法运算过程相当于降维。

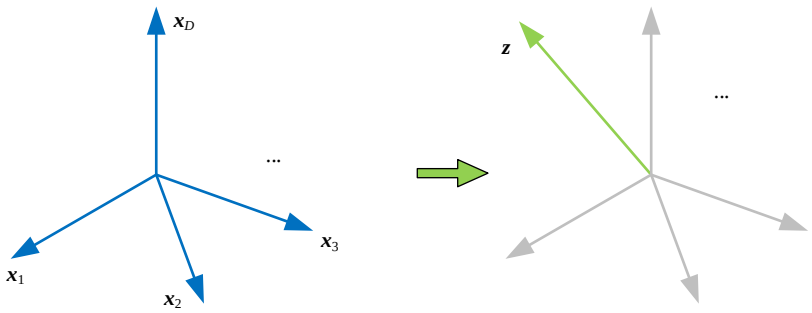



图 27. 多维到一维映射

### 以鸢尾花数据为例

为了方便理解，下面我们将给 $\boldsymbol{v}$ 赋予具体数值来讲解。

以鸢尾花数据为例，矩阵 $\boldsymbol{X}$ 的 4 列分别对应 4 个特征——花萼长度、花萼宽度、花瓣长度、花瓣宽度。$\boldsymbol{Xv} = \boldsymbol{z}$ 结果只有 1 列，相当于只有 1 个特征。

举个例子，如果 $\boldsymbol{v}$ 中第三个元素为 1，其余元素均为 0，如图 28 所示。$\boldsymbol{Xv}$ 的结果是从 $\boldsymbol{X}$ 中提取第 3 列 $\boldsymbol{x}_3$：

$$\boldsymbol{Xv} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \boldsymbol{x}_4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \boldsymbol{x}_3 \tag{73}$$

也就是说，运算结果只保留第三列花瓣长度相关数据。

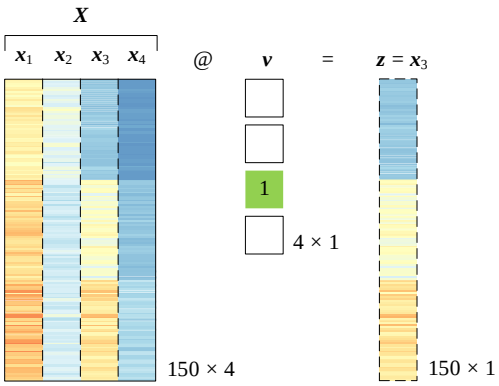



图 28. $\boldsymbol{v}$ 只有第三个元素为 1，其余均为 0

再举个例子，若我们想要计算每个样本花萼长度 ($\boldsymbol{x}_1$)、花萼宽度 ($\boldsymbol{x}_2$) 的平均值，可以通过如下运算得到：

$$\boldsymbol{Xv} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \boldsymbol{x}_4 \end{bmatrix} \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \\ 0 \end{bmatrix} = \frac{\boldsymbol{x}_1 + \boldsymbol{x}_2}{2} \tag{74}$$

同理，下式可以计算每个样本花萼长度 ($\boldsymbol{x}_1$)、花萼宽度 ($\boldsymbol{x}_2$)、花瓣长度 ($\boldsymbol{x}_3$)、花瓣宽度 ($\boldsymbol{x}_4$) 四个特征平均值：

$$\boldsymbol{Xv} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \boldsymbol{x}_4 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \frac{\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3 + \boldsymbol{x}_4}{4} \tag{75}$$

几何角度来看上式运算，(75) 相当于 4 维空间的散点，被压缩到了一条轴上，具体如图 29。

⚠ 注意，除非用点的颜色代表第 4 维度，4 维空间的散点图是很难一幅图上完成可视化，图 29 中 4 维空间的散点仅仅是示意图而已。此外，请大家回顾本书第 1 章介绍的成对特征散点图。

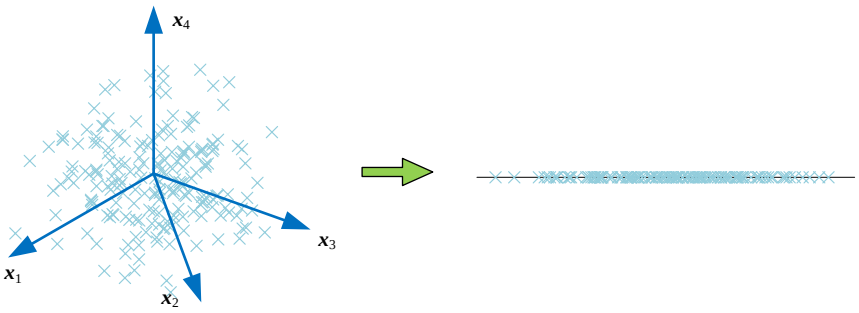



图 29. 4 维空间散点压缩到 1 维

# 5.10 矩阵乘矩阵：映射到多维

有了上一节内容做基础，这一节我们介绍矩阵乘法在多维映射中扮演色角色。

## 两个方向映射

还是以鸢尾花数据矩阵 $\boldsymbol{X}$ 为例，矩阵乘法 $\boldsymbol{X}[\boldsymbol{v}_1, \boldsymbol{v}_2]$ 代表 $\boldsymbol{X}$ 将朝着 $[\boldsymbol{v}_1, \boldsymbol{v}_2]$ 两个方向映射。如果 $[\boldsymbol{v}_1, \boldsymbol{v}_2]$ 的取值如图 30 所示，矩阵乘法 $\boldsymbol{X}$ $[\boldsymbol{v}_1, \boldsymbol{v}_2]$ 完成的是提取 $\boldsymbol{X}$ 的第 1、3 两列，并将两者顺序调换。

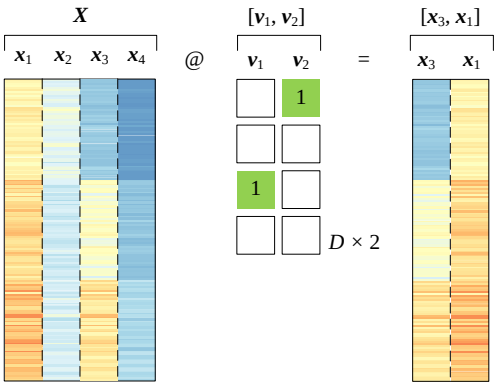



图 30. $\boldsymbol{X}$ 朝两个方向映射

想象一个由鸢尾花四个维度构造的空间 $\mathbb{R}^4$，图 30 相当于将鸢尾花数据映射在一个平面上 $\mathbb{R}^2$，得到的是平面散点图，过程如图 31 所示。根据上一节内容，平面上的数据可以进一步压缩到一条直线上，具体如图 32 所示。

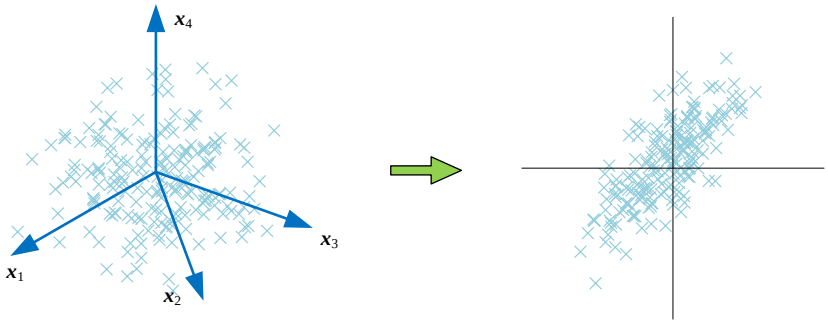



图 31. 4 维空间散点压缩到平面上

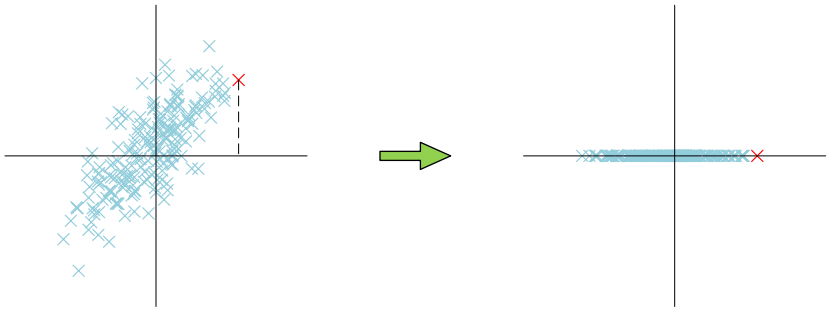


图 32. 平面散点进一步压缩到 1 维

### 多个方向映射

矩阵 $\boldsymbol{X}$ 有 $D$ 个维度，可以通过矩阵乘法，将 $\boldsymbol{X}$ 映射到另外一个 $D$ 维度的空间中。

下例中，$\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$，$\boldsymbol{Z}$ 对应的每一行元素则是新坐标系中各个维度的坐标值：

$$\boldsymbol{XV} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \vdots \\ \boldsymbol{x}^{(n)} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_D \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(1)}\boldsymbol{v}_1 \\ \boldsymbol{x}^{(2)}\boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{x}^{(n)}\boldsymbol{v}_1 \end{bmatrix} & \begin{bmatrix} \boldsymbol{x}^{(1)}\boldsymbol{v}_2 \\ \boldsymbol{x}^{(2)}\boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{x}^{(n)}\boldsymbol{v}_2 \end{bmatrix} & \cdots & \begin{bmatrix} \boldsymbol{x}^{(1)}\boldsymbol{v}_D \\ \boldsymbol{x}^{(2)}\boldsymbol{v}_D \\ \vdots \\ \boldsymbol{x}^{(n)}\boldsymbol{v}_D \end{bmatrix} \end{bmatrix} = \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}^{(1)} \\ \boldsymbol{z}^{(2)} \\ \vdots \\ \boldsymbol{z}^{(n)} \end{bmatrix} \tag{76}$$

其中，矩阵 $\boldsymbol{V}$ 为方阵。

如果 $\boldsymbol{V}$ 可逆，$\boldsymbol{V}$ 就是 $\boldsymbol{X}$ 和 $\boldsymbol{Z}$ 相互转化的桥梁：

$$\boldsymbol{X}=\begin{bmatrix}\boldsymbol{x}^{(1)}\\ \boldsymbol{x}^{(2)}\\ \vdots\\ \boldsymbol{x}^{(n)}\end{bmatrix}\underset{\boldsymbol{V}^{-1}}{\overset{\boldsymbol{V}}{\rightleftharpoons}}\boldsymbol{Z}=\begin{bmatrix}\boldsymbol{z}^{(1)}\\ \boldsymbol{z}^{(2)}\\ \vdots\\ \boldsymbol{z}^{(n)}\end{bmatrix} \tag{77}$$

本书第 10 章还会深入讨论上式。

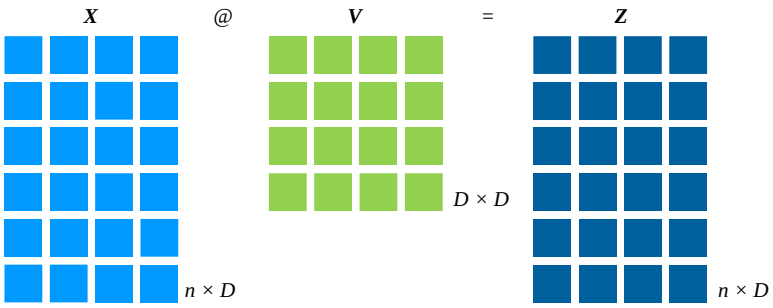



图 33. 一个 $D$ 维度空间 $\boldsymbol{X}$ 数据映射到另一个 $D$ 维度空间

## 列向量形式

当读者见到如下形式时，也不用慌张。如图 34 所示，这个也是上文介绍的映射，只不过 $\boldsymbol{x}$ 为列向量：

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{z} \tag{78}$$

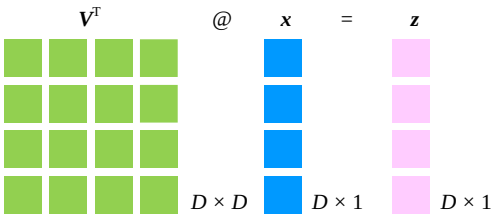



图 34. $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{z}$ 运算示意图

如图 35 所示，(78) 左右转置，便得到类似 (76) 的结构：

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}=\boldsymbol{z}^{\mathrm{T}} \tag{79}$$

可以说，本书后续介绍的内容几乎都离不开映射，比如几何变换、正交投影、特征值分解、奇异值分解等等。

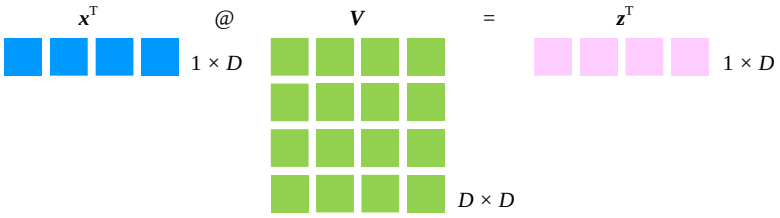



图 35. 等式 $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{z}$ 左右转置

# 5.11 长方阵：奇异值分解、格拉姆矩阵、张量积

本节介绍和长方形矩阵有关的重要矩阵乘法。

## 奇异值分解

请读者格外注意图 36 所示的矩阵乘法结构。这两种形式经常出现在**奇异值分解** (singular vector decomposition, SVD) 和**主成分分析** (principal component analysis, PCA)。

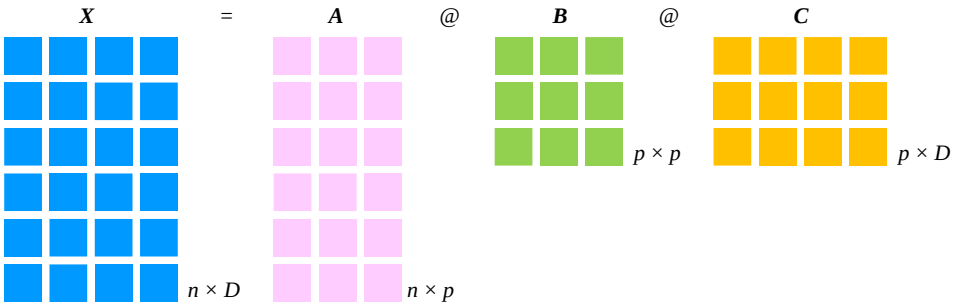



图 36. 三个矩阵相乘

⚠ 请读者注意图 36 中，$D$ 和 $p$ 的大小关系；不同大小关系对应着不同类型的奇异值分解。本书第 16 章将深入讲解。

## 格拉姆矩阵

将矩阵 $\boldsymbol{X}$ 写成一排列向量，如下：

$$\boldsymbol{X}_{n\times D}=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,D} \end{bmatrix}=\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D \end{bmatrix} \tag{80}$$

如图 37 所示，利用 (80)，转置 $\boldsymbol{X}^{\mathrm{T}}$ ($D \times n$) 乘矩阵 $\boldsymbol{X}$ ($n \times D$)，得到一个 $D \times D$ 方阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$，可以写成：

$$\boldsymbol{G}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}=\begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_D^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D \end{bmatrix}=\begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_D \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_D \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_D^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_D^{\mathrm{T}}\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D^{\mathrm{T}}\boldsymbol{x}_D \end{bmatrix} \tag{81}$$

上式是矩阵乘法的第一视角。

(81) 中的 $\boldsymbol{G}$ 有自己的名字——**格拉姆矩阵** (Gram matrix)。格拉姆矩阵在数据分析、机器学习算法中有重要作用。

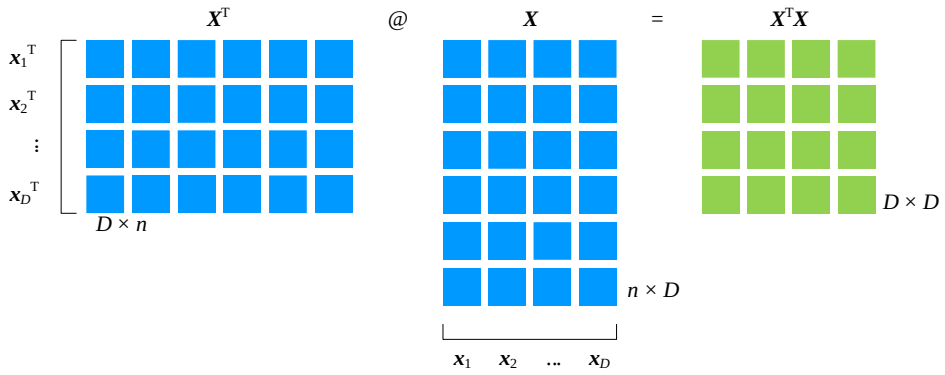



图 37. $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 运算过程

如图 38 所示，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的 $(i,j)$ 元素是 $\boldsymbol{X}$ 中第 $i$ 列向量转置乘以 $\boldsymbol{X}$ 的第 $j$ 列向量：

$$\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)_{i,j}=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \tag{82}$$

当 $i=j$ 时，$\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i$ 对应的是格拉姆矩阵 $\boldsymbol{G}$ 的对角线元素。

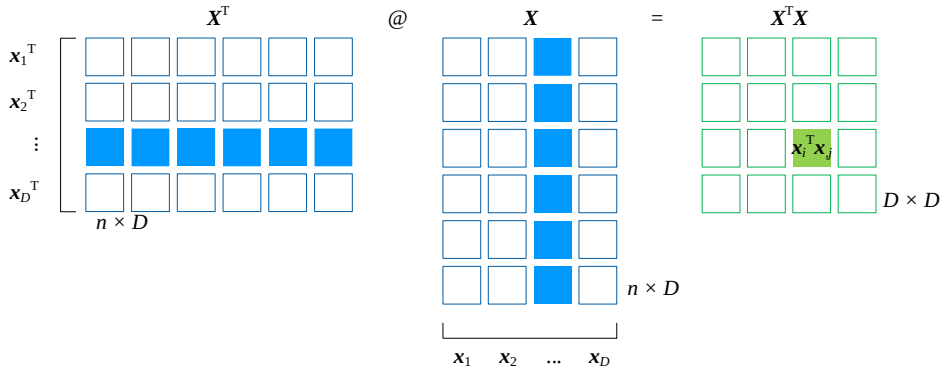



图 38. $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的 $(i,j)$ 元素

## 标量积

$\boldsymbol{G}$ 还可以写成标量积：

$$\boldsymbol{G}=\begin{bmatrix}\boldsymbol{x}_1\cdot\boldsymbol{x}_1 & \boldsymbol{x}_1\cdot\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_1\cdot\boldsymbol{x}_D\\ \boldsymbol{x}_2\cdot\boldsymbol{x}_1 & \boldsymbol{x}_2\cdot\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_2\cdot\boldsymbol{x}_D\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{x}_D\cdot\boldsymbol{x}_1 & \boldsymbol{x}_D\cdot\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D\cdot\boldsymbol{x}_D\end{bmatrix}=\begin{bmatrix}\langle\boldsymbol{x}_1,\boldsymbol{x}_1\rangle & \langle\boldsymbol{x}_1,\boldsymbol{x}_2\rangle & \cdots & \langle\boldsymbol{x}_1,\boldsymbol{x}_D\rangle\\ \langle\boldsymbol{x}_2,\boldsymbol{x}_1\rangle & \langle\boldsymbol{x}_2,\boldsymbol{x}_2\rangle & \cdots & \langle\boldsymbol{x}_2,\boldsymbol{x}_D\rangle\\ \vdots & \vdots & \ddots & \vdots\\ \langle\boldsymbol{x}_D,\boldsymbol{x}_1\rangle & \langle\boldsymbol{x}_D,\boldsymbol{x}_2\rangle & \cdots & \langle\boldsymbol{x}_D,\boldsymbol{x}_D\rangle\end{bmatrix} \tag{83}$$

显然，格拉姆矩阵 $\boldsymbol{G}$ 为对称矩阵：

$$\boldsymbol{G}^{\mathrm{T}}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{\mathrm{T}}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}=\boldsymbol{G} \tag{84}$$

一般情况，数据矩阵 $\boldsymbol{X}$ 都是“细高”长方形矩阵，矩阵运算时这种形状不够友好。比如，细高的 $\boldsymbol{X}$ 显然不存在转置。而把 $\boldsymbol{X}$ 转化为方阵 $\boldsymbol{G}$ (= $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$) 之后，很多运算都变得更加容易。

本书第 22 章介绍协方差矩阵 (covariance matrix) 时，也将采用类似 (81) 的计算思路。

## 张量积

将矩阵 $\boldsymbol{X}$ 写成一系列行向量，如下：

$$\boldsymbol{X}_{n\times D}=\begin{bmatrix}x_{1,1} & x_{1,2} & \cdots & x_{1,D}\\ x_{2,1} & x_{2,2} & \cdots & x_{2,D}\\ \vdots & \vdots & \ddots & \vdots\\ x_{n,1} & x_{n,2} & \cdots & x_{n,D}\end{bmatrix}=\begin{bmatrix}\boldsymbol{x}^{(1)}\\ \boldsymbol{x}^{(2)}\\ \vdots\\ \boldsymbol{x}^{(n)}\end{bmatrix} \tag{85}$$

利用 (85)，格拉姆矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$，可以写成一系列张量积的和：

$$\boldsymbol{G}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}=\begin{bmatrix}\boldsymbol{x}^{(1)\mathrm{T}} & \boldsymbol{x}^{(2)\mathrm{T}} & \cdots & \boldsymbol{x}^{(n)\mathrm{T}}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}^{(1)}\\ \boldsymbol{x}^{(2)}\\ \vdots\\ \boldsymbol{x}^{(n)}\end{bmatrix}=\sum_{i=1}^{n}\boldsymbol{x}^{(i)\mathrm{T}}\boldsymbol{x}^{(i)}=\sum_{i=1}^{n}\boldsymbol{x}^{(i)}\otimes\boldsymbol{x}^{(i)} \tag{86}$$

上式是矩阵乘法的第二视角。

## 另一个格拉姆矩阵

本节前文的数据矩阵 $\boldsymbol{X}$ 是细高的，它转置之后得到粗矮的矩阵 $\boldsymbol{X}^{\mathrm{T}}$。而 $\boldsymbol{X}^{\mathrm{T}}$ 也有自己的格拉姆矩阵，即矩阵 $\boldsymbol{X}$ $(n\times D)$ 乘其转置 $\boldsymbol{X}^{\mathrm{T}}$ $(D\times n)$，得到一个 $n\times n$ 格拉姆矩阵：

$$\begin{aligned}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}&=\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}}\\ \boldsymbol{x}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{x}_D^{\mathrm{T}}\end{bmatrix}\\&=\boldsymbol{x}_1\boldsymbol{x}_1^{\mathrm{T}}+\boldsymbol{x}_2\boldsymbol{x}_2^{\mathrm{T}}+\cdots+\boldsymbol{x}_D\boldsymbol{x}_D^{\mathrm{T}}\\&=\sum_{i=1}^{D}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\end{aligned} \tag{87}$$

观察 (87)，大家是否也发现了张量积的影子？

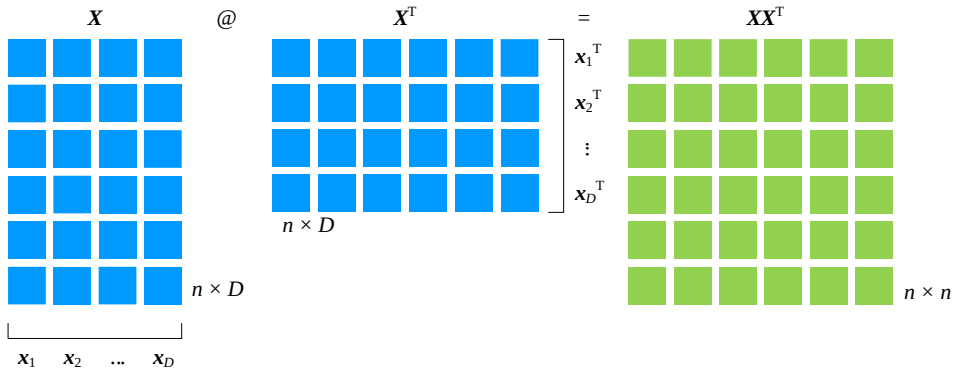



图 39. $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 运算过程

用张量积运算工具，(87) 可以写成：

$$
\begin{aligned}
\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} &= \boldsymbol{x}_1\boldsymbol{x}_1^{\mathrm{T}} + \boldsymbol{x}_2\boldsymbol{x}_2^{\mathrm{T}} + \cdots + \boldsymbol{x}_D\boldsymbol{x}_D^{\mathrm{T}} \\
&= \boldsymbol{x}_1 \otimes \boldsymbol{x}_1 + \boldsymbol{x}_2 \otimes \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_D \otimes \boldsymbol{x}_D \\
&= \sum_{i=1}^{D} \boldsymbol{x}_i \otimes \boldsymbol{x}_i
\end{aligned} \tag{88}
$$

### 元素平方和

此外，下式可以计算得到矩阵 $\boldsymbol{X}$ 的所有元素的平方和：

$$
\begin{aligned}
\operatorname{trace}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right) &= \operatorname{trace}\begin{bmatrix}
\boldsymbol{x}_1\cdot\boldsymbol{x}_1 & \boldsymbol{x}_1\cdot\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_1\cdot\boldsymbol{x}_D \\
\boldsymbol{x}_2\cdot\boldsymbol{x}_1 & \boldsymbol{x}_2\cdot\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_2\cdot\boldsymbol{x}_D \\
\vdots & \vdots & \ddots & \vdots \\
\boldsymbol{x}_D\cdot\boldsymbol{x}_1 & \boldsymbol{x}_D\cdot\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D\cdot\boldsymbol{x}_D
\end{bmatrix} \\
&= \boldsymbol{x}_1\cdot\boldsymbol{x}_1 + \boldsymbol{x}_2\cdot\boldsymbol{x}_2 + \cdots + \boldsymbol{x}_D\cdot\boldsymbol{x}_D \\
&= \sum_{i=1}^{n} x_{i,1}^2 + \sum_{i=1}^{n} x_{i,2}^2 + \cdots + \sum_{i=1}^{n} x_{i,D}^2 = \sum_{j=1}^{D}\sum_{i=1}^{n} x_{i,j}^2
\end{aligned} \tag{89}
$$

上一章讲解矩阵迹 (trace) 时提到，如果 $\boldsymbol{A}\boldsymbol{B}$ 和 $\boldsymbol{B}\boldsymbol{A}$ 都存在，$\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}) = \operatorname{tr}(\boldsymbol{B}\boldsymbol{A})$。也就是说，对于 (89)，下式成立：

$$
\operatorname{trace}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right) = \operatorname{trace}\left(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right) = \sum_{j=1}^{D}\sum_{i=1}^{n} x_{i,j}^2 \tag{90}
$$

本书后文还会在不同位置用到 $\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}) = \operatorname{tr}(\boldsymbol{B}\boldsymbol{A})$，请大家格外注意。

# 5.12 爱因斯坦求和约定

本书之前的所有矩阵运算都是适用于二阶情况，比如 $n \times D$ 的这种 $n$ 行、$D$ 列形式。在数据科学和机器学习很多实践中，我们不可避免地要处理高阶矩阵，比如图 40 所示三阶矩阵。Python 中 Xarray 专门用来存储和运算高阶矩阵。

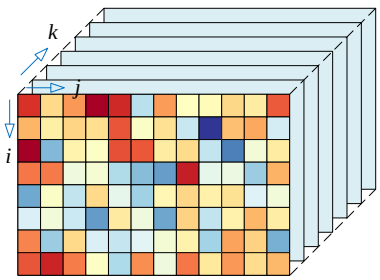



图 40. 三维数组，三阶矩阵

本节则要引出一种简洁表达高阶矩阵运算的数学工具——**爱因斯坦求和约定** (Einstein summation convention 或 Einstein notation)。本节特别介绍如何用 `numpy.einsum()` 函数完成本书前文介绍的主要线性代数运算。此外，PyTorch 中 `torch.einsum()` 函数原理和 `numpy.einsum()` 基本相同，本书不特别介绍。

使用 `numpy.einsum()` 时，大家记住一个要点——输入中重复的索引代表元素相乘，输出中消去的索引意味着相加。

举个例子，矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 相乘用 `numpy.einsum()` 函数可以写成：

```
np.einsum('ij,jk->ik', A, B)
```

"->" 之前分别为矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的索引，它们用逗号隔开。矩阵 $\boldsymbol{A}$ 行索引为 `i`，列索引为 `j`。矩阵 $\boldsymbol{B}$ 行索引为 `j`，列索引为 `k`。`j` 为重复索引，因此在这个方向上元素相乘。

"->" 之后为输出结果的索引。输出结果索引为 `ik`，没有 `j`，因此在 `j` 索引方向上存在求和运算。

表 1 总结如何使用 `numpy.einsum()` 完成常见线性代数运算。现在不需要大家掌握 `numpy.einsum()`。希望大家在日后用到爱因斯坦求和约定时，再回过头来深入学习。

表 1. 使用 `numpy.einsum()` 完成常见线性代数运算

| 运算 | 使用 `numpy.einsum()` 完成运算 |
|---|---|
| 向量 $\boldsymbol{a}$ 所有元素求和 | `np.einsum('ij->',a)`<br>`np.einsum('i->',a_1D)` |
| 等长向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的逐项积 | `np.einsum('ij,ij->ij',a,b)`<br>`np.einsum('i,i->i',a_1D,b_1D)` |
| 等长向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的向量内积 | `np.einsum('ij,ij->',a,b)`<br>`np.einsum('i,i->',a_1D,b_1D)` |
| 向量 $\boldsymbol{a}$ 和自身的张量积 | `np.einsum('ij,ji->ij',a,a)`<br>`np.einsum('i,j->ij',a_1D,a_1D)` |
| 向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的张量积 | `np.einsum('ij,ji->ij',a,b)`<br>`np.einsum('i,j->ij',a_1D,b_1D)` |
| 矩阵 $\boldsymbol{A}$ 的转置 | `np.einsum('ji',A)`<br>`np.einsum('ij->ji',A)` |
| 矩阵 $\boldsymbol{A}$ 所有元素求和 | `np.einsum('ij->',A)` |
| 矩阵 $\boldsymbol{A}$ 对每一列元素求和 | `np.einsum('ij->j',A)` |
| 矩阵 $\boldsymbol{A}$ 对每一行元素求和 | `np.einsum('ij->i',A)` |
| 提取方阵 $\boldsymbol{A}$ 的对角元素 | `np.einsum('ii->i',A)` |

| | |
|---|---|
| 计算方阵 $\boldsymbol{A}$ 的迹 trace($\boldsymbol{A}$) | `np.einsum('ii->',A)` |
| 计算矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 乘积 | `np.einsum('ij,jk->ik', A, B)` |
| 乘积 $\boldsymbol{AB}$ 结果所有元素求和 | `np.einsum('ij,jk->', A, B)` |
| 矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 相乘后再转置，即$(\boldsymbol{AB})^T$ | `np.einsum('ij,jk->ki', A, B)` |
| 形状相同矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 逐项积 | `np.einsum('ij,ij->ij', A, B)` |

表 1 中变量定义和运算都在 `Bk4_Ch5_01.py` 中。

本章全景展示了常见的矩阵乘法形态。每种形态的矩阵乘法都很重要，因此也不能用四幅图来总结本章重要内容。

大家想要活用线性代数这个宝库中的各种数学工具，熟练掌握矩阵乘法规则是绕不过去的一道门槛。再强调一次，大家在学习中，要试图从几何和数据这两个视角去理解不同的矩阵运算。这也是本书要特别强化的一点。

此外，本章针对矩阵乘法运算也没有给出任何代码，因为在 Numpy 中矩阵乘法常用运算符就是 @。本章最后介绍了爱因斯坦求和约定，不要求大家掌握。

矩阵乘法规则像是枷锁，它条条框框、冷酷无情、不容妥协；但是，在枷锁下，我们看到了矩阵乘法的另一面——无拘无束、血脉偾张、海纳百川。

希望大家一边学习本书剩余内容，一边能够不断回头看这一章内容，相信大家一定会和我一样，叹服于矩阵乘法展现出来的自由、包容，和纯粹的美。