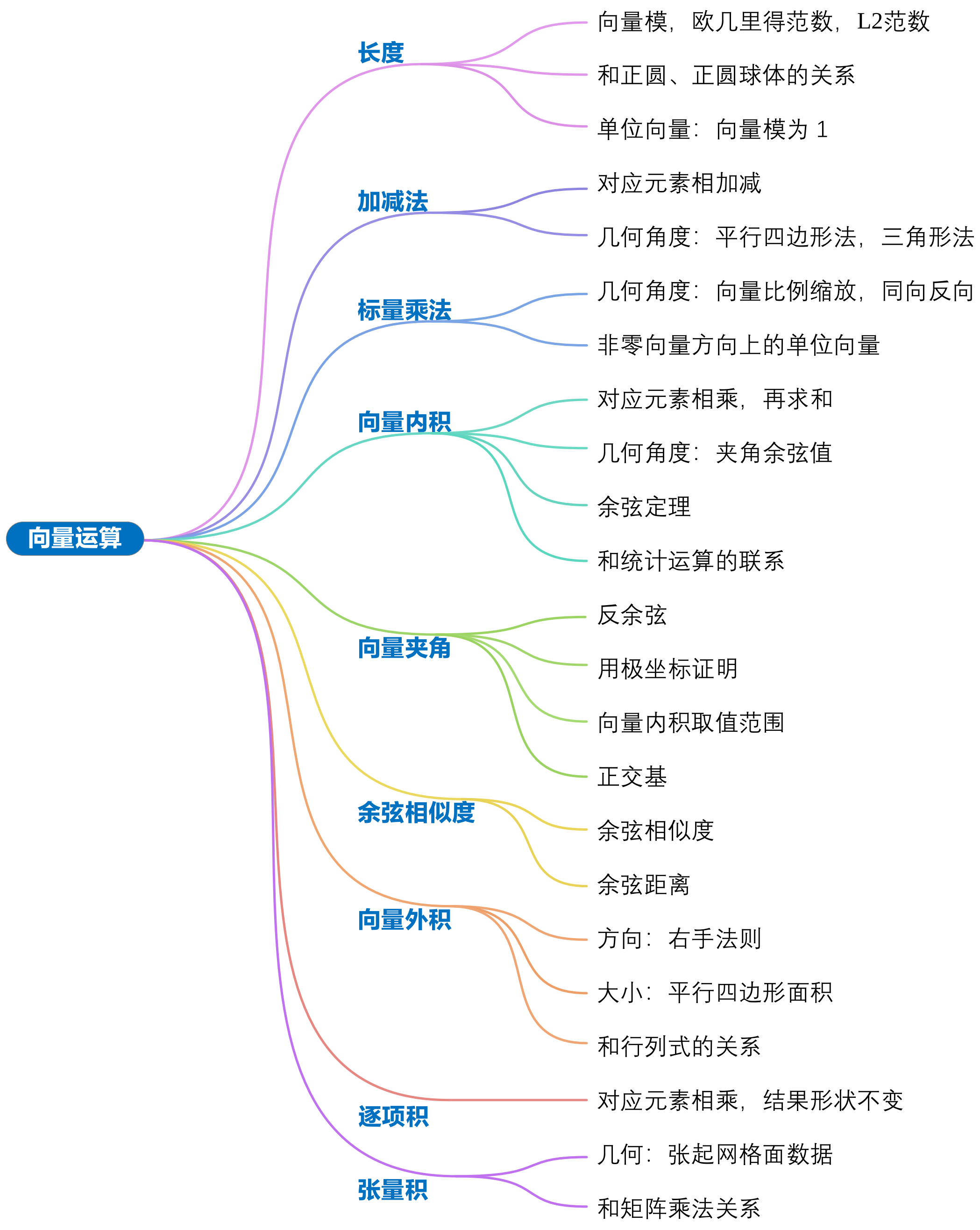
2 Vector Calculations
向量运算
从几何和数据角度解释

> 科学的每一次巨大进步，都源于颠覆性的大胆想象。
>
> ***Every great advance in science has issued from a new audacity of imagination.***
>
> —— 约翰·杜威 (John Dewey) | 美国著名哲学家、教育家、心理学家 | 1859 ~ 1952

- `numpy.arccos()` 反余弦
- `numpy.cross()` 计算列向量和行向量的向量积
- `numpy.dot()` 计算向量内积。值得注意的是，如果输入为一维数组，`numpy.dot()` 输出结果为向量内积；如果输入为矩阵，`numpy.dot()` 输出结果为矩阵乘积，相当于矩阵运算符@
- `numpy.linalg.norm()` 计算范数
- `numpy.multiply()` 计算向量逐项积
- `numpy.outer()` 计算外积，张量积
- `numpy.vdot()` 计算两个向量的向量内积。如果输入是矩阵，矩阵会按照先行后列顺序展开成向量之后，再计算向量内积
- `scipy.spatial.distance.cosine()` 计算余弦距离
- `seaborn.heatmap()` 绘制热图
- `sklearn.datasets.load_iris()` 加载鸢尾花数据


向量运算
长度
向量模，欧几里得范数，L2范数
和正圆、正圆球体的关系
单位向量：向量模为 1
加减法
对应元素相加减
几何角度：平行四边形法，三角形法
标量乘法
几何角度：向量比例缩放，同向反向
非零向量方向上的单位向量
向量内积
对应元素相乘，再求和
几何角度：夹角余弦值
余弦定理
和统计运算的联系
向量夹角
反余弦
用极坐标证明
向量内积取值范围
正交基
余弦相似度
余弦相似度
余弦距离
向量外积
方向：右手法则
大小：平行四边形面积
和行列式的关系
逐项积
对应元素相乘，结果形状不变
张量积
几何：张起网格面数据
和矩阵乘法关系

2.1 向量内积：结果为标量

本章是上一章的自然延续，将继续探讨常见向量运算。

向量内积 (inner product)，又叫**标量积** (scalar product)，或**点积** (dot product)、点乘。向量内积的运算结果为标量，而非向量。

给定如下 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个等长列向量：

$$\begin{aligned}\boldsymbol{a} &= \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{b} &= \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^{\mathrm{T}}\end{aligned} \tag{1}$$

列向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的内积定义如下：

$$\boldsymbol{a} \cdot \boldsymbol{b} = \langle \boldsymbol{a}, \boldsymbol{b} \rangle = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n \tag{2}$$

(2) 定义也适用于两个等长行向量计算内积。

从代数角度看内积，对两列/行数字中的每组对应元素求积，再对所有积求和，结果即为向量内积，如图 1 所示。

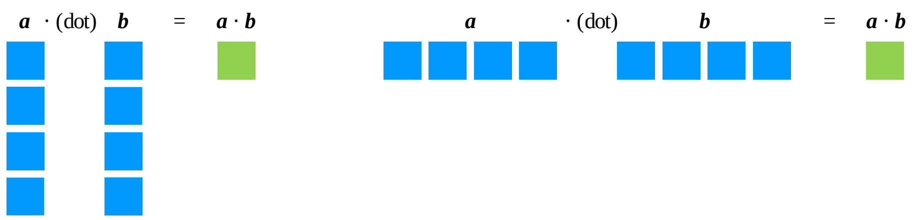


图 1. 向量内积运算

图 2 所示的两个列向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的内积：

$$\boldsymbol{a} \cdot \boldsymbol{b} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ -2 \end{bmatrix} = 4 \times 5 + 3 \times (-2) = 14 \tag{3}$$

Bk4_Ch2_01.py 计算上述向量内积。

```
# Bk4_Ch2_01.py

import numpy as np

a = np.array([[4, 3]])
b = np.array([[5, -2]])
```

```
a_dot_b = np.inner(a, b)

a_2 = np.array([[4], [3]])
b_2 = np.array([[5], [-2]])
a_dot_b_2 = a_2.T@b_2
```

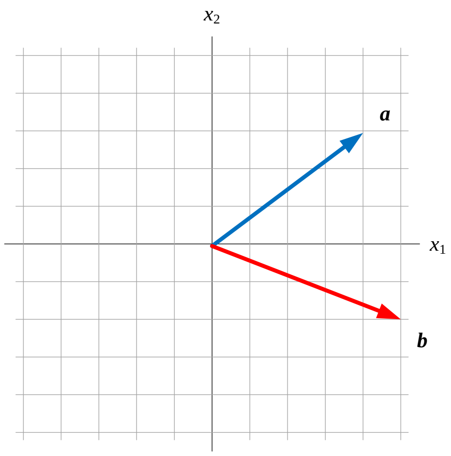


图 2. $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个平面向量

此外，还可以用 numpy.dot() 计算向量内积。值得注意的是，如果输入为一维数组，numpy.dot() 输出结果为内积。如果输入为矩阵，numpy.dot() 输出结果为矩阵乘积，相当于矩阵运算符@，比如 Bk4_Ch2_02.py 给出例子。

```
# Bk4_Ch2_02.py

import numpy as np
a = np.array([[2,3],
              [3,4]])

b = np.array([[3,4],
              [5,6]])

a_@_b = np.dot(a,b)
# a@b
```

numpy.vdot() 函数也可以计算两个向量内积。如果输入是矩阵，矩阵会按照先行后列顺序展开成向量之后，再计算向量内积。Bk4_Ch2_03.py 给出示例。

```
# Bk4_Ch2_03.py

import numpy as np
a = np.array([[1,2],
              [3,4]])
```

```
b = np.array([[3,4],
              [5,6]])

a_dot_b = np.vdot(a,b)
# [1,2,3,4]*[3,4,5,6].T
```

常用的向量内积性质如下：

$$
\begin{aligned}
&\boldsymbol{a}\cdot\boldsymbol{b}=\boldsymbol{b}\cdot\boldsymbol{a}\\
&\boldsymbol{a}\cdot(\boldsymbol{b}+\boldsymbol{c})=\boldsymbol{a}\cdot\boldsymbol{b}+\boldsymbol{a}\cdot\boldsymbol{c}\\
&(k\boldsymbol{a})\cdot(t\boldsymbol{b})=kt(\boldsymbol{a}\cdot\boldsymbol{b})
\end{aligned}
\tag{4}
$$

请读者格外注意以下几个向量内积运算和求和运算的关系：

$$
\begin{aligned}
&\boldsymbol{1}\cdot\boldsymbol{x}=x_1+x_2+\cdots x_n=\sum_{i=1}^{n}x_i\\
&\boldsymbol{x}\cdot\boldsymbol{x}=x_1^2+x_2^2+\cdots x_n^2=\sum_{i=1}^{n}x_i^2\\
&\boldsymbol{x}\cdot\boldsymbol{y}=x_1y_1+x_2y_2+\cdots x_ny_n=\sum_{i=1}^{n}x_iy_i
\end{aligned}
\tag{5}
$$

其中，

$$
\boldsymbol{x}=\begin{bmatrix}x_1 & x_2 & \cdots & x_n\end{bmatrix}^{\mathrm{T}},\quad \boldsymbol{1}=\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}^{\mathrm{T}},\quad \boldsymbol{y}=\begin{bmatrix}y_1 & y_2 & \cdots & y_n\end{bmatrix}^{\mathrm{T}}
\tag{6}
$$

几何视角

如图 3，从几何角度看，向量内积相当于两个向量的长度 (模、L2 范数) 与它们之间夹角余弦的积：

$$
\boldsymbol{a}\cdot\boldsymbol{b}=\|\boldsymbol{a}\|\|\boldsymbol{b}\|\cos\theta
\tag{7}
$$

此外，向量内积还可以从向量投影 (projection) 角度来解释，这是本书后续要介绍的内容。

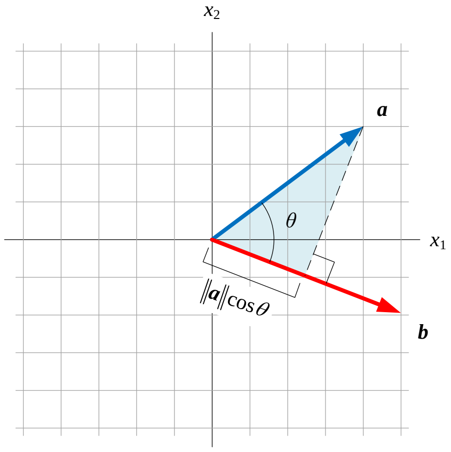


图 3. 向量内积和夹角余弦之间关系

$\boldsymbol{a}$ 的 L^2 范数也可以通过向量内积求得：

$$\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a} \cdot \boldsymbol{a}} \tag{8}$$

(8) 左右等式平方得到：

$$\|\boldsymbol{a}\|^2 = \boldsymbol{a} \cdot \boldsymbol{a} \tag{9}$$

柯西-施瓦茨不等式

观察 (7)，我们可以发现 $\cos\theta$ 的取值范围为 $[-1, 1]$，因此 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 内积取值范围如下：

$$-\|\boldsymbol{a}\|\|\boldsymbol{b}\| \le \boldsymbol{a} \cdot \boldsymbol{b} \le \|\boldsymbol{a}\|\|\boldsymbol{b}\| \tag{10}$$

图 4 所示为 7 个不同向量夹角状态。

$\theta = 0°$时，$\cos\theta = 1$，$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 同向，此时向量内积最大；$\theta = 180°$时，$\cos\theta = -1$，$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 反向，此时向量内积最小。

向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 垂直，即**正交** (orthogonal)，$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 夹角为 90°；$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 内积为 0：

$$\boldsymbol{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\|\|\boldsymbol{b}\|\cos 90° = 0 \tag{11}$$

当两个向量互相垂直时，向量内积为零。

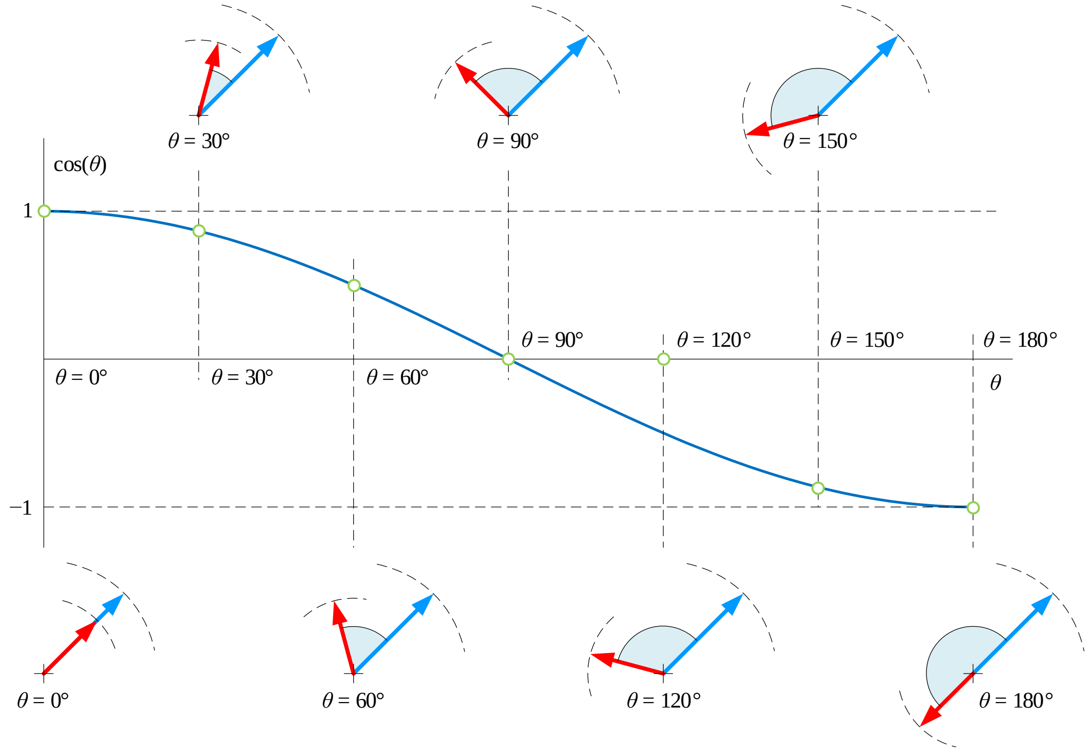


图 4. 向量夹角

有了以上分析，我们就可以引入一个重要的不等式——柯西-施瓦茨不等式 (Cauchy–Schwarz inequality)：

$$(\boldsymbol{a}\cdot\boldsymbol{b})^2 \le \|\boldsymbol{a}\|^2\|\boldsymbol{b}\|^2 \tag{12}$$

即：

$$|\boldsymbol{a}\cdot\boldsymbol{b}| \le \|\boldsymbol{a}\|\|\boldsymbol{b}\| \tag{13}$$

用尖括号来表达向量内积，(12) 可以写成：

$$\langle\boldsymbol{a},\boldsymbol{b}\rangle^2 \le \langle\boldsymbol{a},\boldsymbol{a}\rangle\langle\boldsymbol{b},\boldsymbol{b}\rangle \tag{14}$$

即：

$$|\langle\boldsymbol{a},\boldsymbol{b}\rangle| \le \|\boldsymbol{a}\|\|\boldsymbol{b}\| \tag{15}$$

在 $\mathbb{R}^n$ 空间中，上述不等式等价于：

$$\left(\sum_{i=1}^n a_i b_i\right)^2 \le \left(\sum_{i=1}^n a_i^2\right)\left(\sum_{i=1}^n b_i^2\right) \tag{16}$$

余弦定理

回忆从书第一本书讲解的**余弦定理** (law of cosines)：

$$c^2 = a^2 + b^2 - 2ab\cos\theta \tag{17}$$

其中，a、b 和 c 为图 5 所示三角形的三边的边长。下面，我们用余弦定理来用余弦定理推导 (7)。

如图 5 所示，将三角形三个边视作向量，将三个向量长度代入 (17)，可以得到：

$$\|\boldsymbol{c}\|^2 = \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 - 2\|\boldsymbol{a}\|\|\boldsymbol{b}\|\cos\theta \tag{18}$$

向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 之差为向量 $\boldsymbol{c}$：

$$\boldsymbol{c} = \boldsymbol{a} - \boldsymbol{b} \tag{19}$$

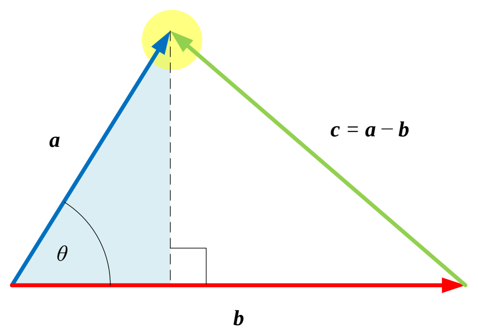


图 5. 余弦定理

(19) 等式左右分别和自身计算向量内积，得到如下等式：

$$\boldsymbol{c}\cdot\boldsymbol{c} = (\boldsymbol{a}-\boldsymbol{b})\cdot(\boldsymbol{a}-\boldsymbol{b}) \tag{20}$$

整理得到：

$$\begin{aligned}\boldsymbol{c}\cdot\boldsymbol{c} &= (\boldsymbol{a}-\boldsymbol{b})\cdot(\boldsymbol{a}-\boldsymbol{b}) = \boldsymbol{a}\cdot\boldsymbol{a} + \boldsymbol{b}\cdot\boldsymbol{b} - \boldsymbol{a}\cdot\boldsymbol{b} - \boldsymbol{b}\cdot\boldsymbol{a} \\ &= \boldsymbol{a}\cdot\boldsymbol{a} + \boldsymbol{b}\cdot\boldsymbol{b} - 2\boldsymbol{a}\cdot\boldsymbol{b}\end{aligned} \tag{21}$$

利用 (9)，(21) 可以写作：

$$\|\boldsymbol{c}\|^2 = \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 - 2\boldsymbol{a}\cdot\boldsymbol{b} \tag{22}$$

比较 (18) 和 (22)，可以得到：

$$\boldsymbol{a}\cdot\boldsymbol{b} = \|\boldsymbol{a}\|\|\boldsymbol{b}\|\cos\theta \tag{23}$$

在概率统计、数据分析、机器学习等领域，向量内积无处不在。下面举几个例子。

在多维空间中，给定 A 和 B 坐标如下：

$$A\left(a_1,a_2,\dots,a_n\right),\quad B\left(b_1,b_2,\dots,b_n\right) \tag{24}$$

计算 A 和 B 两点的距离 AB：

$$\begin{aligned} AB &= \sqrt{\left(a_1-b_1\right)^2+\left(a_2-b_2\right)^2+\cdots+\left(a_n-b_n\right)^2} \\ &= \sqrt{\sum_{i=1}^{n}\left(a_i-b_i\right)^2} \end{aligned} \tag{25}$$

用初始点位于原点的向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 分别代表 A 和 B 点，AB 距离就是 $\boldsymbol{a}-\boldsymbol{b}$ 的 L2 范数，也就是欧几里得距离：

$$AB = \left\|\boldsymbol{a}-\boldsymbol{b}\right\| = \sqrt{\left(\boldsymbol{a}-\boldsymbol{b}\right)\cdot\left(\boldsymbol{a}-\boldsymbol{b}\right)} = \sqrt{\boldsymbol{a}\cdot\boldsymbol{a}+\boldsymbol{b}\cdot\boldsymbol{b}-2\boldsymbol{a}\cdot\boldsymbol{b}} \tag{26}$$

回忆《概率统计》一册中介绍的样本方差公式，具体如下：

$$\operatorname{var}\left(X\right) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i-\mu\right)^2 \tag{27}$$

注意，对于样本方差，上式分母中 n 改为 $n-1$。

令 $\boldsymbol{x}$ 为，

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^{\mathrm{T}} \tag{28}$$

(27) 可以写成：

$$\operatorname{var}\left(X\right) = \frac{\left(\boldsymbol{x}-\mu\right)\cdot\left(\boldsymbol{x}-\mu\right)}{n} \tag{29}$$

根据广播原则，$\boldsymbol{x}-\mu$ 相当于向量 $\boldsymbol{x}$ 的每一个元素分别减去 μ。

回忆总体协方差公式：

$$\operatorname{cov}\left(X,Y\right) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i-\mu_X\right)\left(y_i-\mu_Y\right) \tag{30}$$

注意，对于样本协方差，上式分母中 n 改为 $n-1$。

同样利用向量内积运算法则，上式可以写成：

$$\operatorname{cov}\left(X,Y\right) = \frac{\left(\boldsymbol{x}-\mu_X\right)\cdot\left(\boldsymbol{y}-\mu_Y\right)}{n} \tag{31}$$

本书后续将从线性代数角度再和大家探讨概率统计相关内容。

2.2 向量夹角：反余弦

根据 (7)，可以得到向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 夹角余弦值：

$$\cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} \tag{32}$$

通过反余弦，可以得到向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 夹角：

$$\theta = \arccos\left(\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}\right) \tag{33}$$

arccos() 为反余弦函数，即从余弦值获得弧度。需要时可以进一步将弧度转化为角度。

图 2 中向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 夹角弧度值和角度值可以通过 `Bk4_Ch2_04.py` 计算。

```python
# Bk4_Ch2_04.py

import numpy as np

a, b = np.array([[4], [3]]), np.array([[5], [-2]])

# calculate cosine theta
cos_theta = (a.T @ b) / (np.linalg.norm(a,2) * np.linalg.norm(b,2))

# calculate theta in radian
cos_radian = np.arccos(cos_theta)

# convert radian to degree
cos_degree = cos_radian * ((180)/np.pi)
```

极坐标

下面，将向量放在极坐标中解释向量夹角余弦值。给定向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 坐标如下：

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{34}$$

向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 在极坐标中各自的角度为 θ_a 和 θ_b。角度 θ_a 和 θ_b 的正弦和余弦可以通过下式计算得到：

$$\begin{cases} \cos\theta_a = \dfrac{a_1}{\|\boldsymbol{a}\|}, & \sin\theta_a = \dfrac{a_2}{\|\boldsymbol{a}\|} \\ \cos\theta_b = \dfrac{b_1}{\|\boldsymbol{b}\|}, & \sin\theta_b = \dfrac{b_2}{\|\boldsymbol{b}\|} \end{cases} \tag{35}$$

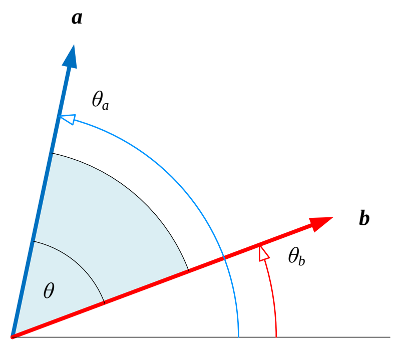


图 6. 极坐标中解释向量夹角

根据角的余弦和差恒等式，cos(θ)可以由 θ_a 和 θ_b 正余弦构造：

$$\begin{aligned}\cos(\theta) &= \cos(\theta_b - \theta_a) = \cos(\theta_b)\cos(\theta_a) + \sin(\theta_b)\sin(\theta_a) \\ &= \frac{a_1}{\|\boldsymbol{a}\|}\frac{b_1}{\|\boldsymbol{b}\|} + \frac{a_2}{\|\boldsymbol{a}\|}\frac{b_2}{\|\boldsymbol{b}\|} = \frac{a_1 b_1 + a_2 b_2}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}\end{aligned} \tag{36}$$

将 (35) 代入 (36) 得到：

$$\cos\theta = \frac{a_1}{\|\boldsymbol{a}\|}\frac{b_1}{\|\boldsymbol{b}\|} + \frac{a_2}{\|\boldsymbol{a}\|}\frac{b_2}{\|\boldsymbol{b}\|} = \frac{\overbrace{a_1 b_1 + a_2 b_2}^{\boldsymbol{a}\cdot\boldsymbol{b}}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} \tag{37}$$

相信大家已经在上式分母中看到向量内积。

单位向量

上一章介绍过某一向量方向上的单位向量这个概念，单位向量为我们提供了观察向量夹角余弦值的另外一个视角。

给定 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 向量，我们可以首先计算它们各自方向上的单位向量：

$$\hat{\boldsymbol{a}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}, \quad \hat{\boldsymbol{b}} = \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|} \tag{38}$$

两个单位向量的内积就是夹角的余弦值：

$$\hat{\boldsymbol{a}} \cdot \hat{\boldsymbol{b}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} \cdot \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|} = \cos\theta \tag{39}$$

正交单位向量

上一章介绍的平面直角坐标系中 $\boldsymbol{e}_1$ 和 $\boldsymbol{e}_2$ 分别代表为沿着 x_1 和 x_2 单位向量。它们相互垂直，也就是向量内积为 0：

$$\boldsymbol{e}_1 \cdot \boldsymbol{e}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{40}$$

也就是说，在一个平面上，单位向量 $\boldsymbol{e}_1$、$\boldsymbol{e}_2$ 相互垂直，它俩可以构造标准直角坐标系，具体如图 8 (a) 所示。

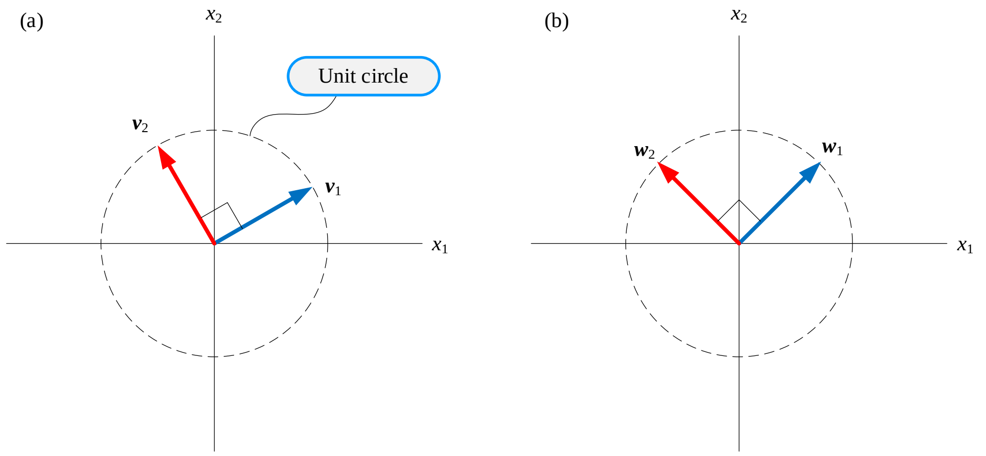


图 7. 正交单位向量

而平面上，成对正交单位向量有无数组，比如图 7 所示平面两组正交单位向量：

$$\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} = 0, \quad \boldsymbol{w}_1 \cdot \boldsymbol{w}_2 = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} = 0 \tag{41}$$

$\boldsymbol{v}_1$、$\boldsymbol{v}_2$ 也可以构造如图 8 (b) 所示直角坐标系。

类似地 $\boldsymbol{w}_1$、$\boldsymbol{w}_2$ 也可以构造如图 8 (c) 所示直角坐标系。也就是一个 $\mathbb{R}^2$ 平面上可以存在无数个直角坐标系。

[$\boldsymbol{e}_1$, $\boldsymbol{e}_2$]、[$\boldsymbol{v}_1$, $\boldsymbol{v}_2$]、[$\boldsymbol{w}_1$, $\boldsymbol{w}_2$] 都叫做 $\mathbb{R}^2$ 的正交基 (orthonormal basis)，而 [$\boldsymbol{e}_1$, $\boldsymbol{e}_2$] 有自己特别的名字——标准正交基 (standard orthonormal basis)。而且大家很快就会发现 [$\boldsymbol{e}_1$, $\boldsymbol{e}_2$] 旋转一定角度可以得到[$\boldsymbol{v}_1$, $\boldsymbol{v}_2$]、[$\boldsymbol{w}_1$, $\boldsymbol{w}_2$]。本书后续会深入介绍相关概念。

如图 8 所示，同一个向量 $\boldsymbol{a}$ 在三个直角坐标系中有不同的坐标值。向量 $\boldsymbol{a}$ 在图 8 (a) 所示直角坐标系的坐标值很容易确定 (2, 2)。目前我们还没有掌握足够的数学工具来计算向量 $\boldsymbol{a}$ 在图 8 (b) 和 (c) 两个直角坐标系中的坐标值。本书后续将继续深入讲解。

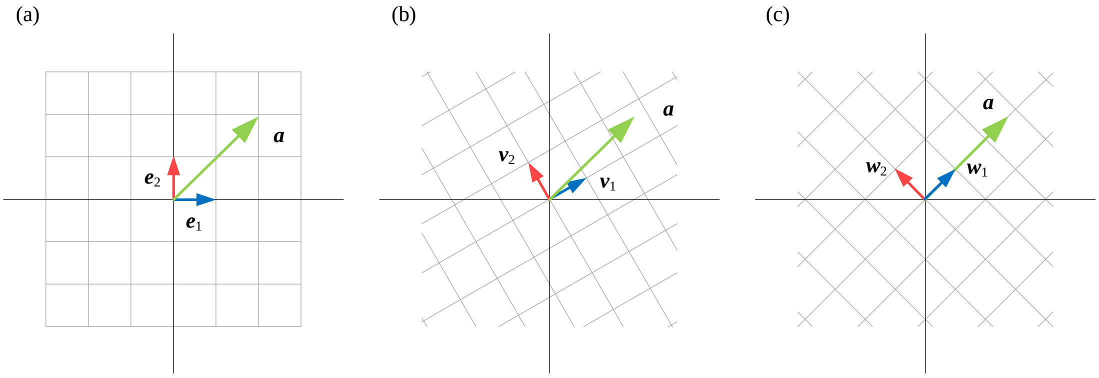


图 8. 向量 $\boldsymbol{a}$ 在三个不同的正交直角坐标系中位置

2.3 余弦相似度和余弦距离

机器学习中有一个重要的概念，叫做**余弦相似度** (cosine similarity)。余弦相似度用向量夹角的余弦值度量样本数据的相似性。

用 $k(\boldsymbol{x}, \boldsymbol{q})$ 来表达 $\boldsymbol{x}$ 和 $\boldsymbol{q}$ 两个列向量的余弦相似度，定义如下：

$$k(\boldsymbol{x},\boldsymbol{q}) = \frac{\boldsymbol{x}\cdot\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|} = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|} \tag{42}$$

上一节我们介绍过，如果两个向量方向相同，则夹角 θ 余弦值 $\cos(\theta) = 1$；如果，两个向量方向完全相反，夹角 θ 余弦值 $\cos(\theta) = -1$。

因此，余弦相似度取值范围在 $[-1, +1]$ 之间。另外，大家是否在余弦相似度中看到相关性系数的影子？

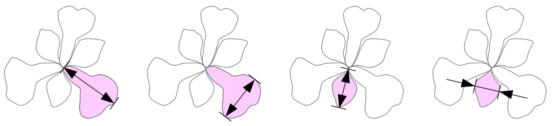
下面再介绍**余弦距离** (cosine distance)。余弦距离定义基于余弦相似度。用 $d(\boldsymbol{x}, \boldsymbol{q})$ 来表达 $\boldsymbol{x}$ 和 $\boldsymbol{q}$ 两个列向量的余弦距离，具体定义如下：

$$d(\boldsymbol{x},\boldsymbol{q}) = 1 - k(\boldsymbol{x},\boldsymbol{q}) = 1 - \frac{\boldsymbol{x}\cdot\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|} \tag{43}$$

上一章介绍的欧几里得距离，即 L^2 范数，是一种最常见的距离度量。本节介绍的余弦距离也是一种常见的距离度量。本书后续将逐步介绍常见距离度量，“距离”的内涵会不断丰富。

鸢尾花例子

图 9 给出鸢尾花四个样本数据。$\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(2)}$ 两个样本对应的鸢尾花都是 setosa 这一亚属。$\boldsymbol{x}^{(51)}$ 样本对应的鸢尾花为 versicolor 这一亚属；$\boldsymbol{x}^{(101)}$ 样本对应的鸢尾花为 virginica 这一亚属。



	Sepal length	Sepal width	Petal length	Petal width	
$\boldsymbol{x}^{(1)}$, 1	5.1	3.5	1.4	0.2	setosa
$\boldsymbol{x}^{(2)}$, 2	4.9	3	1.4	0.2	setosa
$\boldsymbol{x}^{(51)}$, 51	7	3.2	4.7	1.4	versicolor
$\boldsymbol{x}^{(101)}$, 101	6.3	3.3	6	2.5	virginica

图 9. 鸢尾花的四个样本数据

计算 $\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(2)}$ 两个向量余弦距离：

$$
\begin{aligned}
d\left(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}\right) &= 1-k\left(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}\right) \\
&= 1-\frac{5.1\times4.9+3.5\times3 \ + \ 1.4\times1.4 \ + \ 0.2\times0.2}{\sqrt{5.1^2+3.5^2 \ + \ 1.4^2 \ + \ 0.2^2}\times\sqrt{4.9^2+3^2 \ + \ 1.4^2 \ + \ 0.2^2}} \\
&= 1-\frac{37.49}{6.34507\times5.9169} \\
&= 1-0.99857=0.00142
\end{aligned}
\tag{44}
$$

同理，可以计算得到 $\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(51)}$， $\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(101)}$ 两各余弦距离：

$$
\begin{aligned}
d\left(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(51)}\right) &= 0.07161 \\
d\left(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(101)}\right) &= 0.13991
\end{aligned}
\tag{45}
$$

可以发现，$\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(2)}$两朵同属于 setosa 亚属的鸢尾花，余弦距离最近，也就是最为相似。

$\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(101)}$ 分别属于 setota 和 virginica 亚属，余弦距离最近，也就是极不不似。

Bk4_Ch2_05.py 可以完成上述计算。感兴趣的读者可以修改代码计算 $\boldsymbol{x}^{(51)}$ 和 $\boldsymbol{x}^{(101)}$ 的余弦距离，并结合样本标签分析结果。

```python
# Bk4_Ch2_05.py

from scipy.spatial import distance
from sklearn import datasets
import numpy as np

# import the iris data
iris = datasets.load_iris()

# Only use the first two features: sepal length, sepal width
X = iris.data[:, :]

# Extract 4 data points
x1_data = X[0,:]
x2_data = X[1,:]
```

```
x51_data = X[50,:]
x101_data = X[100,:]

# calculate cosine distance
x1_x2_cos_dist = distance.cosine(x1_data,x2_data)
x1_norm = np.linalg.norm(x1_data)
x2_norm = np.linalg.norm(x2_data)
x1_dot_x2 = x1_data.T@x2_data
x1_x2_cos = x1_dot_x2/x1_norm/x2_norm


x1_x51_cos_dist = distance.cosine(x1_data,x51_data)

x1_x101_cos_dist = distance.cosine(x1_data,x101_data)
```

2.4 向量外积：结果为向量

向量积 (vector product) 也叫**叉乘** (cross product) 或外积，向量积结果为向量。

$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 向量积，记做 $\boldsymbol{a} \times \boldsymbol{b}$。$\boldsymbol{a} \times \boldsymbol{b}$ 作为一个向量，我们需要了解它的方向和大小两个成分。

方向

如图 10 所示，$\boldsymbol{a} \times \boldsymbol{b}$ 方向分别垂直于向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，即 $\boldsymbol{a} \times \boldsymbol{b}$ 垂直于向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 构成平面。

向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 以及 $\boldsymbol{a} \times \boldsymbol{b}$ 三者关系可以用右手法则判断，如图 11 所示。图 11 这幅图中，我们可以看到 $\boldsymbol{a} \times \boldsymbol{b}$ 和 $\boldsymbol{b} \times \boldsymbol{a}$ 方向相反。

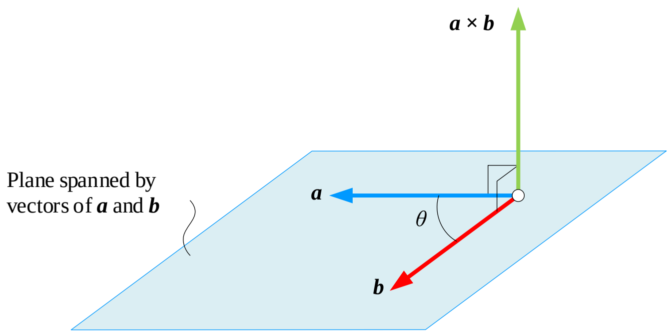


图 10. $\boldsymbol{a} \times \boldsymbol{b}$ 垂直于向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 构成平面

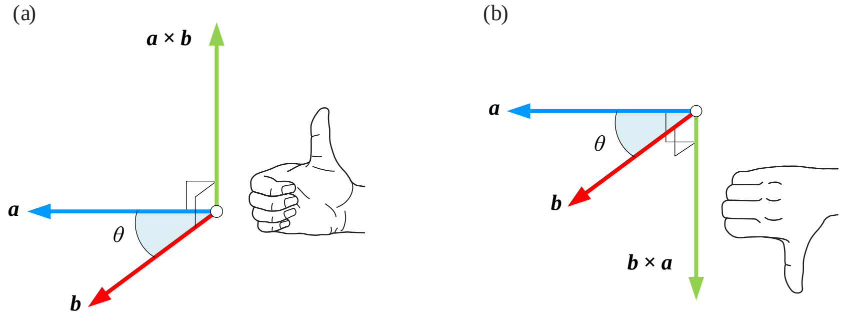


图 11. 向量叉乘右手定则

大小

$\boldsymbol{a} \times \boldsymbol{b}$ 模，也就是 $\boldsymbol{a} \times \boldsymbol{b}$ 向量积大小，通过下式获得：

$$\|\boldsymbol{a} \times \boldsymbol{b}\| = \|\boldsymbol{a}\|\|\boldsymbol{b}\|\sin(\theta) \tag{46}$$

其中 θ 为向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 夹角。如图 12 所示，从几何角度，向量积的模 $\|\boldsymbol{a} \times \boldsymbol{b}\|$ 相当于图中平行四边形的面积。

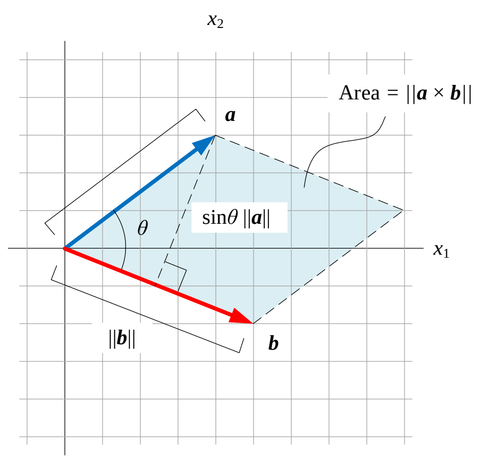


图 12. $\boldsymbol{a} \times \boldsymbol{b}$ 向量积的几何含义

如图 13 所示，从数据结构角度，形状相同的两个列向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 叉乘得到的向量积 $\boldsymbol{a} \times \boldsymbol{b}$ 形状不变。

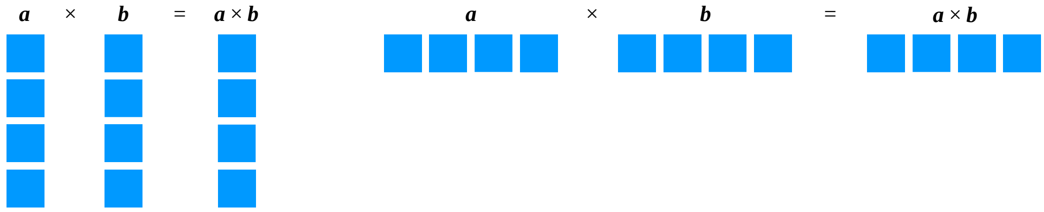


图 13. 数据结构角度看向量内积运算

正交向量之间的叉乘

如图 14 (a) 所示，空间直角坐标系中三个正交向量 $\boldsymbol{e}_1$ ($\boldsymbol{i}$) (x_1 轴正方向)、$\boldsymbol{e}_2$ ($\boldsymbol{j}$) (x_2 轴正方向) 和 $\boldsymbol{e}_3$ ($\boldsymbol{k}$) (x_3 轴正方向) 之间满足向量叉乘关系，如下：

$$\boldsymbol{i} \times \boldsymbol{j} = \boldsymbol{k}, \quad \boldsymbol{j} \times \boldsymbol{k} = \boldsymbol{i}, \quad \boldsymbol{k} \times \boldsymbol{i} = \boldsymbol{j} \tag{47}$$

图 14 (b) 展示以上三个等式中 $\boldsymbol{i}$、$\boldsymbol{j}$ 和 $\boldsymbol{k}$ 前后顺序关系。若调换它们顺序，得到以下三个运算式：

$$\boldsymbol{j} \times \boldsymbol{i} = -\boldsymbol{k}, \quad \boldsymbol{k} \times \boldsymbol{j} = -\boldsymbol{i}, \quad \boldsymbol{i} \times \boldsymbol{k} = -\boldsymbol{j} \tag{48}$$

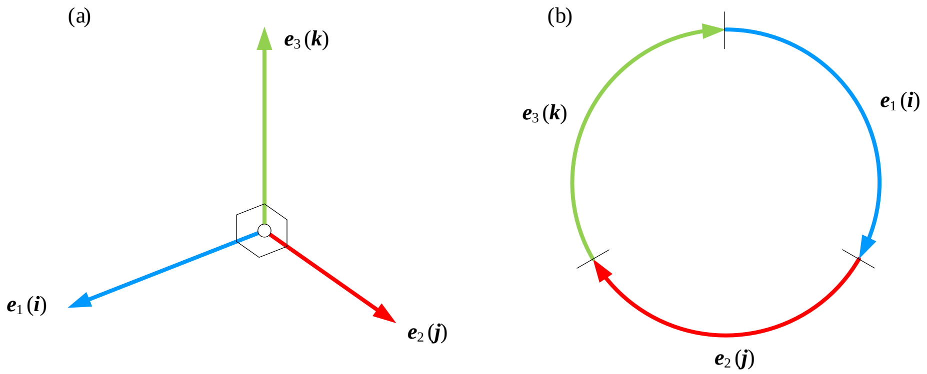


图 14. 三维空间正交单位向量基底之间关系

特别的，向量与自身叉乘等于 $\boldsymbol{0}$ 向量，如下：

$$\boldsymbol{i} \times \boldsymbol{i} = \boldsymbol{0}, \quad \boldsymbol{j} \times \boldsymbol{j} = \boldsymbol{0}, \quad \boldsymbol{k} \times \boldsymbol{k} = \boldsymbol{0} \tag{49}$$

任意两个向量的叉乘

用基底向量 $\boldsymbol{i}$、$\boldsymbol{j}$ 和 $\boldsymbol{k}$ 表达向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$：

$$\begin{aligned} \boldsymbol{a} &= a_1\boldsymbol{i} + a_2\boldsymbol{j} + a_3\boldsymbol{k} \\ \boldsymbol{b} &= b_1\boldsymbol{i} + b_2\boldsymbol{j} + b_3\boldsymbol{k} \end{aligned} \tag{50}$$

整理向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 叉乘，如下：

$$\begin{aligned}\boldsymbol{a}\times\boldsymbol{b} &= (a_1\boldsymbol{i}+a_2\boldsymbol{j}+a_3\boldsymbol{k})\times(b_1\boldsymbol{i}+b_2\boldsymbol{j}+b_3\boldsymbol{k}) \\ &= a_1b_1(\boldsymbol{i}\times\boldsymbol{i})+a_1b_2(\boldsymbol{i}\times\boldsymbol{j})+a_1b_3(\boldsymbol{i}\times\boldsymbol{k}) \\ &\quad +a_2b_1(\boldsymbol{j}\times\boldsymbol{i})+a_2b_2(\boldsymbol{j}\times\boldsymbol{j})+a_2b_3(\boldsymbol{j}\times\boldsymbol{k}) \\ &\quad +a_3b_1(\boldsymbol{k}\times\boldsymbol{i})+a_3b_2(\boldsymbol{k}\times\boldsymbol{j})+a_3b_3(\boldsymbol{k}\times\boldsymbol{k}) \\ &= (a_2b_3-a_3b_2)\boldsymbol{i}+(a_3b_1-a_1b_3)\boldsymbol{j}+(a_1b_2-a_2b_1)\boldsymbol{k}\end{aligned} \tag{51}$$

$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 叉乘还可以通过行列式求解，我们将在本书后续内容讲解。

下列为叉乘运算常见性质：

$$\begin{aligned}&\boldsymbol{a}\times\boldsymbol{a}=\boldsymbol{0} \\ &\boldsymbol{a}\times(\boldsymbol{b}+\boldsymbol{c})=\boldsymbol{a}\times\boldsymbol{b}+\boldsymbol{a}\times\boldsymbol{c} \\ &(\boldsymbol{a}+\boldsymbol{b})\times\boldsymbol{c}=\boldsymbol{a}\times\boldsymbol{b}+\boldsymbol{a}\times\boldsymbol{c} \\ &\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{c})\neq(\boldsymbol{a}\times\boldsymbol{b})\times\boldsymbol{c} \\ &k(\boldsymbol{a}\times\boldsymbol{b})=k(\boldsymbol{a})\times\boldsymbol{b}=\boldsymbol{a}\times(k\boldsymbol{b}) \\ &\boldsymbol{a}\cdot(\boldsymbol{b}\times\boldsymbol{c})=(\boldsymbol{a}\times\boldsymbol{b})\cdot\boldsymbol{c}\end{aligned} \tag{52}$$

举个例子

下面结合代码计算 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个向量叉乘：

$$\boldsymbol{a}=\begin{bmatrix}-2\\1\\1\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}1\\-2\\-1\end{bmatrix} \tag{53}$$

即

$$\begin{aligned}\boldsymbol{a}&=-2\boldsymbol{i}+\boldsymbol{j}+\boldsymbol{k}\\ \boldsymbol{b}&=\boldsymbol{i}-2\boldsymbol{j}-\boldsymbol{k}\end{aligned} \tag{54}$$

$\boldsymbol{a}\times\boldsymbol{b}$ 结果如下：

$$\boldsymbol{a}\times\boldsymbol{b}=\begin{bmatrix}1\\-1\\3\end{bmatrix} \tag{55}$$

即

$$\boldsymbol{a}\times\boldsymbol{b}=\boldsymbol{i}-\boldsymbol{j}+3\boldsymbol{k} \tag{56}$$

numpy.cross() 函数可以用来计算列向量和行向量的向量积。Bk4_Ch2_06.py 计算上例。

```python
# Bk4_Ch2_06.py

import numpy as np
a = np.array([-2, 1, 1])
b = np.array([1, -2, -1])
# a = [-2, 1, 1]
# b = [1, -2, -1]

# calculate cross product of row vectors
a_cross_b = np.cross(a, b)

a_col = np.array([[-2], [1], [1]])
b_col = np.array([[1], [-2], [-1]])

# calculate cross product of column vectors
a_cross_b_col = np.cross(a_col,b_col,axis=0)
```

2.5 逐项积：对应元素分别相乘

元素乘积 (element-wise multiplication)，也称为**阿达玛乘积** (Hadamard product) 或**逐项积** (piecewise product)，即两个形状相同的矩阵，对应元素相乘得到同样形状的矩阵。向量是一种特殊矩阵，阿达玛乘积也适用于向量。图 15 给出的是从数据角度看逐项积运算。

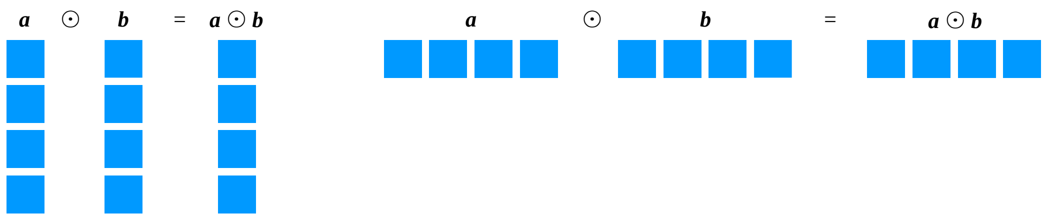


图 15. 向量逐项积运算

给定如下 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 两个列向量：

$$\begin{aligned}\boldsymbol{a} &= \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{b} &= \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^{\mathrm{T}}\end{aligned} \tag{57}$$

列向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的逐项积定义如下：

$$\boldsymbol{a} \odot \boldsymbol{b} = \begin{bmatrix} a_1 b_1 & a_2 b_2 & \cdots & a_n b_n \end{bmatrix}^{\mathrm{T}} \tag{58}$$

给定如 (53) 两个列向量，它们的逐项积为：

$$\boldsymbol{a} \odot \boldsymbol{b} = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} \odot \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ -1 \end{bmatrix} \tag{59}$$

Bk4_Ch2_07.py 计算行向量逐项积。

```python
# Bk4_Ch2_07.py

import numpy as np
a = np.array([-2, 1, 1])
b = np.array([1, -2, -1])
# a = [-2, 1, 1]
# b = [1, -2, -1]


# calculate element-wise product of row vectors
a_times_b = np.multiply(a, b)
a_times_b_2 = a*b

a_col = np.array([[-2], [1], [1]])
b_col = np.array([[1], [-2], [-1]])

# calculate element-wise product of column vectors
a_times_b_col = np.multiply(a_col,b_col)
a_times_b_col_2 = a_col*b_col
```

2.6 向量张量积：张起网格面

张量积 (tensor product) 又叫**外积** (outer product)，两个列向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 张量积 $\boldsymbol{a} \otimes \boldsymbol{b}$ 定义如下：

$$\boldsymbol{a} \otimes \boldsymbol{b} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}_{n\times 1} \otimes \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m\times 1} = \boldsymbol{a}\boldsymbol{b}^{\mathrm{T}} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} a_1b_1 & a_1b_2 & \cdots & a_1b_m \\ a_2b_1 & a_2b_2 & \cdots & a_2b_m \\ \vdots & \vdots & \ddots & \vdots \\ a_nb_1 & a_nb_2 & \cdots & a_nb_m \end{bmatrix}_{n\times m} \tag{60}$$

图 16 (a) 给出上述运算的示意图。

注意，上式中 $\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}$ 为向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}^{\mathrm{T}}$ 的乘法运算，它遵循矩阵乘法规则。矩阵乘法是我们下两章要介绍的内容。

观察 (60)，可以发现 $\boldsymbol{a} \otimes \boldsymbol{b}$ 可以写成两种形式：

$$\boldsymbol{a} \otimes \boldsymbol{b} = \begin{bmatrix} a_1\boldsymbol{b}^{\mathrm{T}} \\ a_2\boldsymbol{b}^{\mathrm{T}} \\ \vdots \\ a_n\boldsymbol{b}^{\mathrm{T}} \end{bmatrix}_{n\times 1} = \begin{bmatrix} b_1\boldsymbol{a} & b_2\boldsymbol{a} & \cdots & b_1\boldsymbol{a} \end{bmatrix} \tag{61}$$

第一种形式相当于，$\boldsymbol{b}^{\mathrm{T}}$ 先按不同比例 (a_i) 缩放得到 $a_i\boldsymbol{b}^{\mathrm{T}}$，再上下叠加。

第二种形式相当于，$\boldsymbol{a}$ 先按不同比例 (b_j) 缩放得到 $b_j\boldsymbol{a}$，再左右排列。

向量 $\boldsymbol{a}$ 和其自身张量积 $\boldsymbol{a}\otimes\boldsymbol{a}$ 定义如下：

$$\boldsymbol{a}\otimes\boldsymbol{a}=\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}_{n\times 1}\otimes\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}_{n\times 1}=\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}=\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}^{\mathrm{T}}=\begin{bmatrix}a_1a_1 & a_1a_2 & \cdots & a_1a_n\\a_2a_1 & a_2a_2 & \cdots & a_2a_n\\\vdots & \vdots & \ddots & \vdots\\a_na_1 & a_na_2 & \cdots & a_na_n\end{bmatrix}\tag{62}$$

图 16 (b) 给出上述运算的示意图。

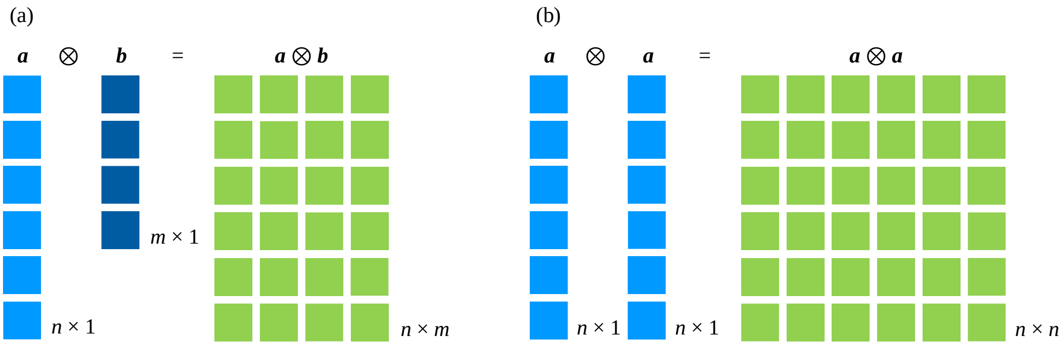


图 16. 向量张量积

请大家注意张量积一些常见性质：

$$\begin{aligned}&(\boldsymbol{a}\otimes\boldsymbol{b})^{\mathrm{T}}=\boldsymbol{b}\otimes\boldsymbol{a}\\&(\boldsymbol{a}+\boldsymbol{b})\otimes\boldsymbol{v}=\boldsymbol{a}\otimes\boldsymbol{v}+\boldsymbol{b}\otimes\boldsymbol{v}\\&\boldsymbol{v}\otimes(\boldsymbol{a}+\boldsymbol{b})=\boldsymbol{v}\otimes\boldsymbol{a}+\boldsymbol{v}\otimes\boldsymbol{b}\\&t(\boldsymbol{a}\otimes\boldsymbol{b})=(t\boldsymbol{a})\otimes\boldsymbol{b}=\boldsymbol{a}\otimes(t\boldsymbol{b})\\&(\boldsymbol{a}\otimes\boldsymbol{b})\otimes\boldsymbol{v}=\boldsymbol{a}\otimes(\boldsymbol{b}\otimes\boldsymbol{v})\end{aligned}\tag{63}$$

几何视角

图 17 所示为从几何图像角度解释向量的张量积。向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 相当于两个维度上的支撑框架，两者的张量积则“张起”一个网格面数据 $\boldsymbol{a}\otimes\boldsymbol{b}$。

当我们关注 $\boldsymbol{b}$ 方向时，网格面沿同一方向的每一条曲线都类似 $\boldsymbol{b}$，唯一的差别是高度上存在一定比例的缩放，这个比值就是 a_i。a_i 是向量 $\boldsymbol{a}$ 中的一个元素。

同理，观察 $\boldsymbol{a}$ 方向的网格面，每一条曲线都类似 $\boldsymbol{a}$。向量 $\boldsymbol{b}$ 的某一元素 b_j 提供曲线高度的缩放系数。

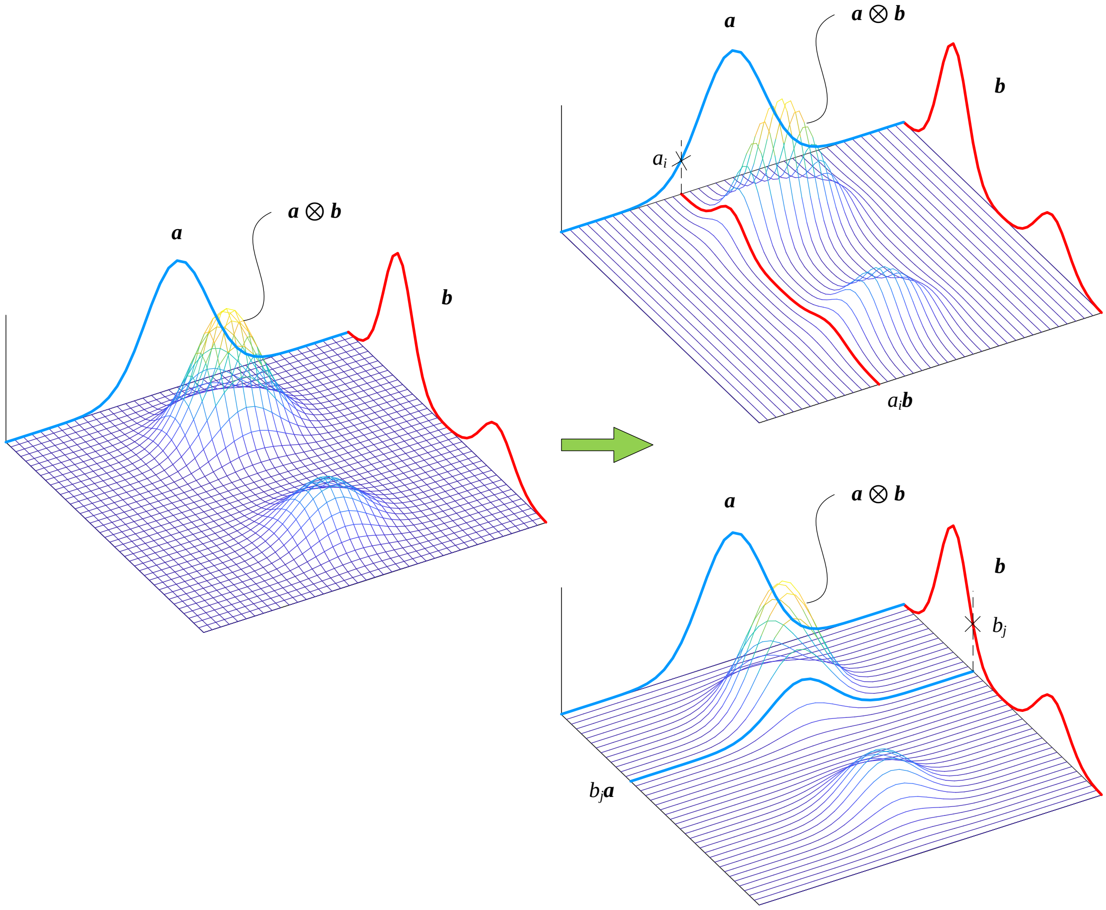


图 17. 从几何图像角度解释向量张量积

数据视角

下面再从数据角度可视化张量积运算。给定列向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 分别为:

$$\begin{aligned} \boldsymbol{a} &= \begin{bmatrix} 0.5 & -0.7 & 1 & 0.25 & -0.6 & -1 \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{b} &= \begin{bmatrix} -0.8 & 0.5 & -0.6 & 0.9 \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{64}$$

图 18 所示为张量积 $\boldsymbol{a} \otimes \boldsymbol{b}$ 结果热图，形状为 5 × 4。

如图 19 所示，$\boldsymbol{a} \otimes \boldsymbol{b}$ 的每一列都和 $\boldsymbol{a}$“相似”，也就是说它们之间呈现倍数关系。

类似地，如图 20 所示，$\boldsymbol{a} \otimes \boldsymbol{b}$ 等价于 $\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}$，因此 $\boldsymbol{a} \otimes \boldsymbol{b}$ 每一行都和 $\boldsymbol{b}^{\mathrm{T}}$“相似”，也呈现倍数关系。

本书之后会聊到向量的秩 (rank)，大家就会知道 $\boldsymbol{a} \otimes \boldsymbol{b}$ 的秩为 1，就是因为这种“相似”。

图 21 所示为张量积 $\boldsymbol{a} \otimes \boldsymbol{a}$ 结果热图，形状为 5 × 5 方阵。显然，$\boldsymbol{a} \otimes \boldsymbol{a}$ 为对称矩阵。

图 22 所示为张量积 $\boldsymbol{b} \otimes \boldsymbol{b}$ 结果热图，形状为 4 × 4 对称方阵。

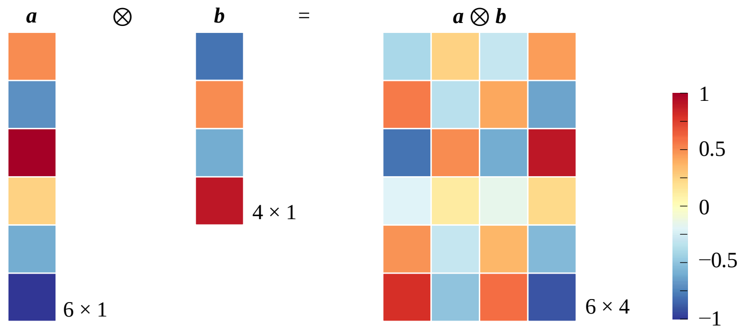


图 18.张量积 $\boldsymbol{a} \otimes \boldsymbol{b}$

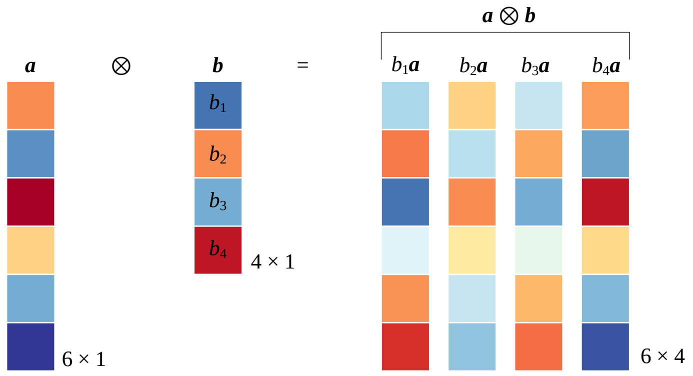


图 19. $\boldsymbol{a} \otimes \boldsymbol{b}$ 的每一列都和 $\boldsymbol{a}$“相似”

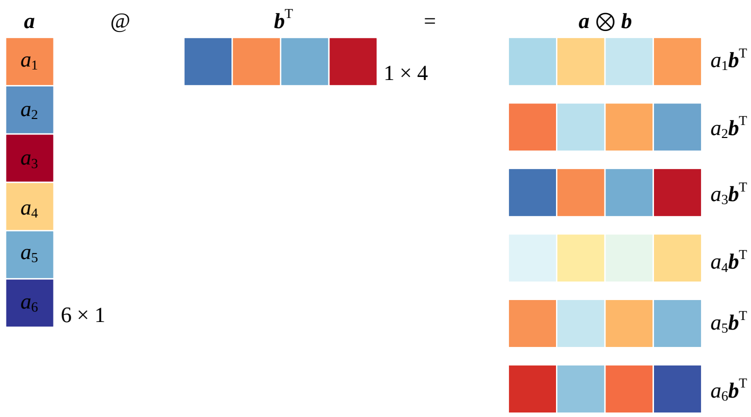


图 20. $\boldsymbol{a} \otimes \boldsymbol{b}$ 的每一行都和 $\boldsymbol{b}$“相似”

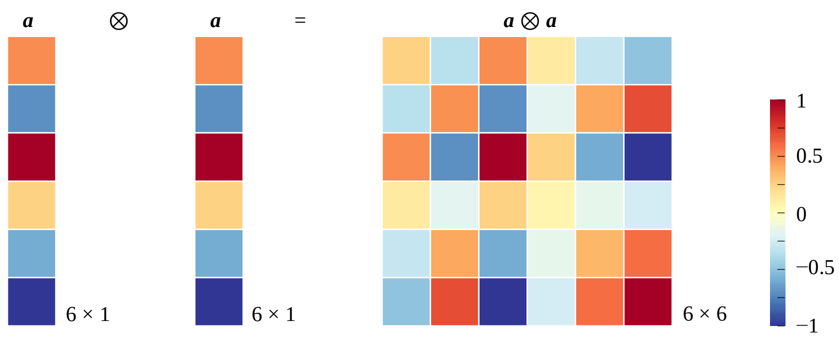


图 21. 向量张量积 $\boldsymbol{a} \otimes \boldsymbol{a}$

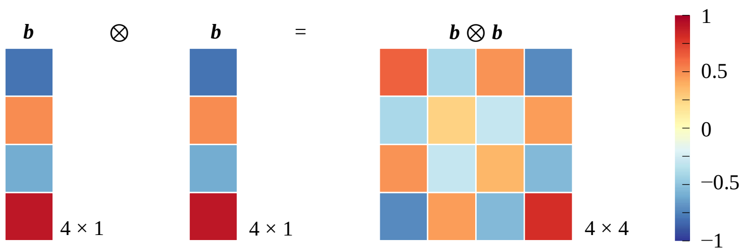


图 22. 向量张量积 $\boldsymbol{b} \otimes \boldsymbol{b}$

Bk4_Ch2_08.py 绘制图 18、图 21 和图 22。

```python
# Bk4_Ch2_08.py

import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

def plot_heatmap(x,title):

    fig, ax = plt.subplots()
    ax = sns.heatmap(x,
                     cmap='RdYlBu_r',
                     cbar_kws={"orientation": "horizontal"}, vmin=-1, vmax=1)
    ax.set_aspect("equal")
    plt.title(title)

a = np.array([[0.5],[-0.7],[1],[0.25],[-0.6],[-1]])
b = np.array([[-0.8],[0.5],[-0.6],[0.9]])

a_outer_b = np.outer(a, b)
a_outer_a = np.outer(a, a)
b_outer_b = np.outer(b, b)

# Visualizations
plot_heatmap(a,'a')

plot_heatmap(b,'b')

plot_heatmap(a_outer_b,'a outer b')

plot_heatmap(a_outer_a,'a outer a')
```

```
plot_heatmap(b_outer_b,'b outer b')
```

《概率统计》一本中介绍过，如果两个离散随机变量 X 和 Y 独立，联合概率 $p_{X,Y}(x,y)$ 等于 $p_X(x)$ 和 $p_Y(y)$ 两个边缘概率质量函数 PMF 乘积：

$$\underbrace{p_{X,Y}(x,y)}_{\text{Joint}} = \underbrace{p_X(x)}_{\text{Marginal}} \cdot \underbrace{p_Y(y)}_{\text{Marginal}} \tag{65}$$

如图 23 所示，$p_X(x)$ 和 $p_Y(y)$ 可以用一维火柴梗图可视化，$p_{X,Y}(x,y)$ 用二维火柴梗图展示。

从线性代数角度，$p_X(x)$ 和 $p_Y(y)$ 在 x 和 y 分别取不同值时，相当于是向量；而 x 和 y 分别取不同值时，$p_{X,Y}(x,y)$ 相当于是矩阵。

X 和 Y 独立时，$p_{X,Y}(x,y)$ 值的矩阵就是 $p_Y(y)$ 和 $p_X(x)$ 两个向量的张量积。

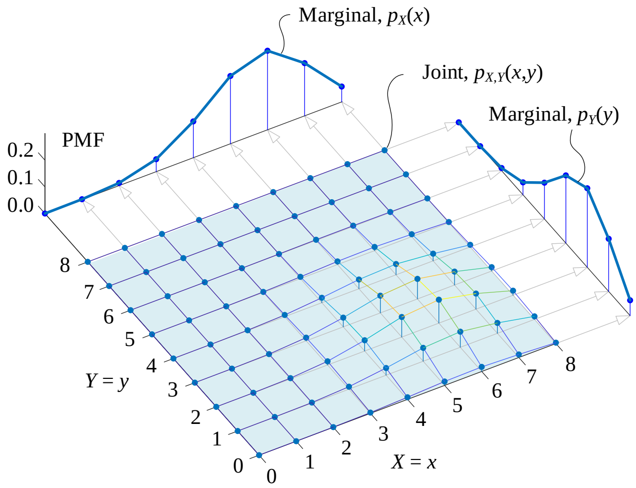


图 23. 离散随机变量独立条件下，联合概率 $p_{X,Y}(x,y)$ 等于 $p_Y(y)$ 和 $p_X(x)$ 两个边缘概率乘积

本章聊了聊常见的几种向量运算。也是用四幅图总结本章主要内容。向量内积的结果是个标量，请大家格外注意向量内积和矩阵乘法联系，以及和 Σ 求和运算之间的关系。

从几何视角看向量内积特别重要，请大家格外关注向量夹角余弦值、余弦定理、余弦相似度、余弦距离，以及本书后续要讲的标量投影、向量投影、协方差、相关性系数等数学概念之间的关系。

向量的外积结果还是个向量，这个向量垂直于原来两个向量构成的平面。

几何视角下，张量积像是张起一个网格面。张量积在机器学习和数据科学算法中应用特别广泛，有关这个运算的性质我们会慢慢展开讲解。

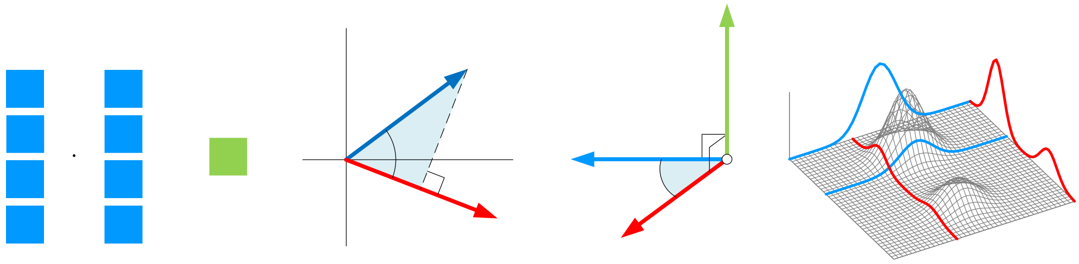


图 24. 总结本章重要内容的四副图