

# 2

## Vector Calculations

# 向量运算

从几何和数据角度解释



几何——指向真理之乡，创造哲学之魂。

*Geometry will draw the soul toward truth and create the spirit of philosophy.*

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



- ◀ matplotlib.pyplot.quiver() 绘制箭头图
- ◀ numpy.add() 向量/矩阵加法
- ◀ numpy.arccos() 计算反余弦
- ◀ numpy.array([[4,3]]) 构造行向量，注意双重方括号
- ◀ numpy.array([[4,3]]).T 行向量转置得到列向量，注意双重方括号
- ◀ numpy.array([4], [3]) 构造列向量，注意双重方括号
- ◀ numpy.array([4, 3])[:, None] 构造列向量
- ◀ numpy.array([4, 3])[:, numpy.newaxis] 构造列向量
- ◀ numpy.array([4, 3])[None, :] 构造行向量
- ◀ numpy.array([4, 3])[numpy.newaxis, :] 构造行向量
- ◀ numpy.array([4,3]) 构造一维数组，严格来说不是行向量
- ◀ numpy.array([4,3]).reshape((-1, 1)) 构造列向量
- ◀ numpy.array([4,3]).reshape((1, -1)) 构造行向量
- ◀ numpy.array([4,3], ndmin=2) 构造行向量
- ◀ numpy.cross() 计算列向量或行向量的向量积
- ◀ 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() 默认计算 L2 范数
- ◀ numpy.multiply() 计算向量逐项积
- ◀ numpy.ones() 生成全 1 向量/矩阵
- ◀ numpy.outer() 计算外积、张量积
- ◀ numpy.r\_[ ] 将一系列数组合并；'r' 设定结果以行向量（默认）展示，比如  
numpy.r\_[numpy.array([1,2]), 0, 0, numpy.array([4,5])] 默认产生行向量
- ◀ numpy.r\_['c', [4,3]] 构造列向量
- ◀ numpy.subtract() 向量/矩阵减法
- ◀ 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
- ◀ numpy.zeros() 生成全 0 向量/矩阵
- ◀ scipy.spatial.distance.cosine() 计算余弦距离
- ◀ zip(\*) 用于将可迭代的对象作为参数，将对象中对应的元素打包成一个个元组，然后返回由这些元组组成的列表。\*代表解包，返回的每一个都是元组类型，而并非是原来的数据类型

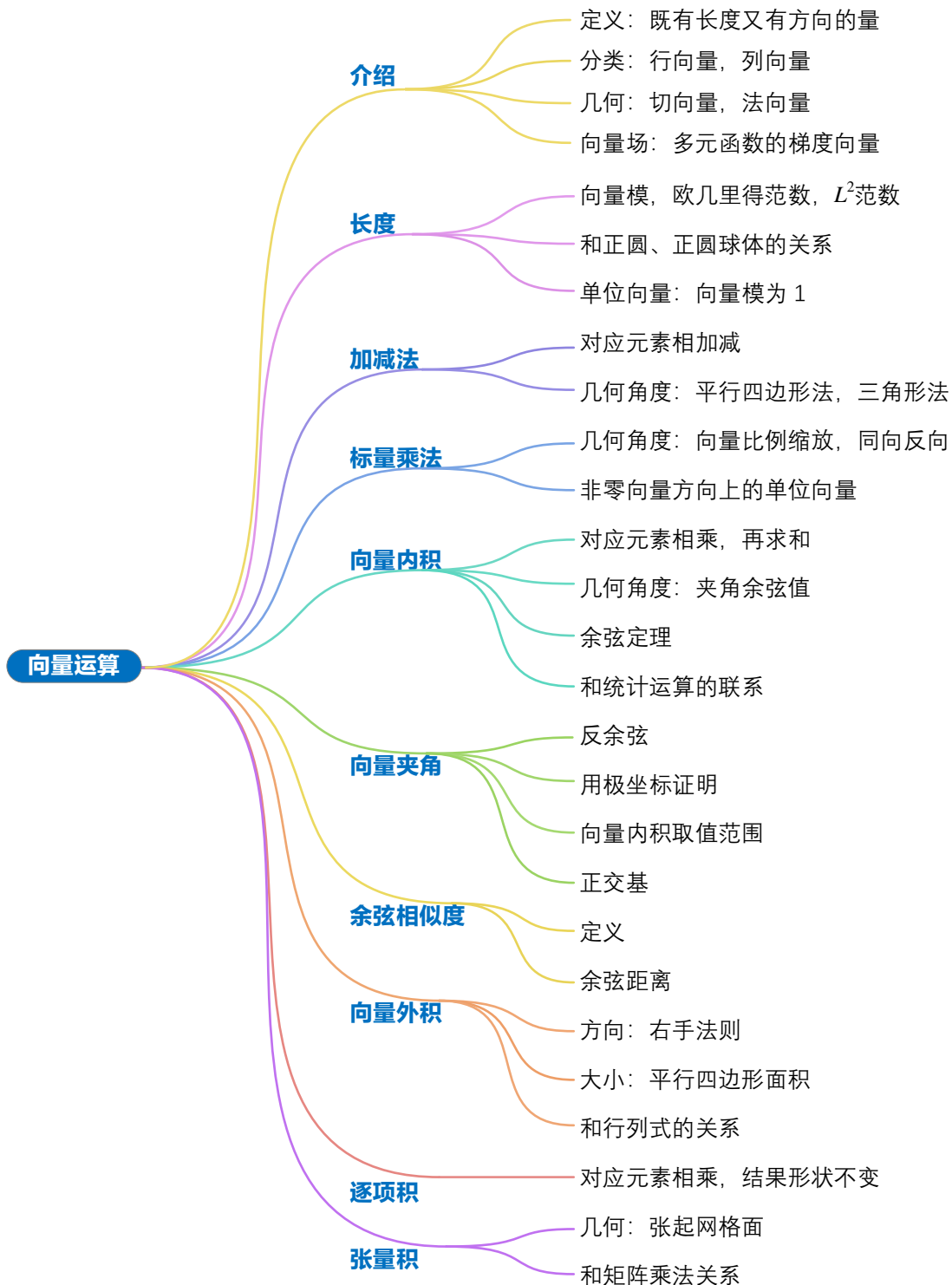
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

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

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

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



## 2.1 向量：多面手

### 几何视角

如图 1 所示，平面上，向量是**有方向的线段** (directed line segment)。**线段的长度代表向量的大小** (the length of the line segment represents the magnitude of the vector)。**箭头代表向量的方向** (the direction of the arrowhead indicates the direction of the vector)。

再次强调，本书中向量符号采用加粗、斜体、小写字母，比如  $\mathbf{a}$ ；矩阵符号则采用加粗、斜体、大写字母，比如  $\mathbf{A}$ 。

图 1 中，向量  $\mathbf{a}$  的**起点** (initial point) 是原点  $O$ ，向量的**终点** (terminal point) 是  $A$ 。如果向量的起点和终点相同，向量则为**零向量** (zero vector)，可以表示为  $\mathbf{0}$ 。

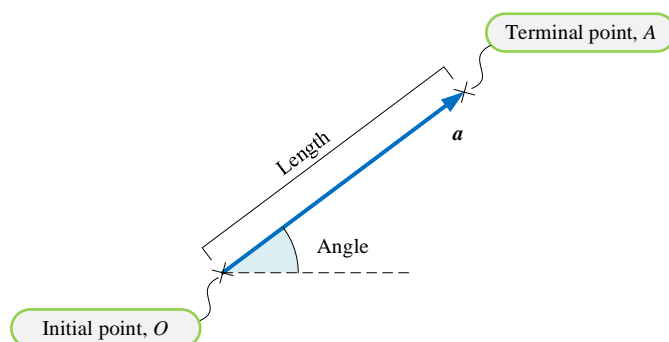


图 1. 向量起点、终点、大小和方向

图 2 给出的是几种向量的类型。

和起点无关的向量叫做**自由向量** (free vector)，如图 2 (a)。和起点有关的向量被称作，**固定向量** (fixed vector)，如图 2 (b) 和 (c)。称方向上沿着某一个特定直线的向量为**滑动向量** (sliding vector)，如图 2 (d)。

没有特别说明时，本书的向量一般是固定向量，且起点一般都在原点，除非特别说明。

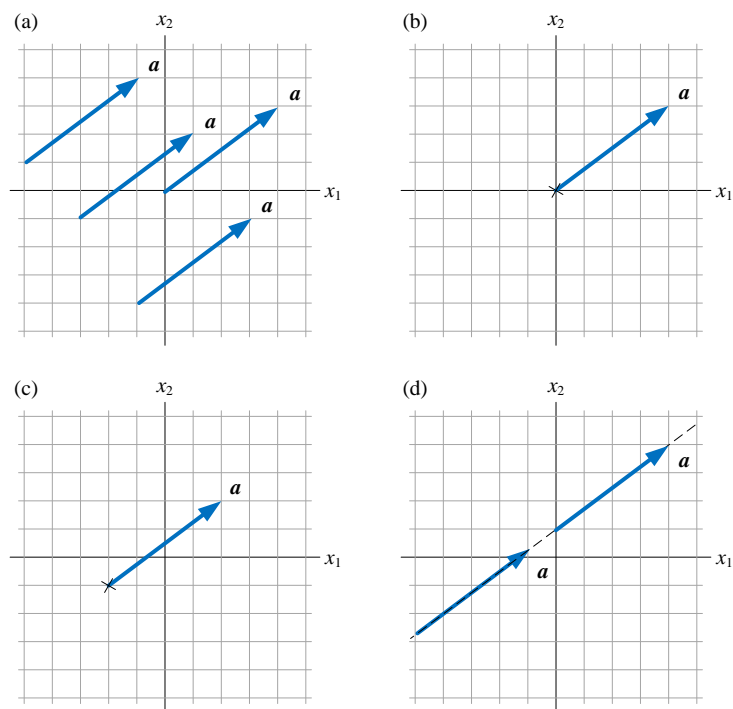


图 2. 几种向量类型

## 坐标点

从解析几何角度看，向量和坐标存在直接联系。

一般情况下直角坐标系中任意一点坐标可以通过**多元组** (tuple) 来表达。比如，图 3 (a) 所示平面直角坐标系上，A 点坐标为 (4, 3)，B 点坐标为 (-3, 4)。

图 3 (b) 所示，以原点  $O$  作为向量起点 A 为终点的向量  $\overrightarrow{OA}$  对应向量  $\mathbf{a}$ ，而  $\overrightarrow{OB}$  对应向量  $\mathbf{b}$ 。

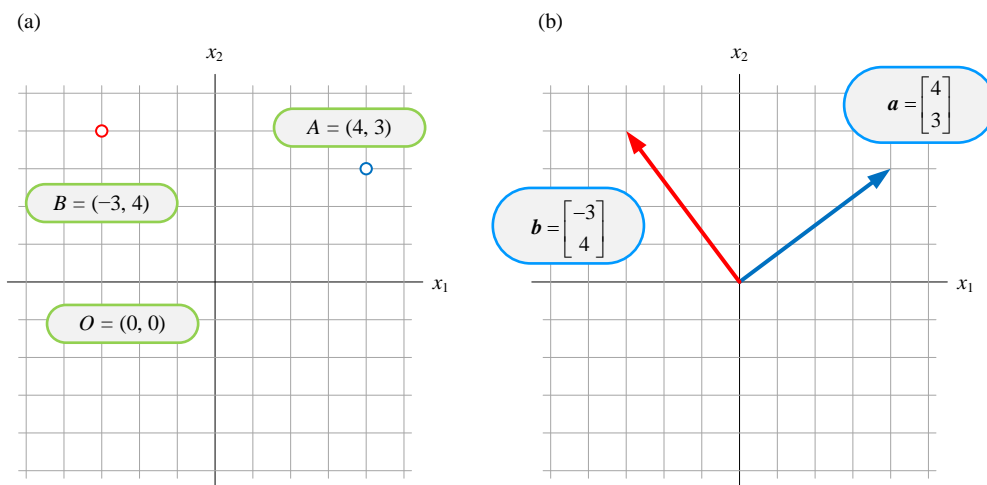


图 3. 平面坐标和向量关系

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

版权归清华大学出版社所有，请勿商用，引用请注明出处。

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

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

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

向量的元素也可以是未知量，比如  $\mathbf{x} = [x_1, x_2]^T$ 、 $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$ 。



Bk4\_Ch2\_01.py 绘制图 3 (b) 所示向量。matplotlib.pyplot.quiver() 绘制箭头图。

### 继续丰富向量几何内涵

几何上，切线指的是一条刚好触碰到曲线上某一点的直线。曲线的法线则是垂直于曲线上一点的切线的直线。将向量引入切线、法线可以得到**切向量** (tangent vector) 和**法向量** (normal vector)。图 4 所示为直线和曲线某一点处的切向量和法向量，两个向量的起点都是**切点** (point of tangency)。

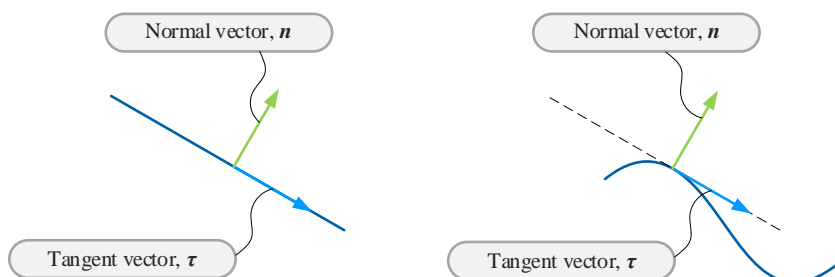


图 4. 切向量和法向量

### 梯度

自然界的风、水流、电磁场，在各自空间的每一个点上对应的物理量既有强度、也有方向。将这些既有大小又有方向的场抽象出来便得到**向量场** (vector field)。本书中，我们会使用向量场来描述函数在一系列排列整齐点的梯度向量。

图 5 (a) 所示为某个二元函数  $f(x_1, x_2)$  对应的曲面。把图 5 (a) 比作一座山峰的话，在坡面上放置一个小球，松手瞬间小球运动的方向在  $x_1x_2$  平面上的投影就是梯度下降方向，也叫做下山方向；而它的反方向叫做**梯度向量** (gradient vector) 方向，也叫上山方向。

图 5 (b) 所示为在  $x_1x_2$  平面上，二元函数  $f(x_1, x_2)$  在不同点处的平面等高线和梯度向量。坡面越陡峭，梯度向量长度越大。仔细观察，可以发现任意一点处梯度向量垂直于该点处等高线。

二元函数  $f(x_1, x_2)$  梯度向量定义如下：

$$\text{grad } f(x_1, x_2) = \nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} \quad (1)$$

在  $f(x_1, x_2)$  梯度向量中，我们看到了两个偏导数。



本书将在第 17 章回顾偏导数并讲解梯度向量。

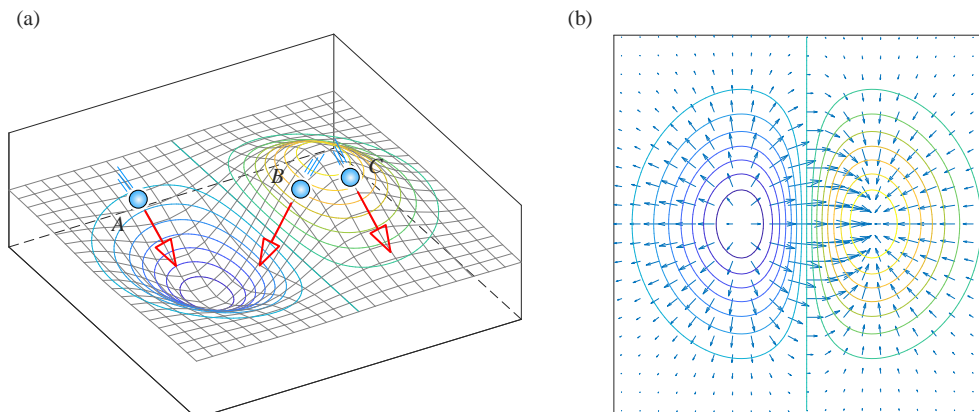
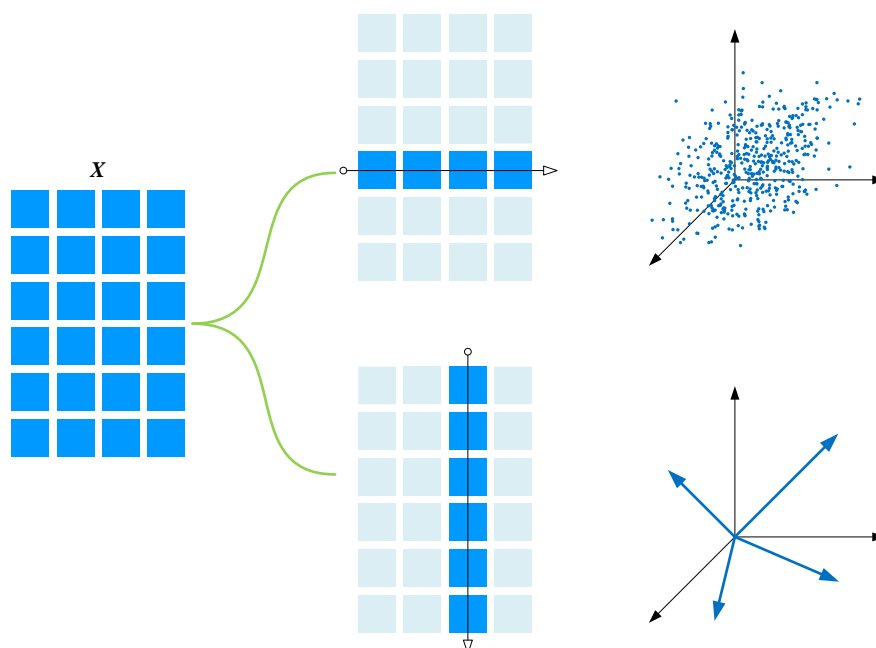


图 5. 梯度向量

## 2.2 行向量、列向量

上一章提到，向量要么一行多列、要么一列多行，因此向量可以看做是特殊的矩阵——**一维矩阵** (one-dimensional matrix)。一行多列的向量是**行向量** (row vector)，一列多行的向量叫**列向量** (column vector)。

一个矩阵可以视作由若干行向量或列向量整齐排列而成。如图 6 所示，数据矩阵  $X$  的每一行是一个行向量，代表一个样本点； $X$  的每一列为一个列向量，代表某个特征上的所有样本数据。



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

版权归清华大学出版社所有，请勿商用，引用请注明出处。

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

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

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

图 6. 观察数据矩阵的两个视角

**行向量：一行多列，一个样本数据点**

行向量将  $n$  个元素排成一行，形状为  $1 \times n$  (代表 1 行、 $n$  列)。下式行向量  $a$  为 1 行 4 列：

$$a = [1 \quad 2 \quad 3 \quad 4] \quad (2)$$

如图 7 所示，行向量**转置** (transpose) 得到列向量，反之亦然。转置运算符号为正体上标  $T$ 。

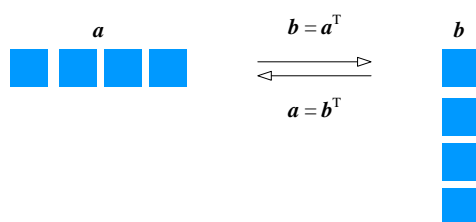


图 7. 行向量的转置是列向量

表 1 所示为利用 Numpy 构造行向量几种常见方法。可以用 `len(a)` 计算向量的长度，即向量中元素个数。

表 1. 用 Numpy 构造行向量

代码	注意事项
<code>a = numpy.array([4,3])</code>	严格地说，这种方法产生的并不是行向量；运行 <code>a.ndim</code> 发现 <code>a</code> 只有一个维度。因此，转置 <code>numpy.array([4,3]).T</code> 得到的仍然是一维数组，只不过默认展示方式为行向量。
<code>a = numpy.array([[4,3]])</code>	运行 <code>a.ndim</code> 发现 <code>a</code> 有二个维度，这个行向量转置 <code>a.T</code> 可以获得列向量。 <code>a.T</code> 求 <code>a</code> 转置，等价于 <code>a.transpose()</code> 。 请大家注意双重方括号。
<code>a = numpy.array([4,3], ndmin=2)</code>	<code>ndmin=2</code> 设定数据有两个维度，转置 <code>a.T</code> 可以获得列向量。
<code>a = numpy.r_['r', [4,3]]</code>	<code>numpy.r_[]</code> 将一系列数组合并；'r' 设定结果以行向量（默认）展示，比如 <code>numpy.r_[numpy.array([1,2]), 0, 0, numpy.array([4,5])]</code> 默认产生行向量。
<code>a = numpy.array([4,3]).reshape((1, -1))</code>	<code>reshape()</code> 按某种形式重新排列数据，-1 自动获取数组长度 <code>n</code> 。
<code>a = numpy.array([4, 3])[None, :]</code>	按照 <code>[None, :]</code> 形式广播数组， <code>None</code> 代表 <code>numpy.newaxis</code> ，增加新维度。
<code>a = numpy.array([4, 3])[numpy.newaxis, :]</code>	等同于上一例。

前文提过， $X$  的行向量序号采用“上标加括号”方式，比如  $x^{(1)}$  代表  $X$  的第一行行向量。

如图 8 所示，矩阵  $X$  可以写成一组行向量上下叠放：

$$X = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(6)} \end{bmatrix} \quad (3)$$

⚠ 再次强调，数据分析偏爱用行向量表达样本点。

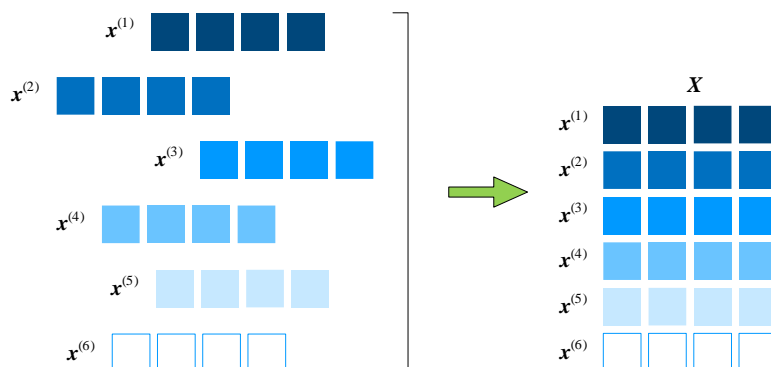


图 8. 矩阵由一系列行向量构造

### 列向量：一列多行，一个特征样本数据

列向量将  $n$  个元素排成一列，形状为  $n \times 1$  (即  $n$  行、1 列)。举个例子，下式中列向量  $\mathbf{b}$  为 4 行 1 列：

$$\mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad (4)$$

⚠ 注意，不加说明时，本书中向量一般指的是列向量。

构造  $X$  的列向量序号则采用下标表达，比如  $\mathbf{x}_1$ 。如图 9 所示，矩阵  $X$  看做是 4 个等长列向量整齐排列得到：

$$X = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3 \quad \mathbf{x}_4] \quad (5)$$

数据分析偏爱列向量表达特征，比如  $\mathbf{x}_j$  代表第  $j$  个特征上的样本数据构成的列向量。因此，列向量又常称作**特征向量** (feature vector)。 $\mathbf{x}_j$  对应概率统计的随机变量  $X_j$ ，或者代数中的变量  $x_j$ 。

⚠ 注意，此处特征向量不同于特征值分解 (eigen decomposition) 中的**特征向量** (eigenvector)。



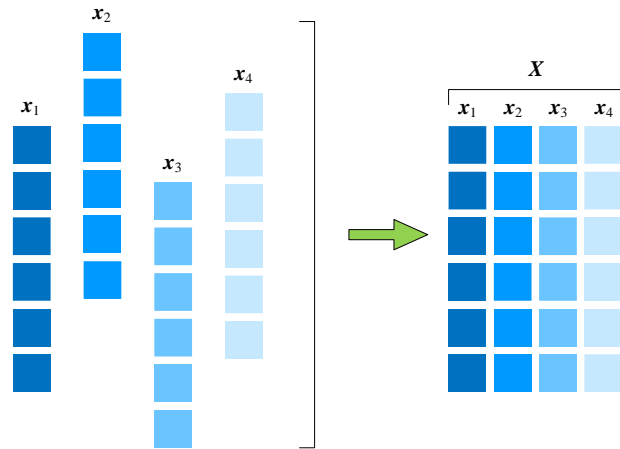


图 9. 矩阵由一排列向量构造

表 2 总结 Numpy 构造列向量几种常见方法。

表 2. 用 Numpy 构造列向量

代码	注意事项
<code>a = numpy.array([[4], [3]])</code>	运行 <code>a.ndim</code> 发现 <code>a</code> 有二个维度。 <code>numpy.array([[4], [3]]).T</code> 获得行向量。请大家注意两层方括号。
<code>a = numpy.r_['c', [4,3]]</code>	<code>numpy.r_[]</code> 将一系列的数组合并。'c' 设定结果以列向量展示
<code>a = numpy.array([4,3]).reshape((-1, 1))</code>	<code>reshape()</code> 按某种形式重新排列数据；-1 自动获取数组长度 <code>n</code>
<code>a = numpy.array([4, 3])[:, None]</code>	按照 <code>[:, None]</code> 形式广播数组； <code>None</code> 代表 <code>numpy.newaxis</code> ，增加新维度
<code>a = numpy.array([4, 3])[:, numpy.newaxis]</code>	等同于上一例

特殊列向量

**全零列向量** (zero column vector)  $\mathbf{0}$ ，是指每个元素均为 0 的列向量：

$$\mathbf{0} = [0 \ 0 \ \dots \ 0]^T$$

(6)

代码 `numpy.zeros((4, 1))` 可以生成  $4 \times 1$  全 0 列向量。多维空间中，原点也常记做零向量  $\mathbf{0}$ 。

**全 1 列向量** (all-ones column vector)  $\mathbf{1}$ ，是指每个元素均为 1 的列向量：

$$\mathbf{1} = [1 \quad 1 \quad \cdots \quad 1]^T \quad (7)$$

代码 `numpy.ones((4, 1))` 可以生成  $4 \times 1$  全 1 列向量。

➡ 全 1 列向量  $\mathbf{1}$  在矩阵乘法中有特殊的地位，本书第 5、22 章将分别从矩阵乘法和统计两个角度讲解。

## 2.3 向量长度：模，欧氏距离， $L^2$ 范数

**向量长度** (length of a vector) 又叫做**向量模** (vector norm)、**欧几里得距离** (Euclidean distance)、**欧几里得范数** (Euclidean norm) 或  **$L^2$  范数** (L2-norm)。

给定向量  $\mathbf{a}$ ：

$$\mathbf{a} = [a_1 \quad a_2 \quad \cdots \quad a_n]^T \quad (8)$$

向量  $\mathbf{a}$  的模为：

$$\|\mathbf{a}\| = \|\mathbf{a}\|_2 = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} = \left( \sum_{i=1}^n a_i^2 \right)^{\frac{1}{2}} \quad (9)$$

观察上式，容易知道向量模非负，即  $\|\mathbf{a}\| \geq 0$ 。

⚠ 注意， $\|\mathbf{a}\|_2$  下角标 2，代表  $L^2$  范数。没有特殊说明， $\|\mathbf{a}\|$  默认代表  $L^2$  范数。

➡  $L^2$  范数是  $L^p$  范数的一种，本书第 3 章将介绍其他范数。

请大家注意如下有关  $L^2$  范数性质：

$$\begin{aligned} \|\mathbf{-a}\| &= \|\mathbf{a}\| \\ \|k\mathbf{a}\| &= |k| \|\mathbf{a}\| \end{aligned} \quad (10)$$

其中， $k$  为任意实数。

### 二维向量的模

特别地，对于如下二维向量  $\mathbf{a}$ ：

$$\mathbf{a} = [a_1 \quad a_2]^T \quad (11)$$

二维向量指的是有两个元素的向量。

二维向量  $\mathbf{a}$  的  $L^2$  范数为：

$$\|\mathbf{a}\| = \sqrt{a_1^2 + a_2^2} \quad (12)$$

图 3 (b) 中向量  $\mathbf{a}$  和  $\mathbf{b}$  的模可以这样计算得到：

$$\begin{aligned} \|\mathbf{a}\| &= \sqrt{4^2 + 3^2} = \sqrt{25} = 5 \\ \|\mathbf{b}\| &= \sqrt{(-3)^2 + 4^2} = \sqrt{25} = 5 \end{aligned} \quad (13)$$

二维向量  $\mathbf{a}$  和横轴夹角可以通过反正切求解：

$$\theta_a = \arctan\left(\frac{a_2}{a_1}\right) \quad (14)$$

上述角度和直角坐标系直接关联，因此可以视作“绝对角度”。本章后续将介绍如何用向量内积求两个向量之间的“相对角度”。



Bk4\_Ch2\_02.py 计算图 3 (b) 中向量  $\mathbf{a}$  和  $\mathbf{b}$  模。函数 `numpy.linalg.norm()` 默认计算  $L^2$  范数，也可以用 `numpy.sqrt(np.sum(a**2))` 计算向量  $\mathbf{a}$  的  $L^2$  范数。

## 等距线

值得一提的是，如果起点重合，和  $\|\mathbf{a}\|$  等长的二维向量的终点位于同一个圆上，如图 10 (a) 所示。看到这里大家是否想到了本系列丛书《数学要素》第 7 章讲过的“等距线”。

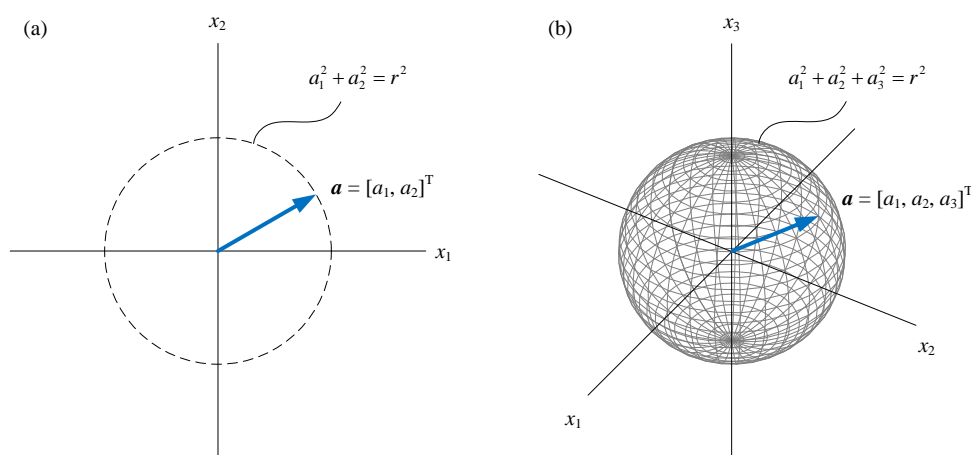


图 10. 等  $L^2$  范数向量

如图 11 所示，起点位于原点的二维向量  $\mathbf{x}$  的模  $\|\mathbf{x}\|$  取不同数值  $c$  时，我们可以得到一系列同心圆，对应的解析式如下：

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2} = c \quad (15)$$

强调一点， $\mathbf{x}$  是向量，既有大小、又有方向；而  $\|\mathbf{x}\|$  是标量，代表“距离”。 $\|\cdot\|$  这个运算符是一种“向量  $\rightarrow$  标量”的运算规则。

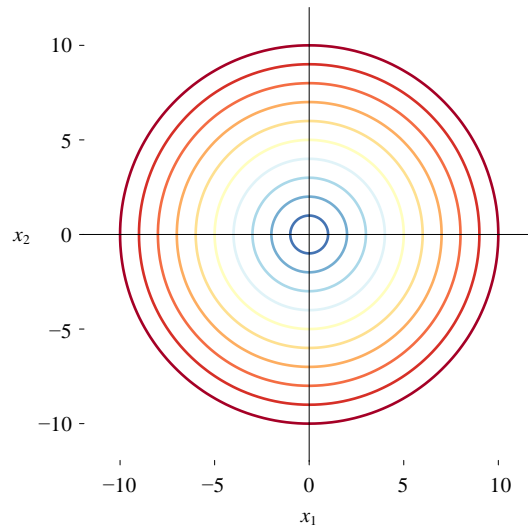
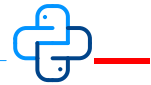


图 11. 起点为  $\mathbf{0}$ 、 $L^2$  范数相等的向量终点位于一系列同心圆上



Bk4\_Ch2\_03.py 绘制图 11。

### 三维向量的模

类似地，给定三维向量  $\mathbf{a}$ ：

$$\mathbf{a} = [a_1 \quad a_2 \quad a_3]^T \quad (16)$$

三维向量  $\mathbf{a}$  的  $L^2$  范数为：

$$\|\mathbf{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2} \quad (17)$$

如图 10 (b) 所示，起点为原点的等长三维向量终点落在同一正圆球面上。

### 单位向量

长度为 1 的向量叫做**单位向量** (unit vector)。

非  $\mathbf{0}$  向量  $\mathbf{a}$  除以自身的模得到  **$\mathbf{a}$  方向上的单位向量** (unit vector in the direction of vector  $\mathbf{a}$ ):

$$\hat{a} = \frac{a}{\|a\|} \quad (18)$$

$\hat{a}$  读作“vector a hat”。`a/numpy.linalg.norm(a)` 可以计算非  $0$  向量  $a$  方向上的单位向量。

图 12 (a) 所示平面直角坐标系，起点位于原点的单位向量  $x = [x_1, x_2]^T$  终点位于单位圆 (unit circle) 上，对应的解析式为：

$$\|x\| = \sqrt{x_1^2 + x_2^2} = 1 \Rightarrow x_1^2 + x_2^2 = 1 \quad (19)$$

这无数个单位向量  $x$  中，有两个单位向量最为特殊—— $e_1$  ( $i$ ) 和  $e_2$  ( $j$ )。如图 12 (b) 所示平面直角坐标系中， $e_1$  和  $e_2$  分别为沿着  $x_1$  (水平) 和  $x_2$  (竖直) 方向的单位向量：

$$e_1 = i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad e_2 = j = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (20)$$

显然， $e_1$  和  $e_2$  相互垂直。

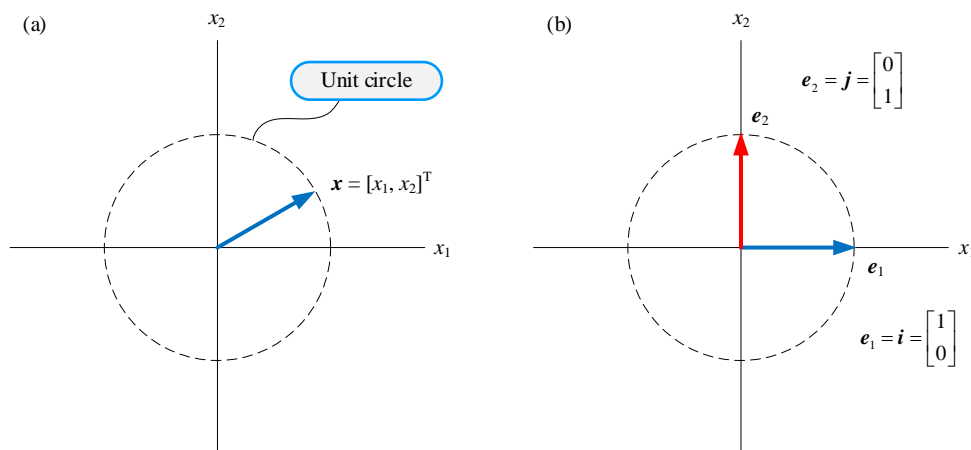


图 12. 单位向量

## 张成

图 3 (b) 给出向量  $a$  和  $b$  可以用  $e_1$  和  $e_2$  合成得到：

$$\begin{aligned} a &= 4e_1 + 3e_2 \\ b &= -3e_1 + 4e_2 \end{aligned} \quad (21)$$

(21) 用到的便是向量加减法，这是下一节要介绍的内容。

$e_1$  和  $e_2$  张成 (span) 图 3 (b) 整个平面。白话说， $e_1$  和  $e_2$  好比经纬度，可以定位  $\mathbb{R}^2$  平面任意一点。比如， $\mathbb{R}^2$  平面上的任意一点  $x$  都可以写成：

$$x = x_1 e_1 + x_2 e_2 \quad (22)$$

从集合角度来看,  $\mathbf{x} \in \mathbb{R}^2$ 。



本书第 7 章将讲解张成、向量空间等概念。

### 三维直角坐标系

三维直角坐标系中,  $\mathbf{e}_1$  ( $\mathbf{i}$ )、 $\mathbf{e}_2$  ( $\mathbf{j}$ ) 和  $\mathbf{e}_3$  ( $\mathbf{k}$ ) 代表沿着横轴、纵轴、竖轴的单位向量:

$$\mathbf{e}_1 = \mathbf{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \mathbf{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \mathbf{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (23)$$

如图 13 所示,  $\mathbf{e}_1$  ( $\mathbf{i}$ )、 $\mathbf{e}_2$  ( $\mathbf{j}$ ) 和  $\mathbf{e}_3$  ( $\mathbf{k}$ ) 两两相互垂直。

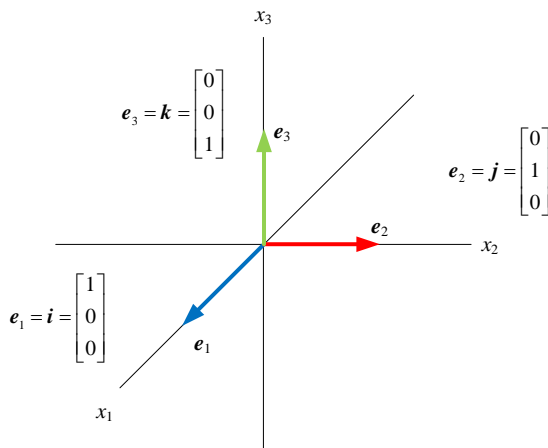


图 13. 三维空间单位向量

同理, 图 13 这个三维空间是用  $\mathbf{e}_1$ 、 $\mathbf{e}_2$ 、 $\mathbf{e}_3$  张成的。白话说,  $\mathbf{e}_1$ 、 $\mathbf{e}_2$ 、 $\mathbf{e}_3$  相当于经度、维度、海拔, 定位能力从地表扩展到整个地球空间。

$\mathbb{R}^3$  空间任意一点  $\mathbf{x}$  可以写成:

$$\mathbf{x} = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2 + x_3 \mathbf{e}_3 \quad (24)$$

此外, 大家可能已经注意到,  $\mathbf{e}_1$  可以用不同的形式表达, 比如:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (25)$$

上式中几个  $e_i$  虽然维度不同，但是本质等价，它们代表不同维度空间中的  $e_i$ 。这些  $e_i$  之间的关系是，从低维到高维或从高维到低维投影。



本书将在第 8、9、10 三章由浅入深介绍投影这一重要线性代数工具。

## 2.4 加减法：对应位置元素分别相加减

从数据角度看，两个等长列向量相加，结果为对应位置元素分别相加，得到长度相同的列向量，比如下例：

$$a + b = \begin{bmatrix} -2 \\ 5 \end{bmatrix} + \begin{bmatrix} 5 \\ -1 \end{bmatrix} = \begin{bmatrix} -2+5 \\ 5-1 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \quad (26)$$

两个等长列向量相减，则是对应元素分别相减，得到等长列向量，比如：

$$a - b = \begin{bmatrix} -2 \\ 5 \end{bmatrix} - \begin{bmatrix} 5 \\ -1 \end{bmatrix} = \begin{bmatrix} -2-5 \\ 5-(-1) \end{bmatrix} = \begin{bmatrix} -7 \\ 6 \end{bmatrix} \quad (27)$$

以上法则也适用于行向量。

### 几何视角

从几何角度看，**向量加法** (vector addition) 结果可以用**平行四边形法则** (parallelogram method) 或**三角形法则** (triangle method) 获得，具体如图 14 所示。

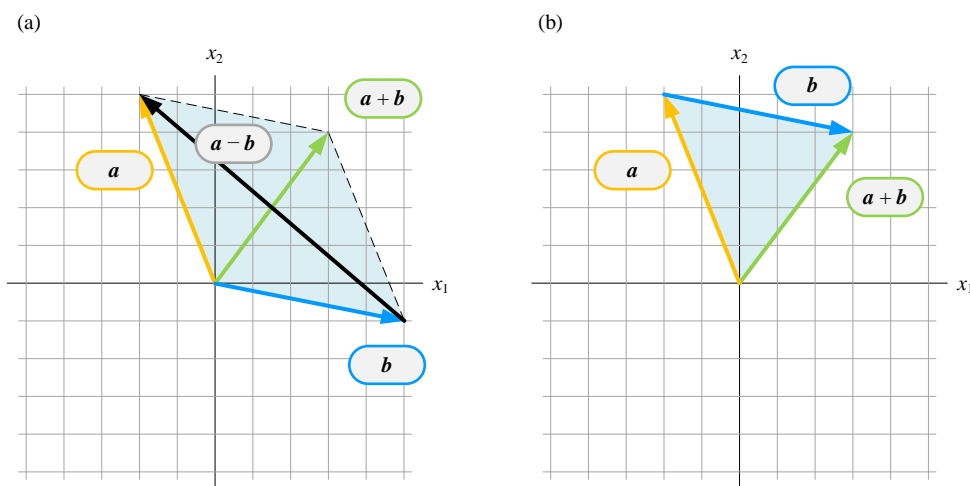


图 14. 几何角度看向量加法

**向量减法** (vector subtraction) 可以写成向量加法。比如，向量  $a$  减去向量  $b$ ，可以将向量  $b$  换向得到  $-b$ ；然后再计算向量  $a$  与向量  $-b$  之和，即：

$$a - b = a + (-b) = \begin{bmatrix} -2 \\ 5 \end{bmatrix} + \begin{bmatrix} -5 \\ 1 \end{bmatrix} = \begin{bmatrix} -7 \\ 6 \end{bmatrix} \quad (28)$$

$-b$

▲ 注意，向量  $a$  减去向量  $b$ ，结果  $a - b$  对应向量箭头指向  $a$  终点；相反，向量  $b$  减去向量  $a$  得到  $b - a$  指向  $b$  终点。

两个向量相同，当且仅当两者大小方向均相同。如果两个向量的模（长度）相同但是方向相反，两者互为反向量。若两个向量方向相同或相反，则称向量平行。

请大家注意以下向量加减法性质：

$$\begin{aligned} a + b &= b + a \\ (a + b) + c &= a + (b + c) \\ a + (-a) &= 0 \end{aligned} \quad (29)$$

## 两点距离

向量差  $a - b$  的模 ( $L^2$  范数)  $\|a - b\|$  就是图 14 (a) 中  $a$  和  $b$  两点的欧氏距离，即：

$$\|a - b\| = \|a - b\|_2 = \sqrt{(-7)^2 + 8^2} = \sqrt{49 + 64} = \sqrt{113} \quad (30)$$

$a$  和  $b$  两点的欧氏距离的平方为：

$$\|a - b\|^2 = \|a - b\|_2^2 = (-7)^2 + 8^2 = 113 \quad (31)$$



Bk4\_Ch2\_04.py 计算本节向量加减法示例。

## 2.5 标量乘法：向量缩放

**向量标量乘法** (scalar multiplication of vectors) 指的是标量和向量每个元素分别相乘，结果仍为向量。从几何角度来看，标量乘法将原向量按标量比例缩放，结果中向量方向同向或反向，如图 15 所示。



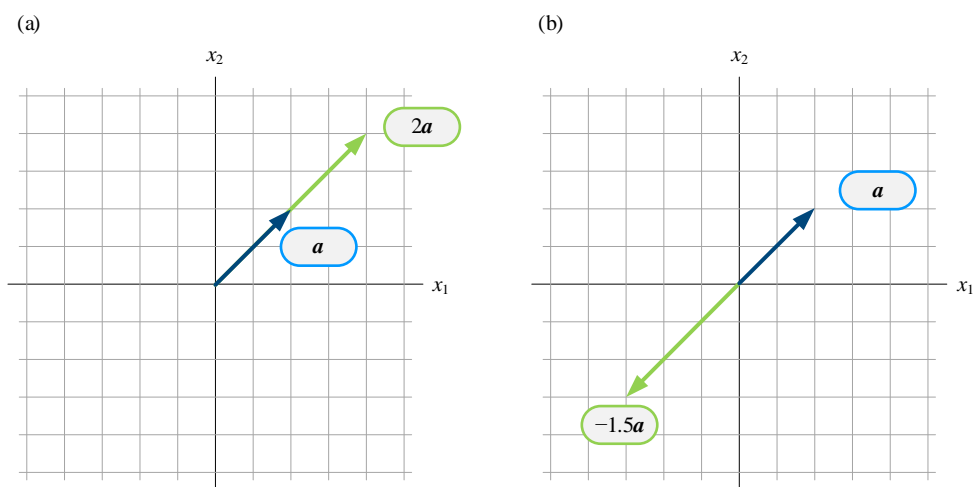


图 15. 向量标量乘法



Bk4\_Ch2\_05.py 完成图 15 中运算。

请大家注意以下向量标量乘法性质：

$$\begin{aligned}
 (t+k)a &= ta + ka \\
 t(a+b) &= ta + tb \\
 t(ka) &= tka \\
 1a &= a \\
 -1a &= -a \\
 0a &= \mathbf{0}
 \end{aligned} \tag{32}$$

其中， $t$  和  $k$  为标量。请大家特别注意， $0$  乘向量  $a$  结果不是  $0$ ，而是零向量  $\mathbf{0}$ ，这个零向量的形状取决于向量  $a$ 。

## 2.6 向量内积：结果为标量

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

给定如下  $a$  和  $b$  两个等长列向量：

$$\begin{aligned} \mathbf{a} &= [a_1 \ a_2 \ \cdots \ a_n]^T \\ \mathbf{b} &= [b_1 \ b_2 \ \cdots \ b_n]^T \end{aligned} \quad (33)$$

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

$$\mathbf{a} \cdot \mathbf{b} = \langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n \quad (34)$$

(34) 也适用于两个等长行向量计算内积。注意，向量内积也是一种“向量  $\rightarrow$  标量”的运算规则。

图 16 所示的两个列向量  $\mathbf{a}$  和  $\mathbf{b}$  的内积为：

$$\mathbf{a} \cdot \mathbf{b} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ -2 \end{bmatrix} = 4 \times 5 + 3 \times (-2) = 14 \quad (35)$$

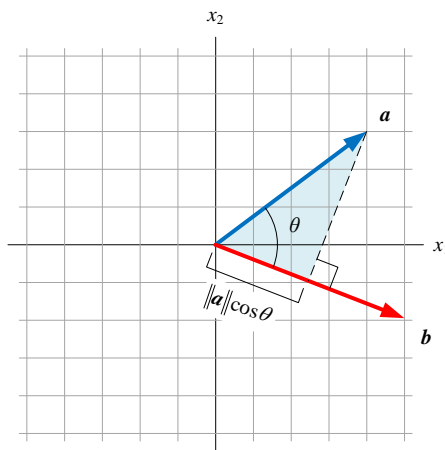
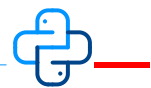


图 16.  $\mathbf{a}$  和  $\mathbf{b}$  两个平面向量



Bk4\_Ch2\_06.py 计算上述向量内积。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\_07.py 给出例子。

`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\_08.py 给出示例。

常用的向量内积性质如下：

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= \mathbf{b} \cdot \mathbf{a} \\ \mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) &= \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \cdot \mathbf{c} \\ (k\mathbf{a}) \cdot (t\mathbf{b}) &= kt(\mathbf{a} \cdot \mathbf{b}) \end{aligned} \quad (36)$$

请读者格外注意以下几个向量内积运算和  $\Sigma$  求和运算的关系：

$$\begin{aligned} \mathbf{1} \cdot \mathbf{x} &= x_1 + x_2 + \cdots x_n = \sum_{i=1}^n x_i \\ \mathbf{x} \cdot \mathbf{x} &= x_1^2 + x_2^2 + \cdots x_n^2 = \sum_{i=1}^n x_i^2 \\ \mathbf{x} \cdot \mathbf{y} &= x_1 y_1 + x_2 y_2 + \cdots x_n y_n = \sum_{i=1}^n x_i y_i \end{aligned} \quad (37)$$

其中，

$$\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T, \quad \mathbf{1} = [1 \ 1 \ \cdots \ 1]^T, \quad \mathbf{y} = [y_1 \ y_2 \ \cdots \ y_n]^T \quad (38)$$

本书第 5 章还会从矩阵乘法角度介绍更多求和运算。

## 几何视角

如图 16 所示，从几何角度看，向量内积相当于两个向量的模 ( $L^2$  范数) 与它们之间夹角余弦值三者之积：

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta \quad (39)$$

注意，上式中  $\theta$  代表向量  $\mathbf{a}$  和  $\mathbf{b}$  的“相对夹角”。



此外，向量内积还可以从投影 (projection) 角度来解释，这是本书第 9 章要介绍的内容。

$\mathbf{a}$  的  $L^2$  范数也可以通过向量内积求得：

$$\|\mathbf{a}\|_2 = \|\mathbf{a}\| = \sqrt{\mathbf{a} \cdot \mathbf{a}} = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle} \quad (40)$$

(40) 左右等式平方得到：

$$\|\mathbf{a}\|_2^2 = \|\mathbf{a}\|^2 = \mathbf{a} \cdot \mathbf{a} = \langle \mathbf{a}, \mathbf{a} \rangle \quad (41)$$

上式相当于“距离的平方”。

## 柯西-施瓦茨不等式

观察 (39)，我们可以发现  $\cos \theta$  的取值范围为  $[-1, 1]$ ，因此  $\mathbf{a}$  和  $\mathbf{b}$  内积取值范围如下：

$$-\|a\|\|b\| \leq a \cdot b \leq \|a\|\|b\| \quad (42)$$

图 17 所示为 7 个不同向量夹角状态。

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

平面上, 非零向量  $a$  和  $b$  垂直,  $a$  和  $b$  夹角为  $90^\circ$ , 两者向量内积为 0:

$$a \cdot b = \|a\|\|b\|\cos 90^\circ = 0 \quad (43)$$

多维向量  $a$  和  $b$  向量内积为 0, 我们称  $a$  和  $b$  正交 (orthogonal)。本书上一章提到, 正交是线性代数的概念, 是垂直的推广。

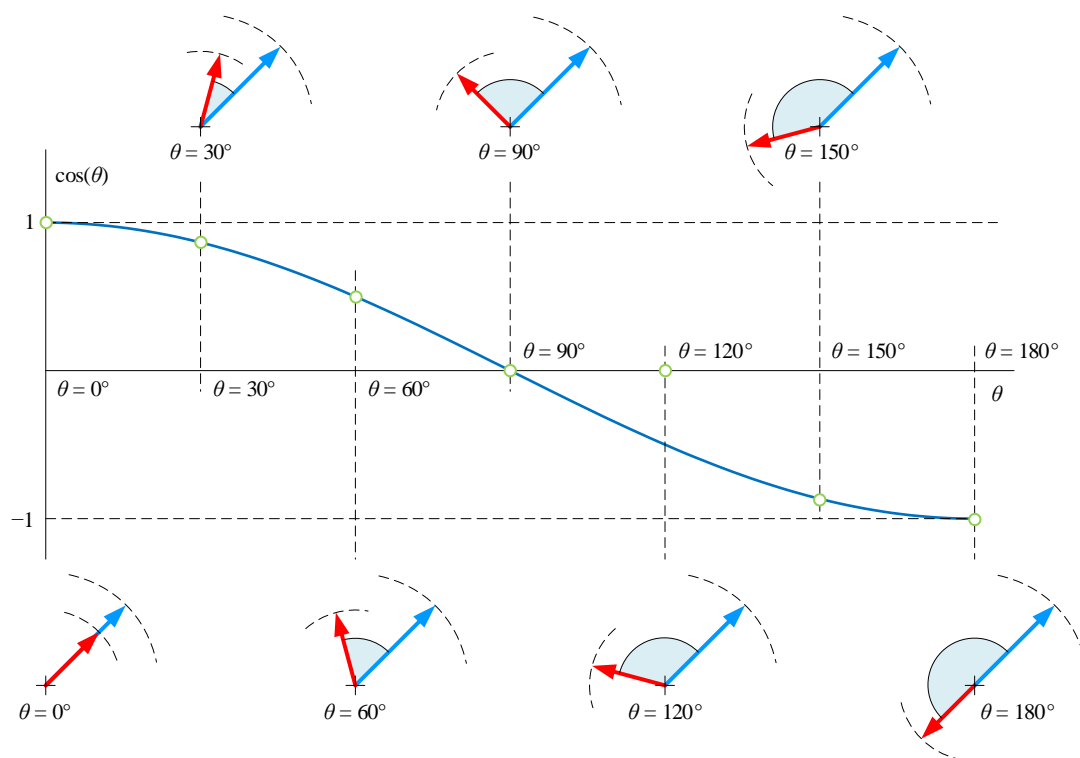


图 17. 向量夹角

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):

$$(a \cdot b)^2 \leq \|a\|^2 \|b\|^2 \quad (44)$$

即:

$$|\mathbf{a} \cdot \mathbf{b}| \leq \|\mathbf{a}\| \|\mathbf{b}\| \quad (45)$$

$|\mathbf{a} \cdot \mathbf{b}|$  代表  $\mathbf{a}$  和  $\mathbf{b}$  向量内积绝对值。

用尖括号来表达向量内积，(44) 可以写成：

$$\langle \mathbf{a}, \mathbf{b} \rangle^2 \leq \langle \mathbf{a}, \mathbf{a} \rangle \langle \mathbf{b}, \mathbf{b} \rangle \quad (46)$$

即：

$$|\langle \mathbf{a}, \mathbf{b} \rangle| \leq \|\mathbf{a}\| \|\mathbf{b}\| \quad (47)$$

在  $\mathbb{R}^n$  空间中，上述不等式等价于：

$$\left( \sum_{i=1}^n a_i b_i \right)^2 \leq \left( \sum_{i=1}^n a_i^2 \right) \left( \sum_{i=1}^n b_i^2 \right) \quad (48)$$

### 余弦定理

回忆丛书第一本书讲解的余弦定理 (law of cosines)：

$$c^2 = a^2 + b^2 - 2ab \cos \theta \quad (49)$$

其中， $a$ 、 $b$  和  $c$  为图 18 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 (39)。

如图 18 所示，将三角形三个边视作向量，将三个向量长度代入 (49)，可以得到：

$$\|\mathbf{c}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\|\mathbf{a}\| \|\mathbf{b}\| \cos \theta \quad (50)$$

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

$$\mathbf{c} = \mathbf{a} - \mathbf{b} \quad (51)$$

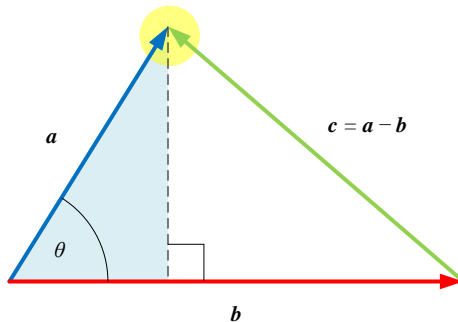


图 18. 余弦定理

(51) 等式左右分别和自身计算向量内积，得到如下等式：

$$\mathbf{c} \cdot \mathbf{c} = (\mathbf{a} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b}) \quad (52)$$

整理上式得到：

$$\begin{aligned} \mathbf{c} \cdot \mathbf{c} &= (\mathbf{a} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b}) = \mathbf{a} \cdot \mathbf{a} + \mathbf{b} \cdot \mathbf{b} - \mathbf{a} \cdot \mathbf{b} - \mathbf{b} \cdot \mathbf{a} \\ &= \mathbf{a} \cdot \mathbf{a} + \mathbf{b} \cdot \mathbf{b} - 2\mathbf{a} \cdot \mathbf{b} \end{aligned} \quad (53)$$

利用 (41)，(53) 可以写作：

$$\|\mathbf{c}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a} \cdot \mathbf{b} \quad (54)$$

比较 (50) 和 (54)，可以得到：

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta \quad (55)$$



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：

$$A(a_1, a_2, \dots, a_n), B(b_1, b_2, \dots, b_n) \quad (56)$$

计算  $A$  和  $B$  两点的距离  $AB$ ：

$$\begin{aligned} AB &= \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2} \\ &= \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \end{aligned} \quad (57)$$

用起点位于原点的向量  $\mathbf{a}$  和  $\mathbf{b}$  分别代表  $A$  和  $B$  点， $AB$  距离就是  $\mathbf{a} - \mathbf{b}$  的  $L_2$  范数，也就是欧几里得距离：

$$AB = \|\mathbf{a} - \mathbf{b}\| = \sqrt{(\mathbf{a} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b})} = \sqrt{\mathbf{a} \cdot \mathbf{a} + \mathbf{b} \cdot \mathbf{b} - 2\mathbf{a} \cdot \mathbf{b}} \quad (58)$$

回忆《数学要素》一册中介绍的样本方差公式，具体如下：

$$\text{var}(X) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2 \quad (59)$$

注意，对于总体方差，上式分母中  $n-1$  改为  $n$ 。上式还默认  $X$  为有  $n$  个相等概率值的平均分布。

令  $\mathbf{x}$  为，

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T \quad (60)$$

(59) 可以写成：

$$\text{var}(X) = \frac{(\mathbf{x} - \boldsymbol{\mu}) \cdot (\mathbf{x} - \boldsymbol{\mu})}{n-1} \quad (61)$$

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

回忆总样本协方差公式：

$$\text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_X)(y_i - \mu_Y) \quad (62)$$

同样，对于总体协方差，上式分母中  $n-1$  改为  $n$ 。

同样利用向量内积运算法则，上式可以写成：

$$\text{cov}(X, Y) = \frac{(\mathbf{x} - \boldsymbol{\mu}_X) \cdot (\mathbf{y} - \boldsymbol{\mu}_Y)}{n-1} \quad (63)$$

本书第 22 章将从线性代数角度再和大家探讨概率统计相关内容。

## 2.7 向量夹角：反余弦

根据 (39)，可以得到向量  $\mathbf{a}$  和  $\mathbf{b}$  夹角余弦值：

$$\cos \theta = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (64)$$

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

$$\theta = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}\right) \quad (65)$$

$\arccos()$  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。再次强调，(65) 代表向量  $\mathbf{a}$  和  $\mathbf{b}$  之间的“相对角度”。而  $\mathbf{a}$  和  $\mathbf{e}_1$ 、 $\mathbf{b}$  和  $\mathbf{e}_1$  的夹角可以视作“绝对夹角”。

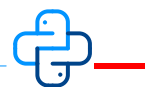


图 16 中向量  $\mathbf{a}$  和  $\mathbf{b}$  夹角弧度值和角度值可以通过 `Bk4_Ch2_09.py` 计算。

### 极坐标

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

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (66)$$

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

$$\begin{cases} \cos \theta_a = \frac{a_1}{\|\mathbf{a}\|}, & \sin \theta_a = \frac{a_2}{\|\mathbf{a}\|} \\ \cos \theta_b = \frac{b_1}{\|\mathbf{b}\|}, & \sin \theta_b = \frac{b_2}{\|\mathbf{b}\|} \end{cases} \quad (67)$$

$\theta_a$  和  $\theta_b$  就相当于绝对角度。

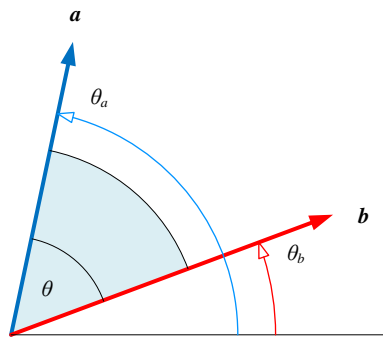


图 19. 极坐标中解释向量夹角

根据角的余弦和差恒等式， $\cos(\theta)$  可以由  $\theta_a$  和  $\theta_b$  正、余弦构造：

$$\begin{aligned} \cos(\theta) &= \cos(\theta_b - \theta_a) = \cos(\theta_b)\cos(\theta_a) + \sin(\theta_b)\sin(\theta_a) \\ &= \frac{a_1}{\|\mathbf{a}\|} \frac{b_1}{\|\mathbf{b}\|} + \frac{a_2}{\|\mathbf{a}\|} \frac{b_2}{\|\mathbf{b}\|} = \frac{a_1 b_1 + a_2 b_2}{\|\mathbf{a}\| \|\mathbf{b}\|} \end{aligned} \quad (68)$$

将 (67) 代入 (68) 得到：

$$\cos \theta = \frac{a_1}{\|\mathbf{a}\|} \frac{b_1}{\|\mathbf{b}\|} + \frac{a_2}{\|\mathbf{a}\|} \frac{b_2}{\|\mathbf{b}\|} = \frac{\overbrace{a_1 b_1 + a_2 b_2}^{a \cdot b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (69)$$

相信大家已经在上式分母中看到向量内积。

## 单位向量

本章前文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。

给定两个非  $\mathbf{0}$  向量  $\mathbf{a}$  和  $\mathbf{b}$ ，首先计算它们各自方向上的单位向量：



$$\hat{a} = \frac{a}{\|a\|}, \quad \hat{b} = \frac{b}{\|b\|} \quad (70)$$

两个单位向量的内积就是夹角的余弦值：

$$\hat{a} \cdot \hat{b} = \frac{a}{\|a\|} \cdot \frac{b}{\|b\|} = \cos \theta \quad (71)$$

## 正交单位向量

本章前文介绍的平面直角坐标系中  $e_1$  和  $e_2$  分别代表为沿着横轴、纵轴的单位向量。它们相互正交，也就是向量内积为 0：

$$e_1 \cdot e_2 = \langle e_1, e_2 \rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \quad (72)$$

在一个平面上，单位向量  $e_1$ 、 $e_2$  相互垂直，它俩“张起”的方方正正的网格，就是标准直角坐标系，具体如图 20 (a) 所示。

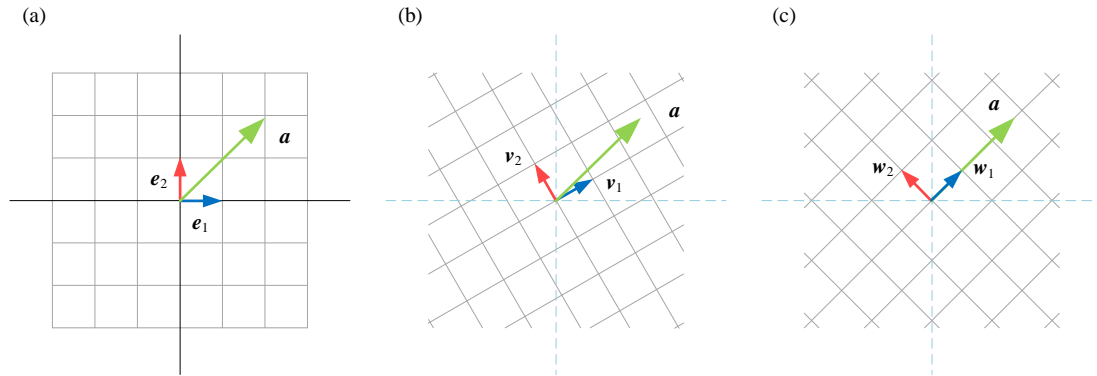


图 20. 向量  $a$  在三个不同的正交直角坐标系中位置

而平面上，成对正交单位向量有无数组，比如图 21 所示平面两组正交单位向量：

$$v_1 \cdot v_2 = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} = 0, \quad w_1 \cdot w_2 = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} = 0 \quad (73)$$

$v_1$ 、 $v_2$  构造如图 20 (b) 所示直角坐标系。类似地， $w_1$ 、 $w_2$  也可以构造如图 20 (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。

比较图 20 三幅子图，同一个向量  $a$  在三个直角坐标系中有不同的坐标值。向量  $a$  在图 20 (a) 所示直角坐标系的坐标值很容易确定为 (2, 2)。目前我们还没有掌握足够的数学工具来计算向量  $a$  在图 20 (b) 和 (c) 两个直角坐标系中的坐标值。这个问题要留到本书第 7 章来解决。

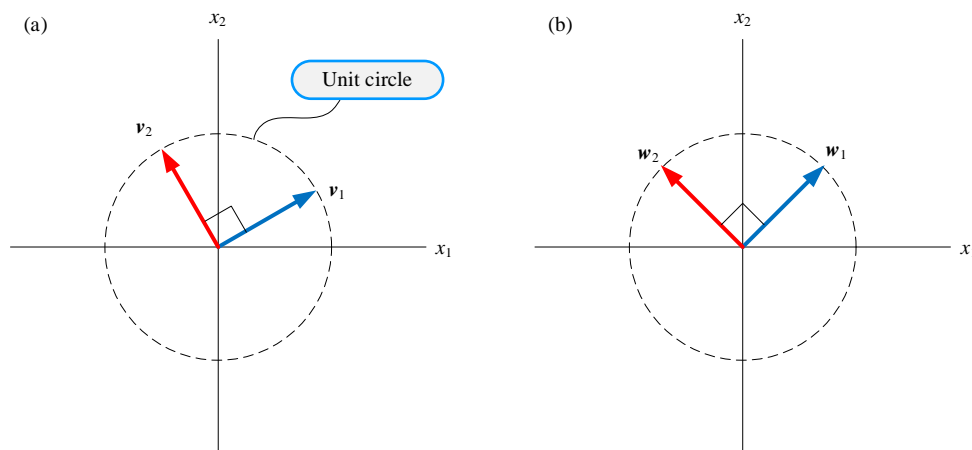


图 21. 两组正交单位向量

➔  $[e_1, e_2]$ 、 $[v_1, v_2]$ 、 $[w_1, w_2]$  都叫做  $\mathbb{R}^2$  的规范正交基 (orthonormal basis)，而  $[e_1, e_2]$  有自己特别的名字——标准正交基 (standard basis)。而且大家很快就会发现  $[e_1, e_2]$  旋转一定角度可以得到  $[v_1, v_2]$ 、 $[w_1, w_2]$ 。本书第 7 章将深入介绍相关概念。

## 2.8 余弦相似度和余弦距离

### 余弦相似度

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

$$k(x, q) = \frac{x \cdot q}{\|x\| \|q\|} = \frac{x^T q}{\|x\| \|q\|} \quad (74)$$

上一节我们介绍过，如果两个向量方向相同，则夹角  $\theta$  余弦值  $\cos(\theta) = 1$ 。若两个向量方向完全相反，夹角  $\theta$  余弦值  $\cos(\theta) = -1$ 。

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

### 余弦距离

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  $d(\mathbf{x}, \mathbf{q})$  来表达  $\mathbf{x}$  和  $\mathbf{q}$  两个列向量的余弦距离，具体定义如下：

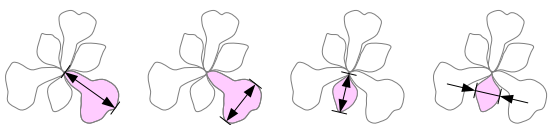
$$d(\mathbf{x}, \mathbf{q}) = 1 - k(\mathbf{x}, \mathbf{q}) = 1 - \frac{\mathbf{x} \cdot \mathbf{q}}{\|\mathbf{x}\| \|\mathbf{q}\|} \quad (75)$$

本章前文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。 $L^2$  范数的取值范围为  $[0, +\infty)$ ，而余弦距离的取值范围为  $[0, 2]$ 。

本书下一章，以及《概率统计》、《机器学习》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 22 给出鸢尾花四个样本数据。 $\mathbf{x}^{(1)}$  和  $\mathbf{x}^{(2)}$  两个样本对应的鸢尾花都是 setosa 这一亚属。 $\mathbf{x}^{(51)}$  样本对应的鸢尾花为 versicolor 这一亚属； $\mathbf{x}^{(101)}$  样本对应的鸢尾花为 virginica 这一亚属。



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

图 22. 鸢尾花的四个样本数据

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

$$\begin{aligned} d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) &= 1 - k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \\ &= 1 - \frac{5.1 \times 4.9 + 3.5 \times 3 + 1.4 \times 1.4 + 0.2 \times 0.2}{\sqrt{5.1^2 + 3.5^2 + 1.4^2 + 0.2^2} \times \sqrt{4.9^2 + 3^2 + 1.4^2 + 0.2^2}} \\ &= 1 - \frac{37.49}{6.34507 \times 5.9169} \\ &= 1 - 0.99857 = 0.00142 \end{aligned} \quad (76)$$

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

$$\begin{aligned} d(\mathbf{x}^{(1)}, \mathbf{x}^{(51)}) &= 0.07161 \\ d(\mathbf{x}^{(1)}, \mathbf{x}^{(101)}) &= 0.13991 \end{aligned} \quad (77)$$

可以发现,  $\mathbf{x}^{(1)}$  和  $\mathbf{x}^{(2)}$  两朵同属于 setosa 亚属的鸢尾花, 余弦距离较近, 也就是较为相似。

$\mathbf{x}^{(1)}$  和  $\mathbf{x}^{(101)}$  分别属于 setosa 和 virginica 亚属, 余弦距离较远, 也就是不相似。

大家思考以下几个问题, 鸢尾花数据有 150 个数据点, 任意两个数据点可以计算得到一个余弦相似度。因此成对余弦相似度有 11175 个, 大家想想该怎么便捷计算、存储这些数据呢?

此外, 大家可以试着先给数据去均值, 如图 23 所示, 将向量起点移动到原点, 然后再计算余弦距离, 并比较结果差异。和之前相比, 去均值是否有利于区分不同类别鸢尾花?

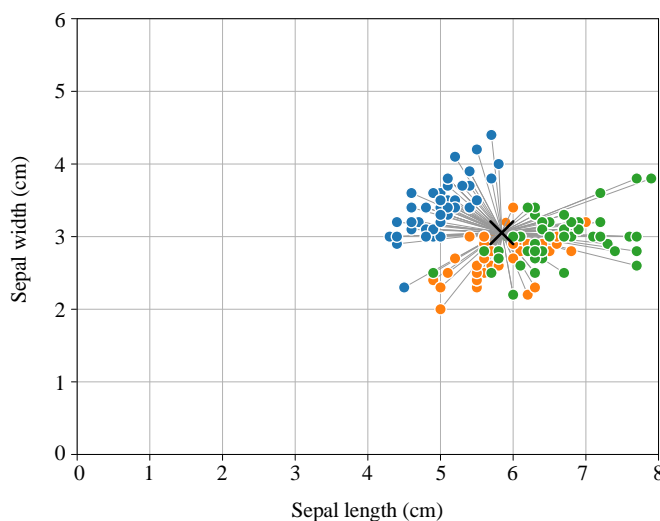


图 23. 向量起点移到鸢尾花数据质心



Bk4\_Ch2\_10.py 可以完成上述计算。感兴趣的读者可以修改代码计算  $\mathbf{x}^{(51)}$  和  $\mathbf{x}^{(101)}$  的余弦距离, 并结合样本标签分析结果。

## 2.9 向量积：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。也就是说, 向量积一种“向量  $\rightarrow$  向量”的运算规则。

$\mathbf{a}$  和  $\mathbf{b}$  向量积, 记做  $\mathbf{a} \times \mathbf{b}$ 。 $\mathbf{a} \times \mathbf{b}$  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 24 所示,  $\mathbf{a} \times \mathbf{b}$  方向分别垂直于向量  $\mathbf{a}$  和  $\mathbf{b}$ , 即  $\mathbf{a} \times \mathbf{b}$  垂直于向量  $\mathbf{a}$  和  $\mathbf{b}$  构成平面。

向量  $a$  和  $b$  以及  $a \times b$  三者关系可以用右手法则判断，如图 25 所示。图 25 这幅图中，我们可以看到  $a \times b$  和  $b \times a$  方向相反。

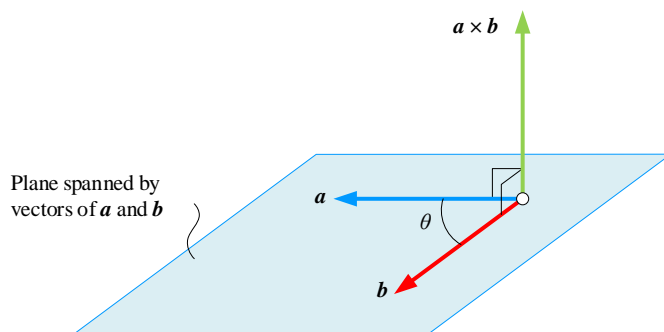


图 24.  $a \times b$  垂直于向量  $a$  和  $b$  构成平面

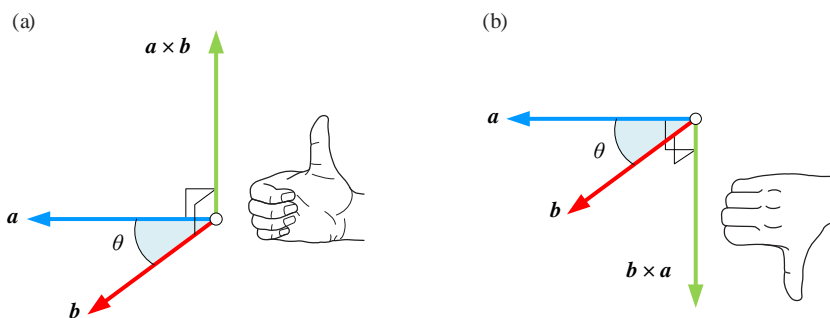


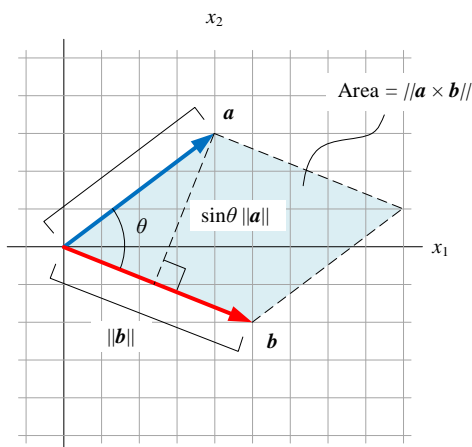
图 25. 向量叉乘右手定则

## 大小

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

$$\|a \times b\| = \|a\| \|b\| \sin(\theta) \quad (78)$$

其中  $\theta$  为向量  $a$  和  $b$  夹角。如图 26 所示，从几何角度，向量积的模  $\|a \times b\|$  相当于图中平行四边形的面积。

图 26.  $a \times b$  向量积的几何含义

### 正交向量之间的叉乘

如图 27 (a) 所示，空间直角坐标系中三个正交向量  $e_1(i)$  (横轴正方向)、 $e_2(j)$  (纵轴正方向) 和  $e_3(k)$  (竖轴正方向) 向量叉乘关系存在如下关系：

$$i \times j = k, \quad j \times k = i, \quad k \times i = j \quad (79)$$

图 27 (b) 展示以上三个等式中  $i$ 、 $j$  和  $k$  前后顺序关系。若调换 (79) 叉乘元素顺序，结果反向，对应以下三个运算式：

$$j \times i = -k, \quad k \times j = -i, \quad i \times k = -j \quad (80)$$

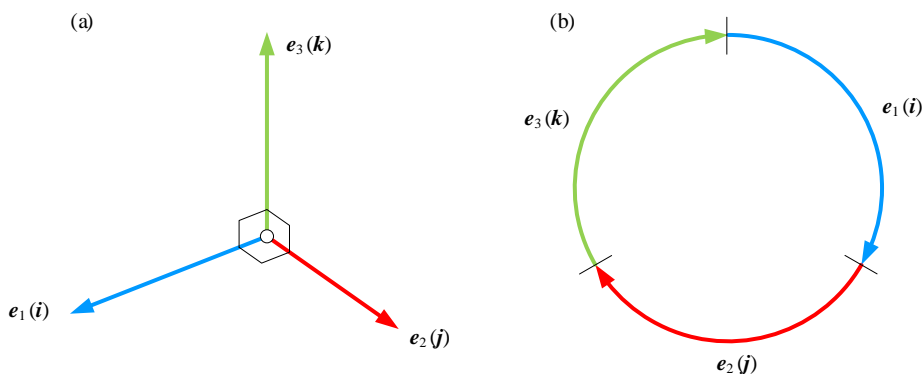


图 27. 三维空间正交单位向量基底之间关系

特别的，向量与自身叉乘等于  $0$  向量，比如：

$$i \times i = 0, \quad j \times j = 0, \quad k \times k = 0 \quad (81)$$

下列为叉乘运算常见性质：

$$\begin{aligned}
\mathbf{a} \times \mathbf{a} &= \mathbf{0} \\
\mathbf{a} \times (\mathbf{b} + \mathbf{c}) &= \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c} \\
(\mathbf{a} + \mathbf{b}) \times \mathbf{c} &= \mathbf{a} \times \mathbf{c} + \mathbf{b} \times \mathbf{c} \\
\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) &\neq (\mathbf{a} \times \mathbf{b}) \times \mathbf{c} \\
k(\mathbf{a} \times \mathbf{b}) &= k\mathbf{a} \times \mathbf{b} = \mathbf{a} \times (k\mathbf{b}) \\
\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) &= (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}
\end{aligned} \tag{82}$$

### 任意两个向量的叉乘

在三维直角坐标系中，用  $\mathbf{i}$ 、 $\mathbf{j}$  和  $\mathbf{k}$  表达向量  $\mathbf{a}$  和  $\mathbf{b}$ ：

$$\begin{aligned}
\mathbf{a} &= a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} \\
\mathbf{b} &= b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k}
\end{aligned} \tag{83}$$

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

$$\begin{aligned}
\mathbf{a} \times \mathbf{b} &= (a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}) \times (b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k}) \\
&= a_1b_1(\mathbf{i} \times \mathbf{i}) + a_1b_2(\mathbf{i} \times \mathbf{j}) + a_1b_3(\mathbf{i} \times \mathbf{k}) \\
&\quad + a_2b_1(\mathbf{j} \times \mathbf{i}) + a_2b_2(\mathbf{j} \times \mathbf{j}) + a_2b_3(\mathbf{j} \times \mathbf{k}) \\
&\quad + a_3b_1(\mathbf{k} \times \mathbf{i}) + a_3b_2(\mathbf{k} \times \mathbf{j}) + a_3b_3(\mathbf{k} \times \mathbf{k}) \\
&= (a_2b_3 - a_3b_2)\mathbf{i} + (a_3b_1 - a_1b_3)\mathbf{j} + (a_1b_2 - a_2b_1)\mathbf{k}
\end{aligned} \tag{84}$$



$\mathbf{a}$  和  $\mathbf{b}$  叉乘还可以通过行列式求解，我们将在本书第 4 章讲解。

### 举个例子

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

$$\mathbf{a} = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} \tag{85}$$

$\mathbf{a} \times \mathbf{b}$  结果如下：

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix} \tag{86}$$



Bk4\_Ch2\_11.py 计算得到 (86)。其中，`numpy.cross()` 函数可以用来计算列向量和行向量的向量积。

## 2.10 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)。逐项积指的是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 28 给出的是从数据角度看向量逐项积运算。

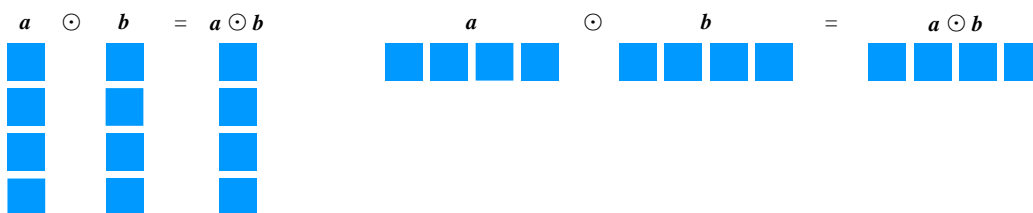


图 28. 向量逐项积运算

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

$$\begin{aligned}\mathbf{a} &= [a_1 \quad a_2 \quad \cdots \quad a_n]^T \\ \mathbf{b} &= [b_1 \quad b_2 \quad \cdots \quad b_n]^T\end{aligned}\tag{87}$$

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

$$\mathbf{a} \odot \mathbf{b} = [a_1 b_1 \quad a_2 b_2 \quad \cdots \quad a_n b_n]^T\tag{88}$$

逐项积是一种“向量  $\rightarrow$  向量”的运算规则。



Bk4\_Ch2\_12.py 计算行向量逐项积。

## 2.11 向量张量积：张起网格面

**张量积** (tensor product) 又叫**克罗内克积** (Kronecker product)，两个列向量  $\mathbf{a}$  和  $\mathbf{b}$  张量积  $\mathbf{a} \otimes \mathbf{b}$  定义如下：



$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}_{n \times 1} \otimes \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m \times 1} = \mathbf{ab}^T = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}^T = \begin{bmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_m \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_m \\ \vdots & \vdots & \ddots & \vdots \\ a_n b_1 & a_n b_2 & \cdots & a_n b_m \end{bmatrix}_{n \times m} \quad (89)$$

向量张量积是一种“向量 → 矩阵”的运算规则。

⚠ 注意，(89) 上式中  $\mathbf{ab}^T$  为向量  $\mathbf{a}$  和  $\mathbf{b}^T$  的矩阵乘法。本书第 4、5、6 三章要从不同角度讲解矩阵乘法。

向量  $\mathbf{a}$  和其自身张量积  $\mathbf{a} \otimes \mathbf{a}$  结果为方阵：

$$\mathbf{a} \otimes \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}_{n \times 1} \otimes \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}_{n \times 1} = \mathbf{aa}^T = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}^T = \begin{bmatrix} a_1 a_1 & a_1 a_2 & \cdots & a_1 a_n \\ a_2 a_1 & a_2 a_2 & \cdots & a_2 a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n a_1 & a_n a_2 & \cdots & a_n a_n \end{bmatrix} \quad (90)$$

请大家注意张量积一些常见性质：

$$\begin{aligned} (\mathbf{a} \otimes \mathbf{a})^T &= \mathbf{a} \otimes \mathbf{a} \\ (\mathbf{a} \otimes \mathbf{b})^T &= \mathbf{b} \otimes \mathbf{a} \\ (\mathbf{a} + \mathbf{b}) \otimes \mathbf{v} &= \mathbf{a} \otimes \mathbf{v} + \mathbf{b} \otimes \mathbf{v} \\ \mathbf{v} \otimes (\mathbf{a} + \mathbf{b}) &= \mathbf{v} \otimes \mathbf{a} + \mathbf{v} \otimes \mathbf{b} \\ t(\mathbf{a} \otimes \mathbf{b}) &= (t\mathbf{a}) \otimes \mathbf{b} = \mathbf{a} \otimes (t\mathbf{b}) \\ (\mathbf{a} \otimes \mathbf{b}) \otimes \mathbf{v} &= \mathbf{a} \otimes (\mathbf{b} \otimes \mathbf{v}) \end{aligned} \quad (91)$$

## 几何视角

图 29 所示为从几何图像角度解释向量的张量积。向量  $\mathbf{a}$  和  $\mathbf{b}$  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  $\mathbf{a} \otimes \mathbf{b}$ 。

当我们关注  $\mathbf{b}$  方向时，网格面沿同一方向的每一条曲线都类似  $\mathbf{b}$ ，唯一的差别是高度上存在一定比例的缩放，这个缩放比例就是  $a_i$ 。 $a_i$  是向量  $\mathbf{a}$  中的某一个元素。

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

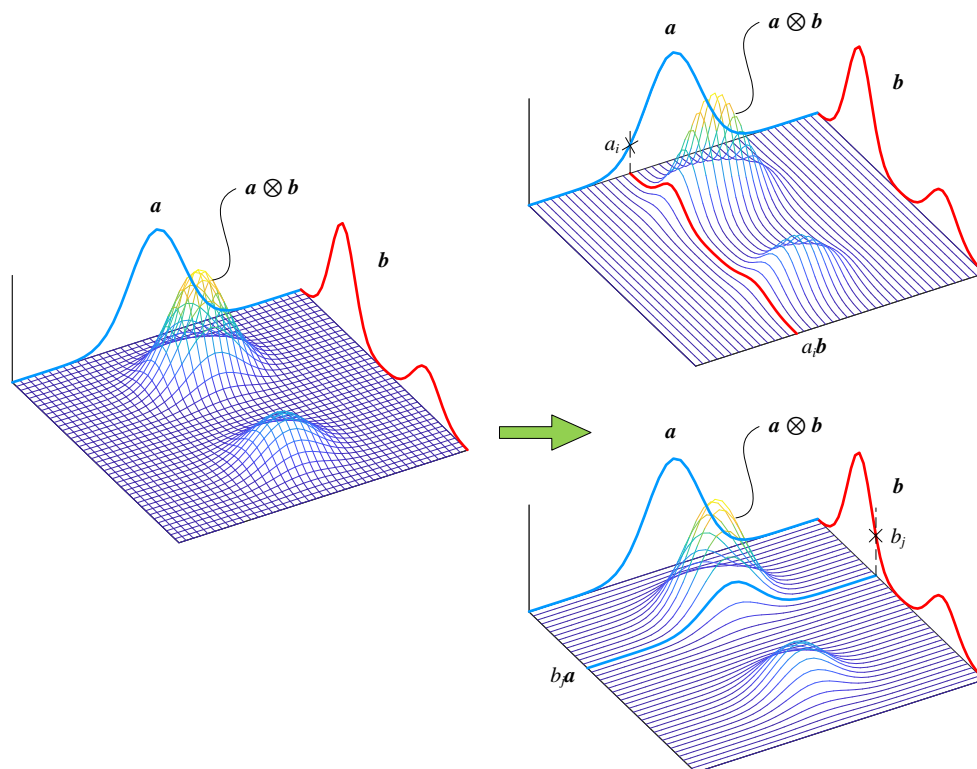


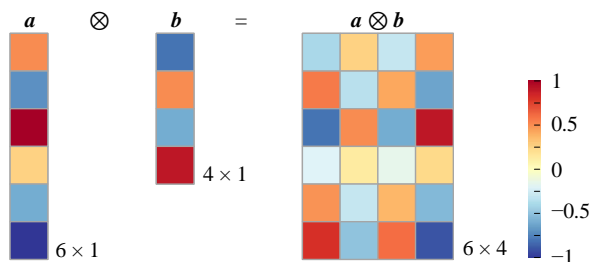
图 29. 从几何角度解释向量张量积

### 举个例子

给定列向量  $a$  和  $b$  分别为：

$$\begin{aligned} a &= [0.5 \quad -0.7 \quad 1 \quad 0.25 \quad -0.6 \quad -1]^T \\ b &= [-0.8 \quad 0.5 \quad -0.6 \quad 0.9]^T \end{aligned} \quad (92)$$

图 30 所示为张量积  $a \otimes b$  结果热图，形状为  $6 \times 4$ 。

图 30. 张量积  $a \otimes b$  热图

观察 (89)，利用矩阵乘法展开，发现  $a \otimes b$  可以写成两种形式：

$$\begin{aligned} \mathbf{a} \otimes \mathbf{b} &= [\mathbf{b}_1 \mathbf{a} \quad \mathbf{b}_2 \mathbf{a} \quad \cdots \quad \mathbf{b}_l \mathbf{a}] \\ \mathbf{a} \otimes \mathbf{b} &= \begin{bmatrix} a_1 \mathbf{b}^T \\ a_2 \mathbf{b}^T \\ \vdots \\ a_n \mathbf{b}^T \end{bmatrix}_{n \times l} \end{aligned} \quad (93)$$

上式中，第一种形式相当于， $\mathbf{a}$  先按不同比例 ( $b_j$ ) 缩放得到  $b_j \mathbf{a}$ ，再左右排列。第二种形式相当于， $\mathbf{b}^T$  先按不同比例 ( $a_i$ ) 缩放得到  $a_i \mathbf{b}^T$ ，再上下叠加。如果对于 (93) 这种矩阵乘法展开方式感到陌生，请大家读完第 4 ~ 6 章后再回头看这部分内容。

如图 31 (a) 所示， $\mathbf{a} \otimes \mathbf{b}$  的每一列都和  $\mathbf{a}$  相似，也就是说它们之间呈现倍数关系。类似地，如图 31 (b) 所示， $\mathbf{a} \otimes \mathbf{b}$  等价于  $\mathbf{a} \mathbf{b}^T$ ，因此  $\mathbf{a} \otimes \mathbf{b}$  每一行都和  $\mathbf{b}^T$  相似，也呈现倍数关系。

➡ 本书第 7 章会聊到向量的秩 (rank)，大家就会知道  $\mathbf{a} \otimes \mathbf{b}$  的秩为 1，就是因为行、列这种“相似”。

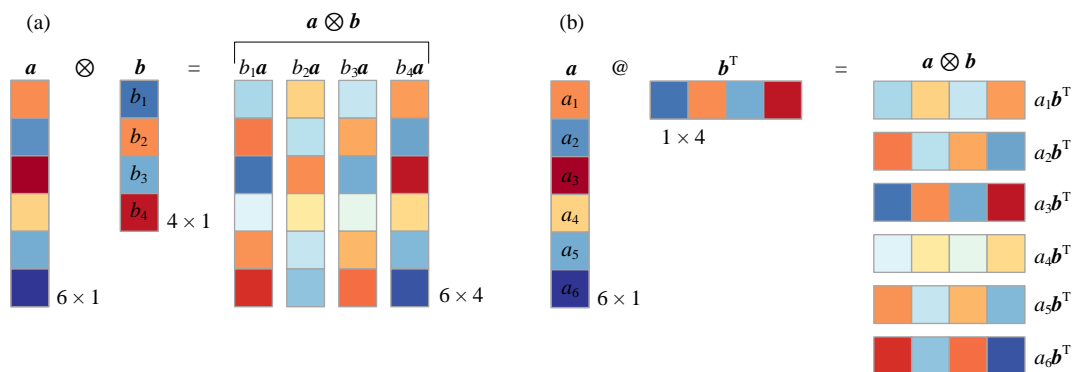


图 31.  $\mathbf{a} \otimes \mathbf{b}$  的列、行存在的相似

图 32 (a) 所示为张量积  $\mathbf{a} \otimes \mathbf{a}$  结果热图，形状为  $6 \times 6$  方阵。图 32 (b) 所示为张量积  $\mathbf{b} \otimes \mathbf{b}$  结果热图，形状为  $4 \times 4$  对称方阵。显然， $\mathbf{a} \otimes \mathbf{a}$  和  $\mathbf{b} \otimes \mathbf{b}$  都是对称矩阵。

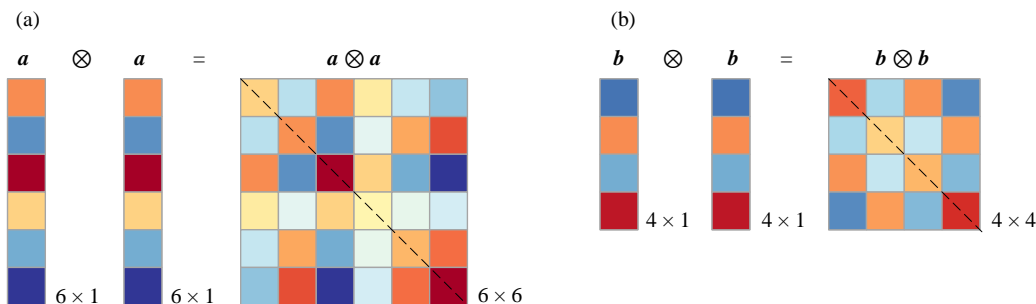
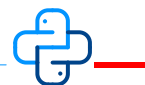


图 32.  $\mathbf{a} \otimes \mathbf{a}$  和  $\mathbf{b} \otimes \mathbf{b}$  向量张量积



Bk4\_Ch2\_13.py 绘制图 30、图 31、图 32。



在 Bk4\_Ch2\_13.py 的基础上，我们用 Streamlit 和 Plotly 制作了一个 App，用来展示向量张量积。App 中，大家可以改变向量元素个数。向量是由随机数发生器产生，保留小数点后一位。请大家参考 Streamlit\_Bk4\_Ch2\_13.py。



《概率统计》将介绍，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  $p_Y(y)$  这两个边缘概率质量函数 PMF 乘积：

$$\underbrace{p_{X,Y}(x,y)}_{\text{Joint}} = \underbrace{p_X(x)}_{\text{Marginal}} \cdot \underbrace{p_Y(y)}_{\text{Marginal}} \quad (94)$$

如图 33 所示， $p_X(x)$  和  $p_Y(y)$  可以分别用火柴梗图可视化，而  $p_{X,Y}(x,y)$  用二维火柴梗图展示。

从线性代数角度，当  $x$  和  $y$  分别取不同值时， $p_X(x)$  和  $p_Y(y)$  相当于两个向量，而  $p_{X,Y}(x,y)$  相当于矩阵。 $X$  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。

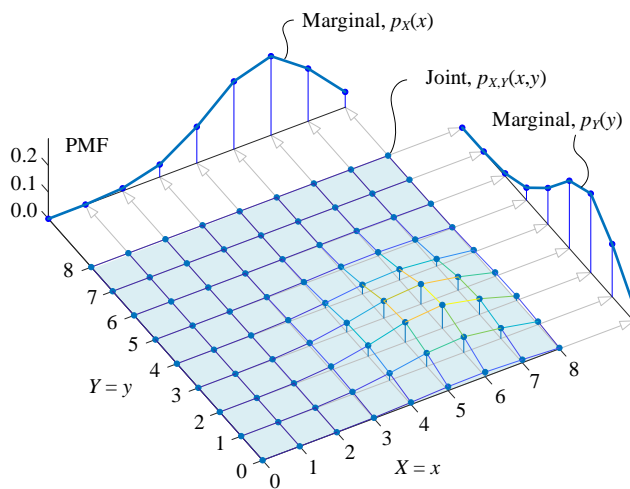


图 33. 离散随机变量独立条件下，联合概率  $p_{X,Y}(x,y)$  等于  $p_Y(y)$  和  $p_X(x)$  乘积



本章聊了聊向量常见运算。学完本章，希望大家看到任何向量和向量运算，可以试着从几何、数据两个角度来思考。

从几何角度，向量是既有长度又有方向的量。从数据角度，表格数据就是矩阵。而矩阵的每一行向量是一个样本点，每一列向量代表一个特征。

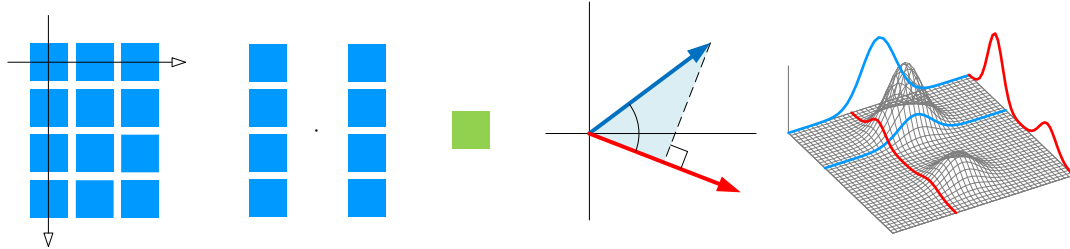


图 34. 总结本章重要内容的四副图

向量有两个元素——长度和方向。向量的长度就是向量的模，向量之间的相对角度可以用向量内积来求解。

提到向量模、 $L^2$  范数、欧几里得距离，希望大家能够联想到正圆、正圆球。本书第 3 章还要介绍更多范数以及它们对应的几何图像。

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  $\Sigma$  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。



对于习惯 MATLAB 或 R 语言的读者，如果转用 Python 感到不适应的话，推荐大家参考：

<http://mathesaurus.sourceforge.net/>

网站整理了常用 MATLAB-R-Python 命令、函数之间关系。