

# 3

## Revisit Distance Metrics

# 再谈距离

丈量事物的尺度，千差万别



人是万物的尺度：是存在事物存在的尺度，也是不存在事物不存在的尺度。

*Man is the measure of all things: of things which are, that they are, and of things which are not, that they are not.*

—— 普罗泰戈拉 (Protagoras) | 古希腊哲学家 | 490 ~ 420 BC



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)



- ◀ `metrics.pairwise.linear_kernel()` 计算线性核成对亲密度矩阵
- ◀ `metrics.pairwise.manhattan_distances()` 计算成对城市街区距离矩阵
- ◀ `metrics.pairwise.paired_cosine_distances(X,Q)` 计算 X 和 Q 样本数据矩阵成对余弦距离矩阵
- ◀ `metrics.pairwise.paired_euclidean_distances(X,Q)` 计算 X 和 Q 样本数据矩阵成对欧氏距离矩阵
- ◀ `metrics.pairwise.paired_manhattan_distances(X,Q)` 计算 X 和 Q 样本数据矩阵成对城市街区距离矩阵
- ◀ `metrics.pairwise.polynomial_kernel()` 计算多项式核成对亲密度矩阵
- ◀ `metrics.pairwise.rbf_kernel()` 计算 RBF 核成对亲密度矩阵
- ◀ `metrics.pairwise.sigmoid_kernel()` 计算 sigmoid 核成对亲密度矩阵
- ◀ `numpy.diag()` 如果 A 为方阵, `numpy.diag(A)` 函数提取对角线元素, 以向量形式输入结果; 如果 a 为向量, `numpy.diag(a)` 函数将向量展开成方阵, 方阵对角线元素为 a 向量元素
- ◀ `numpy.linalg.inv()` 计算逆矩阵
- ◀ `numpy.linalg.norm()` 计算范数
- ◀ `scipy.spatial.distance.chebyshev()` 计算切比雪夫距离
- ◀ `scipy.spatial.distance.cityblock()` 计算城市街区距离
- ◀ `scipy.spatial.distance.euclidean()` 计算欧氏距离
- ◀ `scipy.spatial.distance.mahalanobis()` 计算欧氏距离
- ◀ `scipy.spatial.distance.mahalanobis()` 计算马氏距离
- ◀ `scipy.spatial.distance.minkowski()` 计算闵氏距离
- ◀ `scipy.spatial.distance.seuclidean()` 计算标准化欧氏距离
- ◀ `sklearn.metrics.pairwise.euclidean_distances()` 计算成对欧氏距离矩阵
- ◀ `sklearn.metrics.pairwise_distances()` 计算成对距离矩阵

## 3.1 还聊距离

上一章在讲解  $k$ -NN 分类算法时，默认距离度量为欧几里得距离，实际应用中还有大量其他距离可供选择。

大家对距离这个概念应该非常熟悉，我们从《数学要素》第 7 章“开始就不断丰富“距离”的内涵。我们在《矩阵力量》第 3 章专门介绍了基于  $L^p$  范数的几种距离度量，在《概率统计》第 15 章专门讲解了马氏距离。

本章便专门总结并探讨常用的几个距离度量：

- ◀ 欧氏距离 (Euclidean distance)
- ◀ 标准化欧氏距离 (standardized Euclidean distance)
- ◀ 马氏距离 (Mahalanobis distance, Mahal distance)
- ◀ 城市街区距离 (city block distance)
- ◀ 切比雪夫距离 (Chebyshev distance)
- ◀ 闵氏距离 (Minkowski distance)
- ◀ 余弦距离 (cosine distance)
- ◀ 相关性距离 (correlation distance)

本章最后探讨距离和亲近度的关系。

## 3.2 欧氏距离：最常见的距离

欧几里得距离，也称欧氏距离 (Euclidean distance)。任意样本数据点  $\mathbf{x}$  和查询点  $\mathbf{q}$  欧氏距离定义如下：

$$d(\mathbf{x}, \mathbf{q}) = \|\mathbf{x} - \mathbf{q}\| = \sqrt{(\mathbf{x} - \mathbf{q})^T (\mathbf{x} - \mathbf{q})} \quad (1)$$

其中， $\mathbf{x}$  和  $\mathbf{q}$  为列向量。欧氏距离本质上就是  $\mathbf{x} - \mathbf{q}$  的  $L^2$  范数。从几何视角来看，二维欧氏距离可以看做同心正圆，三维欧氏距离可以视作同心正球体，等等。

当特征数为  $D$  时，上式展开可以得到：

$$d(\mathbf{x}, \mathbf{q}) = \sqrt{(x_1 - q_1)^2 + (x_2 - q_2)^2 + \dots + (x_D - q_D)^2} \quad (2)$$

特别地，当特征数量  $D = 2$  时， $\mathbf{x}$  和  $\mathbf{q}$  两点欧氏距离定义为：

$$d(\mathbf{x}, \mathbf{q}) = \sqrt{(x_1 - q_1)^2 + (x_2 - q_2)^2} \quad (3)$$

### 举个例子

如果查询点  $\mathbf{q}$  有两个特征，并位于原点，即：

$$\mathbf{q} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4)$$

如图 1 所示，三个样本点  $\mathbf{x}^{(1)}$ 、 $\mathbf{x}^{(2)}$  和  $\mathbf{x}^{(3)}$  的位置如下：

$$\mathbf{x}^{(1)} = [-5 \ 0], \mathbf{x}^{(2)} = [4 \ 3], \mathbf{x}^{(3)} = [3 \ -4] \quad (5)$$

根据 (1) 可以计算得到三个样本点  $\mathbf{x}^{(1)}$ 、 $\mathbf{x}^{(2)}$  和  $\mathbf{x}^{(3)}$  距离查询点  $\mathbf{q}$  之间欧氏距离均为 5：

$$\begin{cases} d_1 = \sqrt{([0 \ 0] - [-5 \ 0])([0 \ 0] - [-5 \ 0])^T} = \sqrt{[5 \ 0][5 \ 0]^T} = \sqrt{25 + 0} = 5 \\ d_2 = \sqrt{([0 \ 0] - [4 \ 3])([0 \ 0] - [4 \ 3])^T} = \sqrt{[-4 \ -3][-4 \ -3]^T} = \sqrt{16 + 9} = 5 \\ d_3 = \sqrt{([0 \ 0] - [3 \ -4])([0 \ 0] - [3 \ -4])^T} = \sqrt{[-3 \ 4][-3 \ 4]^T} = \sqrt{9 + 16} = 5 \end{cases} \quad (6)$$

注意，行向量和列向量的转置关系，本章后续不再区分行、列向量。如图 1 所示，当  $d$  取定值时，上式相当于以  $(q_1, q_2)$  为圆心的正圆。

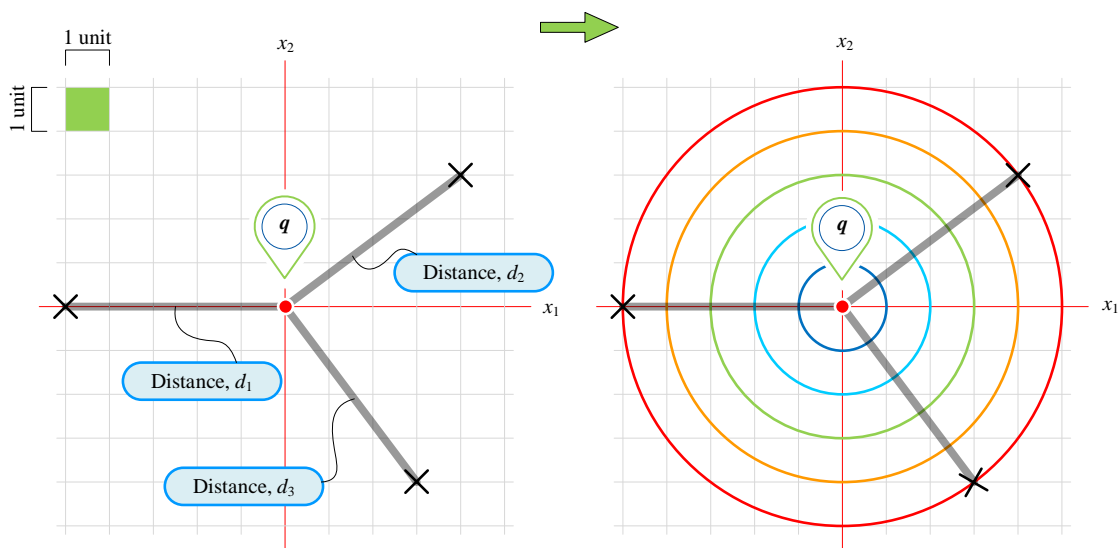


图 1.3 特征 ( $D=2$ ) 欧几里得距离



代码 Bk7\_Ch03\_01.py 计算两点欧氏距离。 `scipy.spatial.distance.euclidean()` 为计算欧氏距离的函数。

## 成对距离

如图 1 所示，三个样本点  $\mathbf{x}^{(1)}$ 、 $\mathbf{x}^{(2)}$  和  $\mathbf{x}^{(3)}$  之间也存在两两距离，我们管它们叫做**成对距离** (pairwise distance)。



代码 Bk7\_Ch03\_02.py 计算图 1 中三个样本点之间的成对欧氏距离。注意，成对结果以矩阵方式呈现。本章最后一节将专门介绍成对距离。

## 3.3 标准化欧氏距离：考虑标准差

### 定义

**标准化欧氏距离** (standardized Euclidean distance) 定义如下。

$$d(\mathbf{x}, \mathbf{q}) = \sqrt{(\mathbf{x} - \mathbf{q})^T \mathbf{D}^{-1} (\mathbf{x} - \mathbf{q})} \quad (7)$$

其中， $\mathbf{D}$  为协方差矩阵  $\Sigma$  对角线元素构造的对角阵，运算如下：

$$\mathbf{V} = \text{diag}(\text{diag}(\Sigma)) = \text{diag} \left( \text{diag} \begin{bmatrix} \sigma_1^2 & \rho_{1,2}\sigma_1\sigma_2 & \cdots & \rho_{1,D}\sigma_1\sigma_D \\ \rho_{2,1}\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2,D}\sigma_2\sigma_D \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{D,1}\sigma_1\sigma_D & \rho_{D,2}\sigma_2\sigma_D & \cdots & \sigma_D^2 \end{bmatrix} \right) = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_D^2 \end{bmatrix} \quad (8)$$

回忆丛书之前介绍过有关 `diag()` 函数的说明。如果  $\mathbf{A}$  为方阵，`diag(A)` 函数提取对角线元素，结果为向量；如果  $\mathbf{a}$  为向量，`diag(a)` 函数将向量  $\mathbf{a}$  展开成对角方阵，方阵对角线元素为  $\mathbf{a}$  向量元素。Numpy 中完成这一计算的函数为 `numpy.diag()`。

将 (8) 带入 (7) 得到：

$$\begin{aligned} d(\mathbf{x}, \mathbf{q}) &= \sqrt{\begin{bmatrix} x_1 - q_1 & x_2 - q_2 & \cdots & x_D - q_D \end{bmatrix} \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_D^2 \end{bmatrix}^{-1} \begin{bmatrix} x_1 - q_1 & x_2 - q_2 & \cdots & x_D - q_D \end{bmatrix}^T} \\ &= \sqrt{\frac{(x_1 - q_1)^2}{\sigma_1^2} + \frac{(x_2 - q_2)^2}{\sigma_2^2} + \cdots + \frac{(x_D - q_D)^2}{\sigma_D^2}} = \sqrt{\sum_{j=1}^D \left( \frac{x_j - q_j}{\sigma_j} \right)^2} \end{aligned} \quad (9)$$

(9) 可以记做：

$$d(\mathbf{x}, \mathbf{q}) = \sqrt{z_1^2 + z_2^2 + \dots + z_D^2} = \sqrt{\sum_{j=1}^D z_j^2} \quad (10)$$

其中,  $z_j$  为:

$$z_j = \frac{x_j - q_j}{\sigma_j} \quad (11)$$

上式类似 z 分数。

## 正椭圆

对于  $D = 2$ , 两特征的情况, 标准化欧氏距离平方可以写成:

$$d^2 = \frac{(x_1 - q_1)^2}{\sigma_1^2} + \frac{(x_2 - q_2)^2}{\sigma_2^2} \quad (12)$$

可以发现, 上式代表的形状是以  $(q_1, q_2)$  为中心的正椭圆。观察 (12), 可以发现, 标准化欧氏距离引入数据每个特征均方差, 但是没有考虑特征之间的相关性。图 2 中, 网格的坐标已经转化为“均方差”, 而标准欧氏距离等距线为正椭圆。

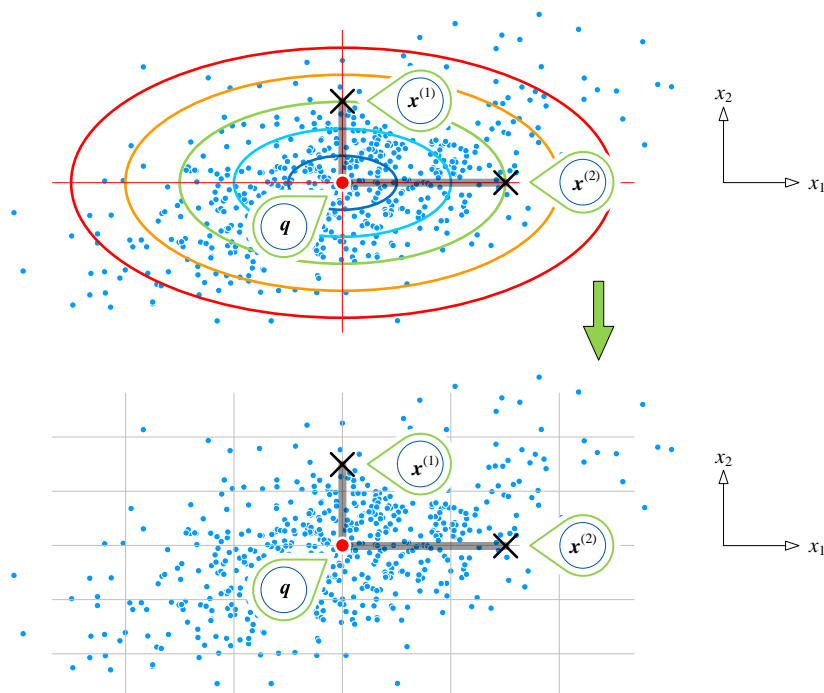


图 2.2 特征 ( $D = 2$ ) 标准化欧氏距离

## 几何变换视角

如图 3 所示，从几何变换角度，标准化欧氏距离相当于对  $X$  数据每个维度，首先**中心化** (centralize)，然后利用均方差进行**缩放** (scale)；但是，标准化欧氏距离没有旋转操作，也就是没有正交化。

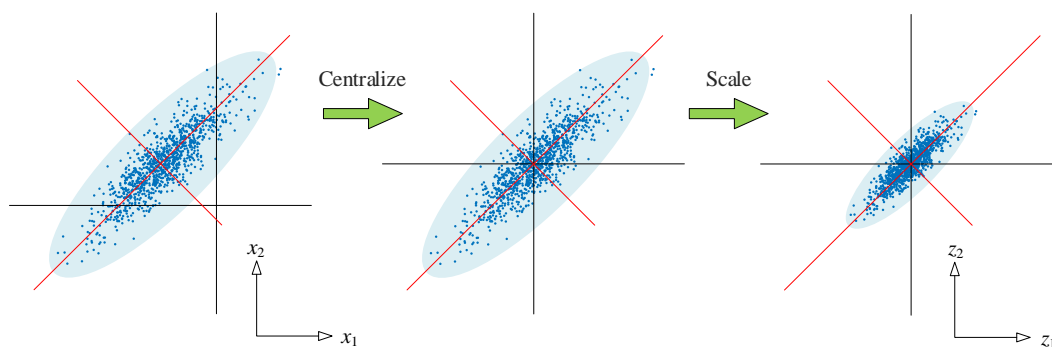


图 3. 标准化欧氏距离运算过程



计算标准化欧氏距离的函数为 `scipy.spatial.distance.seuclidean()`。代码 Bk7\_Ch03\_03.py 计算本节标准化欧氏距离。

## 3.4 马氏距离：考虑标准差和相关性

本系列丛书《矩阵力量》和《概率统计》从不同角度讲过马氏距离，本节稍作回忆。**马氏距离**，**马哈距离** (Mahalanobis distance, Mahal distance)，全称马哈拉诺比斯距离。马氏距离定义如下：

$$d(x, q) = \sqrt{(x - q)^T \Sigma^{-1} (x - q)} \quad (13)$$

其中， $\Sigma$  为样本数据  $X$  协方差矩阵。注意，马氏距离的单位是“标准差”。比如，马氏距离计算结果为 3，应该称作 3 个标准差。

### 特征值分解：缩放 → 旋转 → 平移

$\Sigma$  特征值分解得到：

$$\Sigma = V \Lambda V^T \quad (14)$$

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)

其中，由于  $\Sigma$  为对称矩阵，因此  $V$  为正交矩阵。 $\Sigma^{-1}$  的特征值分解可以写成：

$$\Sigma^{-1} = (V \Lambda V^T)^{-1} = (V^T)^{-1} \Lambda^{-1} V^{-1} = V \Lambda^{-1} V^T \quad (15)$$

将 (15) 代入 (13) 得到：

$$d(\mathbf{x}, \mathbf{q}) = \sqrt{\left[ \begin{array}{c} \Lambda^{-\frac{1}{2}} V^T \\ \text{Scale Rotate Centralize} \end{array} (\mathbf{x} - \mathbf{q}) \right]^T \left[ \Lambda^{-\frac{1}{2}} V^T (\mathbf{x} - \mathbf{q}) \right]} \quad (16)$$

其中， $\mathbf{q}$  列向量完成中心化 (centralize)， $V$  矩阵完成旋转 (rotate)， $\Lambda$  矩阵完成缩放 (scale)。大家如果对这部分内容感到陌生，请回顾《矩阵力量》第 20 章。

### 旋转椭圆

如图 4 所示，当  $D = 2$  时，马氏距离的等距线为旋转椭圆。

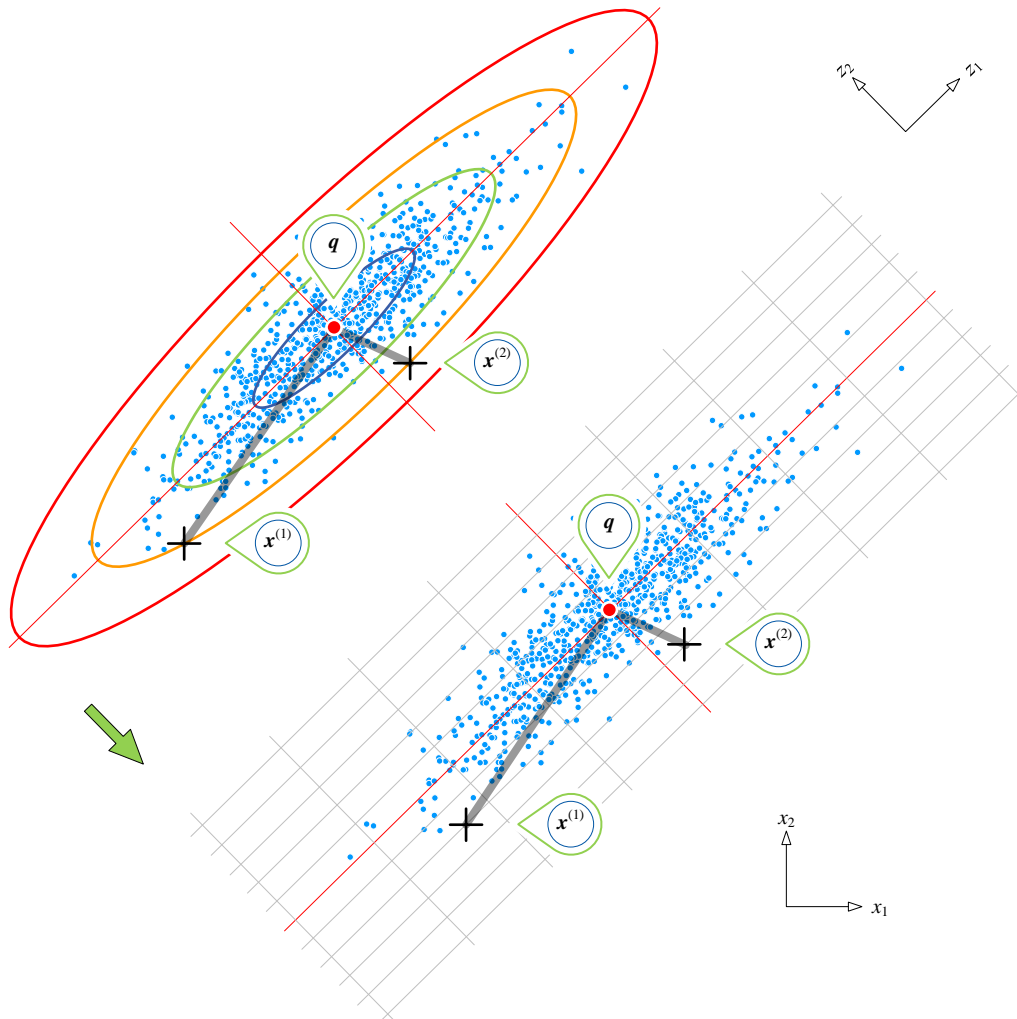


图 4.2 特征 ( $D = 2$ ) 马氏距离





代码 Bk7\_Ch03\_04.py 计算图 4 两个点的马氏距离。

## 3.5 城市街区距离： $L^1$ 范数

城市街区距离 (city block distance), 也称曼哈顿距离 (Manhattan distance), 具体定义如下:

$$d(\mathbf{x}, \mathbf{q}) = \|\mathbf{x} - \mathbf{q}\|_1 = \sum_{j=1}^D |x_j - q_j| \quad (17)$$

其中,  $j$  代表特征序号。城市街区距离就是我们在《矩阵力量》第 3 章中介绍的  $L^1$  范数。

将 (17) 展开得到下式:

$$d(\mathbf{x}, \mathbf{q}) = |x_1 - q_1| + |x_2 - q_2| + \dots + |x_D - q_D| \quad (18)$$

特别地, 当  $D = 2$  时, 城市街区距离为:

$$d(\mathbf{x}, \mathbf{q}) = |x_1 - q_1| + |x_2 - q_2| \quad (19)$$

### 旋转正方形

如图 5 所示, 城市街区距离的等距线为旋转正方形。图中,  $\mathbf{x}^{(1)}$ 、 $\mathbf{x}^{(2)}$  和  $\mathbf{x}^{(3)}$  和  $\mathbf{q}$  欧氏距离均为 5, 但是城市街区距离分别为 5、7 和 7。

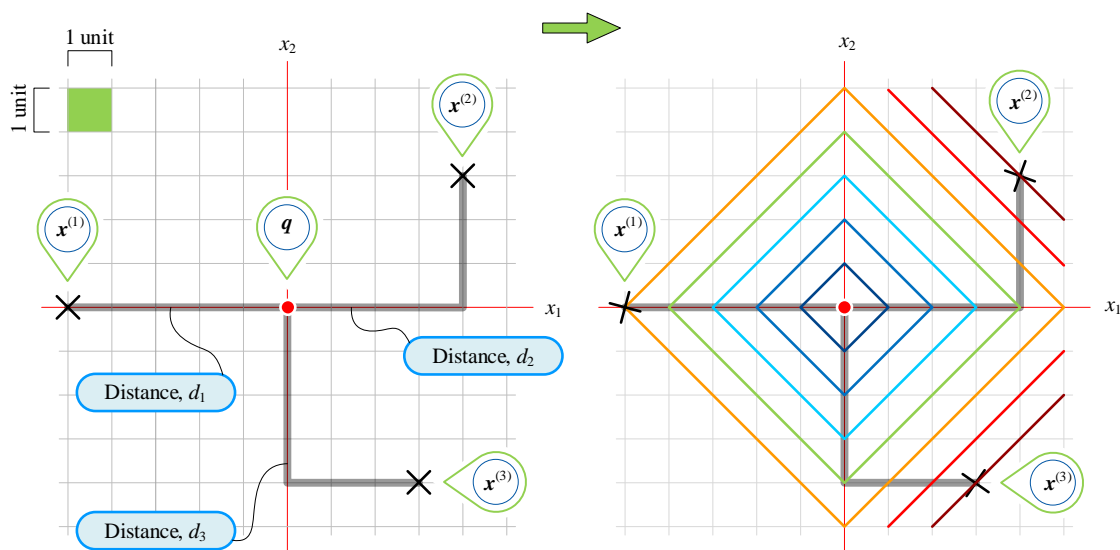


图 5.2 特征 ( $D = 2$ ) 城市街区距离



代码 Bk7\_Ch03\_05.py 给出两种方法计算得到图 5 所示城市街区距离。

## 3.6 切比雪夫距离： $L^\infty$ 范数

切比雪夫距离 (Chebyshev distance)，具体如下。

$$d(\mathbf{x}, \mathbf{q}) = \|\mathbf{x} - \mathbf{q}\|_\infty = \max_j \{|x_j - q_j|\} \quad (20)$$

切比雪夫距离就是我们在《矩阵力量》第 3 章中介绍的  $L^\infty$  范数。

将 (20) 展开得到下式：

$$d(\mathbf{x}, \mathbf{q}) = \max \{|x_1 - q_1|, |x_2 - q_2|, \dots, |x_D - q_D|\} \quad (21)$$

特别地，当  $D = 2$  时，切比雪夫距离为：

$$d(\mathbf{x}, \mathbf{q}) = \max \{|x_1 - q_1|, |x_2 - q_2|\} \quad (22)$$

### 正方形

如图 6 所示，切比雪夫距离等距线为正方形。前文提到， $\mathbf{x}^{(1)}$ 、 $\mathbf{x}^{(2)}$  和  $\mathbf{x}^{(3)}$  和  $\mathbf{q}$  欧氏距离相同，但是切比雪夫距离分别为 5、4 和 4。

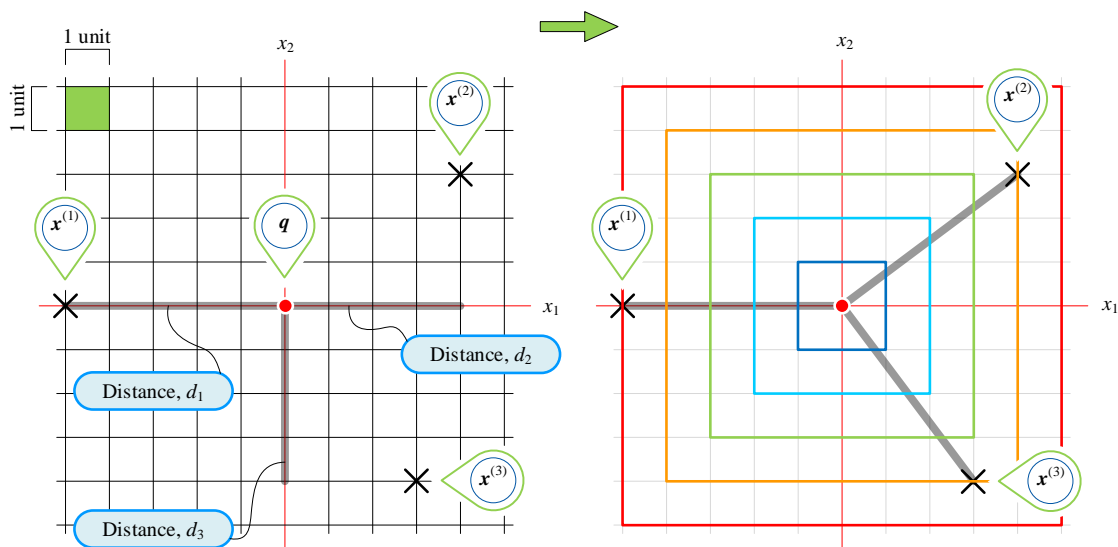


图 6.2 特征 ( $D = 2$ ) 切比雪夫距离



代码 Bk7\_Ch03\_06.py 计算图 6 所示切比雪夫距离。

## 3.7 闵氏距离： $L^p$ 范数

**闵氏距离** (Minkowski distance) 类似  $L^p$  范数，对应定义如下：

$$d(\mathbf{x}, \mathbf{q}) = \|\mathbf{x} - \mathbf{q}\|_p = \left( \sum_{j=1}^D |x_j - q_j|^p \right)^{1/p} \quad (23)$$

注意， $p \geq 1$ 。计算闵氏距离的函数为 `scipy.spatial.distance.minkowski()`。

图 7 所示为  $p$  取不同值时，闵氏距离等距线图。特别地， $p = 1$  时，闵氏距离为城市街区距离； $p = 2$  时，闵氏距离为欧氏距离； $p \rightarrow \infty$  时，闵氏距离为切比雪夫距离。

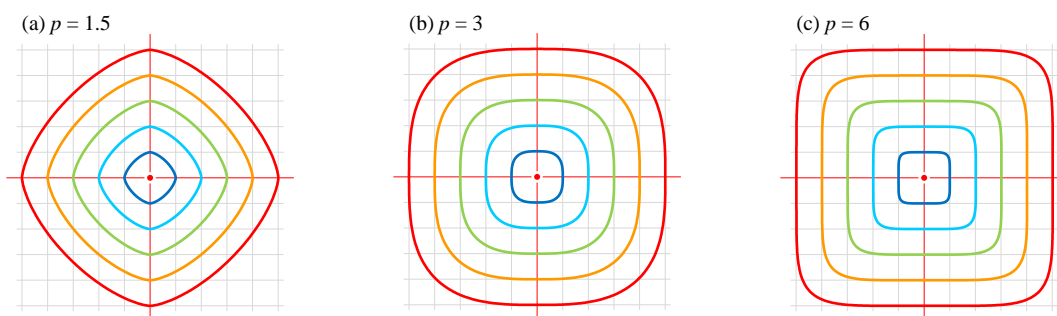


图 7. 闵氏距离 ( $D = 2$ ),  $p$  取不同值

## 3.8 距离与亲近

本节介绍和距离相反的度量——**亲近度** (affinity)。两个样本数据距离越远，两者亲近度越低；而当它们距离越近，亲近度则越高。亲近度，也称**相似度** (similarity)。

### 余弦相似度

《矩阵力量》第 2 章讲过，**余弦相似度** (cosine similarity) 用向量夹角的余弦值度量样本数据的相似性。 $\mathbf{x}$  和  $\mathbf{q}$  两个向量的余弦相似，具体定义如下：

$$k(\mathbf{x}, \mathbf{q}) = \frac{\mathbf{x}^T \mathbf{q}}{\|\mathbf{x}\| \|\mathbf{q}\|} = \frac{\mathbf{x} \cdot \mathbf{q}}{\|\mathbf{x}\| \|\mathbf{q}\|} \quad (24)$$

如图 8 所示，如果两个向量方向相同，则夹角  $\theta$  余弦值  $\cos(\theta)$  为 1；如果，两个向量方向完全相反，夹角  $\theta$  余弦值  $\cos(\theta)$  为 -1。因此余弦相似度取值范围在  $[-1, +1]$  之间。

注意，余弦相似度和向量长度无关，仅仅与两个向量夹角有关。

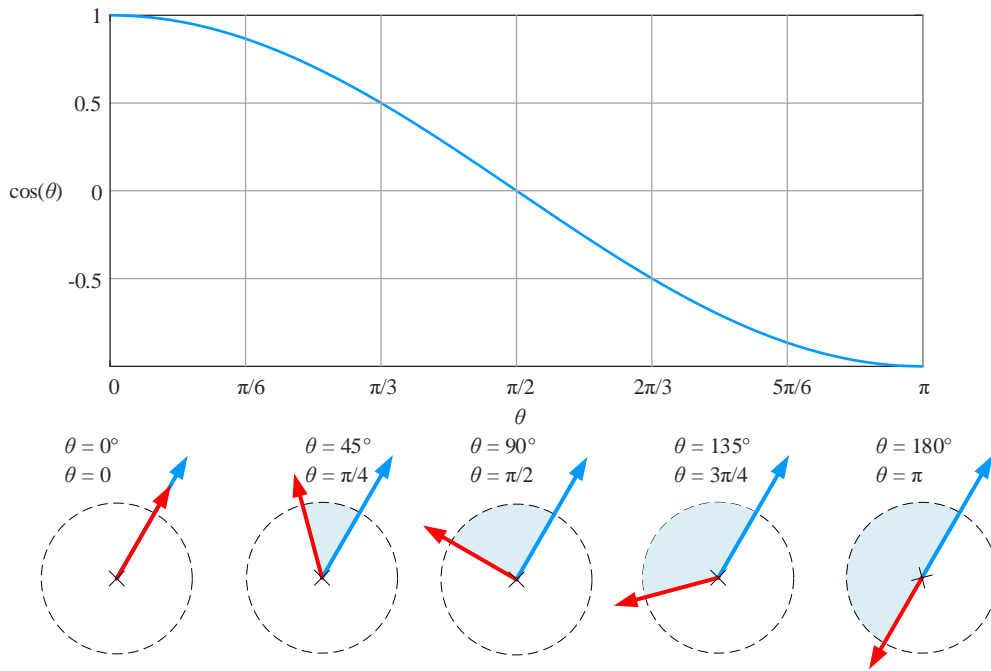


图 8. 余弦相似度

## 举个例子

给定如下两个向量具体值：

$$\mathbf{x} = [8 \ 2]^T, \quad \mathbf{q} = [7 \ 9]^T \quad (25)$$

将 (25) 代入 (24) 得到：

$$k(\mathbf{x}, \mathbf{q}) = \frac{\mathbf{x} \cdot \mathbf{q}}{\|\mathbf{x}\| \|\mathbf{q}\|} = \frac{8 \times 7 + 2 \times 9}{\sqrt{8^2 + 2^2} \times \sqrt{7^2 + 9^2}} = \frac{74}{\sqrt{68} \times \sqrt{130}} = 0.7871 \quad (26)$$



代码 Bk7\_Ch03\_07.py 得到和 (26) 一致结果。

## 余弦距离

**余弦距离** (cosine distance) 的定义如下：

$$d(\mathbf{x}, \mathbf{q}) = 1 - k(\mathbf{x}, \mathbf{q}) = 1 - \frac{\mathbf{x}^T \mathbf{q}}{\|\mathbf{x}\| \|\mathbf{q}\|} = 1 - \frac{\mathbf{x} \cdot \mathbf{q}}{\|\mathbf{x}\| \|\mathbf{q}\|} \quad (27)$$

余弦相似度的取值范围  $[-1, +1]$  之间，因此余弦距离的取值范围为  $[0, 2]$ 。



代码计算 (25) 中两个向量的余弦距离，结果为 0.2129。也可以采用 `scipy.spatial.distance.pdist(X, 'cosine')` 函数计算余弦距离。

## 相关系数相似度

**相关系数相似度** (correlation similarity) 定义如下：

$$k(\mathbf{x}, \mathbf{q}) = \frac{(\mathbf{x} - \bar{\mathbf{x}})^T (\mathbf{q} - \bar{\mathbf{q}})}{\|\mathbf{x} - \bar{\mathbf{x}}\| \|\mathbf{q} - \bar{\mathbf{q}}\|} = \frac{(\mathbf{x} - \bar{\mathbf{x}}) \cdot (\mathbf{q} - \bar{\mathbf{q}})}{\|\mathbf{x} - \bar{\mathbf{x}}\| \|\mathbf{q} - \bar{\mathbf{q}}\|} \quad (28)$$

其中， $\bar{\mathbf{x}}$  为列向量  $\mathbf{x}$  元素均值； $\bar{\mathbf{q}}$  为列向量  $\mathbf{q}$  元素均值。

观察 (28)，发现相关系数相似度类似余弦相似度；稍有不同的是，相关系数相似度需要“中心化”向量。

还是以 (25) 为例，计算  $\mathbf{x}$  和  $\mathbf{q}$  两个向量的相关系数相似度。将 (25) 代入 (28) 可以得到：

$$\begin{aligned} k(\mathbf{x}, \mathbf{q}) &= \frac{\left( \begin{bmatrix} 8 & 2 \end{bmatrix}^T - \frac{8+2}{2} \right) \cdot \left( \begin{bmatrix} 7 & 9 \end{bmatrix}^T - \frac{7+9}{2} \right)}{\left\| \begin{bmatrix} 8 & 2 \end{bmatrix}^T - \frac{8+2}{2} \right\| \left\| \begin{bmatrix} 7 & 9 \end{bmatrix}^T - \frac{7+9}{2} \right\|} \\ &= \frac{\begin{bmatrix} 3 & -3 \end{bmatrix}^T \cdot \begin{bmatrix} -1 & 1 \end{bmatrix}^T}{\left\| \begin{bmatrix} 3 & -3 \end{bmatrix}^T \right\| \left\| \begin{bmatrix} -1 & 1 \end{bmatrix}^T \right\|} = \frac{-6}{6} = -1 \end{aligned} \quad (29)$$

从相关系数相似度可以计算得到相关系数距离：

$$d(\mathbf{x}, \mathbf{q}) = 1 - k(\mathbf{x}, \mathbf{q}) = 1 - \frac{(\mathbf{x} - \bar{\mathbf{x}})^T (\mathbf{q} - \bar{\mathbf{q}})}{\|\mathbf{x} - \bar{\mathbf{x}}\| \|\mathbf{q} - \bar{\mathbf{q}}\|} = 1 - \frac{(\mathbf{x} - \bar{\mathbf{x}}) \cdot (\mathbf{q} - \bar{\mathbf{q}})}{\|\mathbf{x} - \bar{\mathbf{x}}\| \|\mathbf{q} - \bar{\mathbf{q}}\|} \quad (30)$$



代码 Bk7\_Ch03\_09.py 计算得到两个向量的相关系数距离为 2。也可以采用 `scipy.spatial.distance.pdist(X, 'correlation')` 函数计算相关系数距离。

### 核函数亲密度

不考虑常数项，**线性核** (linear kernel) 亲密度定义如下：

$$\kappa(\mathbf{x}, \mathbf{q}) = \mathbf{x}^T \mathbf{q} = \mathbf{x} \cdot \mathbf{q} \quad (31)$$

对比 (24) 和 (31)，(24) 分母上  $\|\mathbf{x}\|$  和  $\|\mathbf{q}\|$  分别对  $\mathbf{x}$  和  $\mathbf{q}$  归一化。

`sklearn.metrics.pairwise.linear_kernel` 为 scikit-learn 工具箱中计算线性核亲密度函数。

将 (25) 代入 (31)，得到线性核亲密度为：

$$\kappa(\mathbf{x}, \mathbf{q}) = 8 \times 7 + 2 \times 9 = 74 \quad (32)$$

**多项式核** (polynomial kernel) 亲密度定义如下：

$$\kappa(\mathbf{x}, \mathbf{q}) = (\gamma \mathbf{x}^T \mathbf{q} + r)^d = (\gamma \mathbf{x} \cdot \mathbf{q} + r)^d \quad (33)$$

其中， $d$  为多项式核次数， $\gamma$  为系数， $r$  为常数。

多项式核亲密度函数为 `sklearn.metrics.pairwise.polynomial_kernel`。

**Sigmoid 核** (sigmoid kernel) 亲密度定义如下：

$$\kappa(\mathbf{x}, \mathbf{q}) = \tanh(\gamma \mathbf{x}^T \mathbf{q} + r) = \tanh(\gamma \mathbf{x} \cdot \mathbf{q} + r) \quad (34)$$

Sigmoid 核亲密度函数为 `sklearn.metrics.pairwise.sigmoid_kernel`。

最常见的莫过于，**高斯核** (Gaussian kernel) 亲密度，即**径向基核函数** (radial basis function kernel, RBF kernel)：

$$\kappa(\mathbf{x}, \mathbf{q}) = \exp(-\gamma \|\mathbf{x} - \mathbf{q}\|^2) \quad (35)$$

注意到，(35) 中  $\|\mathbf{x} - \mathbf{q}\|^2$  为欧氏距离的平方；因此，(35) 也可以写作：

$$\kappa(\mathbf{x}, \mathbf{q}) = \exp(-\gamma d^2) \quad (36)$$

其中， $d$  为欧氏距离  $\|\mathbf{x} - \mathbf{q}\|$ 。高斯核亲密度取值范围为  $[0, 1]$ ；距离值越小，亲密度越高。高斯核亲密度函数为 `sklearn.metrics.pairwise.rbf_kernel`。

图 9 所示为,  $\gamma$  取不同值时, 高斯核亲密度随着欧氏距离  $d$  变化。聚类算法经常采用高斯核亲密度。

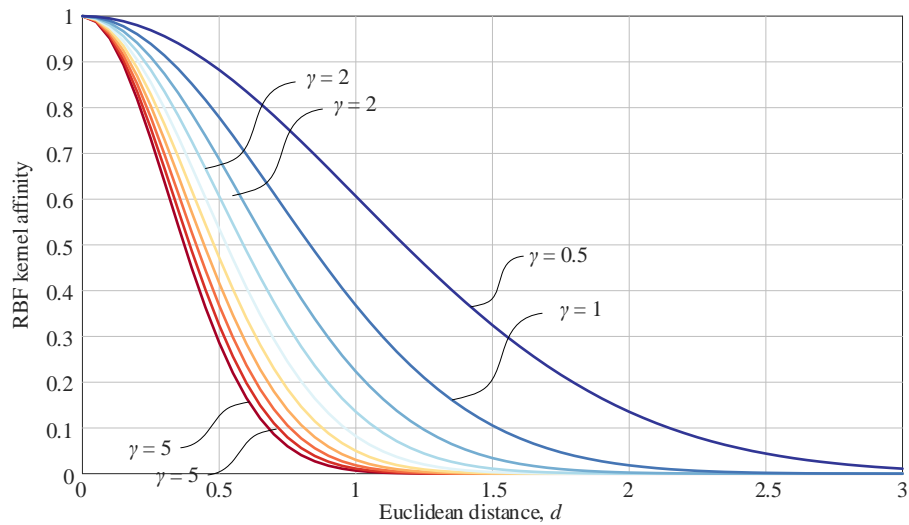


图 9. 高斯核亲密度随欧氏距离变化

**拉普拉斯核** (Laplacian kernel) 亲密度, 定义如下:

$$\kappa(\mathbf{x}, \mathbf{q}) = \exp(-\gamma \|\mathbf{x} - \mathbf{q}\|_1) \quad (37)$$

其中,  $\|\mathbf{x} - \mathbf{q}\|_1$  为城市街区距离。

图 10 所示为,  $\gamma$  取不同值时, 拉普拉斯核亲密度随着城市街区距离  $d$  变化。拉普拉斯核亲密度对应函数为 `sklearn.metrics.pairwise.laplacian_kernel`。

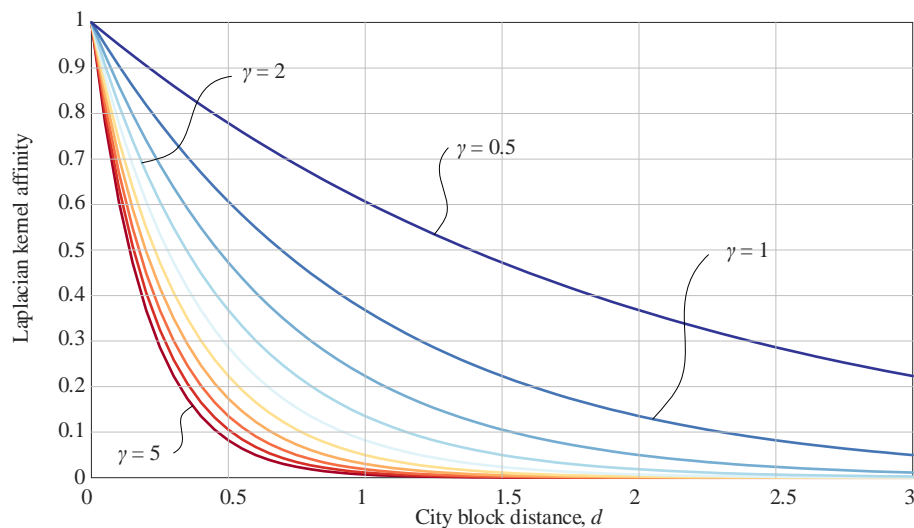


图 10. 拉普拉斯核亲密度随距离变化

## 3.9 成对距离、成对亲近度

《矩阵力量》反复强调，样本数据矩阵  $X$  每一列代表一个特征，而每一行代表一个样本数据点，比如：

$$X_{n \times D} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(n)} \end{bmatrix} \quad (38)$$

$X$  样本点之间距离构成的**成对距离矩阵** (pairwise distance matrix) 形式如下：

$$D_{n \times n} = \begin{bmatrix} 0 & d_{1,2} & d_{1,3} & \cdots & d_{1,n} \\ d_{2,1} & 0 & d_{2,3} & \cdots & d_{2,n} \\ d_{3,1} & d_{3,2} & 0 & \cdots & d_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & d_{n,3} & \cdots & 0 \end{bmatrix} \quad (39)$$

每个样本数据点和自身的距离为 0，因此 (39) 主对角线为 0。很显然矩阵  $D$  为对称矩阵，即  $d_{i,j}$  和  $d_{j,i}$  相等。

图 11 给定 12 个样本数据点坐标点。

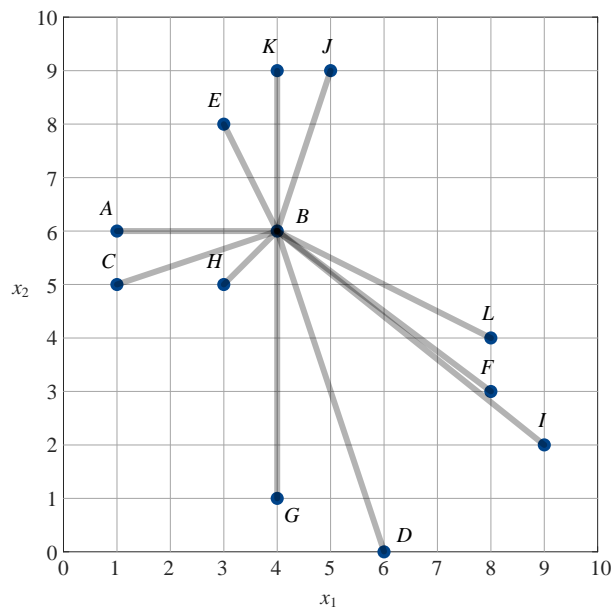


图 11. 样本数据散点图和成对距离

利用 `sklearn.metrics.pairwise.euclidean_distances`，我们可以计算图 11 数据点的成对欧氏距离矩阵。图 12 所示为欧氏距离矩阵数据构造的热图。



实际上，我们关心的成对距离个数为：

$$C_n^2 = \frac{n(n-1)}{2} \quad (40)$$

也就是说，(39) 中不含对角线的下三角矩阵包含的信息足够使用。

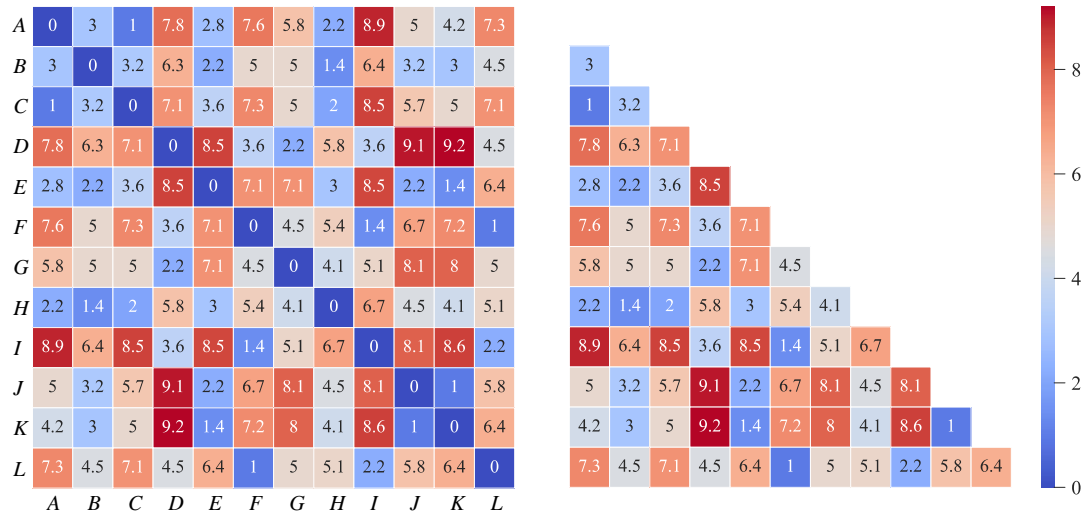
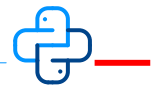


图 12. 样本数据成对距离矩阵热图

表 1 总结计算成对距离、亲密度矩阵常用函数。

表 1. 计算成对距离/亲密度矩阵常见函数

函数	描述
<code>metrics.pairwise.cosine_similarity()</code>	计算余弦相似度成对矩阵
<code>metrics.pairwise.cosine_distances()</code>	计算成对相似性距离矩阵
<code>metrics.pairwise.euclidean_distances()</code>	计算成对欧氏距离矩阵
<code>metrics.pairwise.laplacian_kernel()</code>	计算拉普拉斯核成对亲密度矩阵
<code>metrics.pairwise.linear_kernel()</code>	计算线性核成对亲密度矩阵
<code>metrics.pairwise.manhattan_distances()</code>	计算成对城市街区距离矩阵
<code>metrics.pairwise.polynomial_kernel()</code>	计算多项式核成对亲密度矩阵
<code>metrics.pairwise.rbf_kernel()</code>	计算 RBF 核成对亲密度矩阵
<code>metrics.pairwise.sigmoid_kernel()</code>	计算 sigmoid 核成对亲密度矩阵
<code>metrics.pairwise.paired_euclidean_distances(X,Q)</code>	计算 X 和 Q 样本数据矩阵成对欧氏距离矩阵
<code>metrics.pairwise.paired_manhattan_distances(X,Q)</code>	计算 X 和 Q 样本数据矩阵成对城市街区距离矩阵
<code>metrics.pairwise.paired_cosine_distances(X,Q)</code>	计算 X 和 Q 样本数据矩阵成对余弦距离矩阵



代码 Bk7\_Ch03\_10.py 可以绘制图 11、图 12。

## 树形图

图 12 数据矩阵是很多机器学习算法的起点；看似杂乱无章的图 12，实际上隐含很多重要信息。下面介绍**树形图** (dendrogram)，让大家领略成对距离/亲近度矩阵的力量。

下载 12 只股票历史股价，初值归一走势如图 13 所示。计算日对数回报率，然后估算相关系数矩阵，如图 14 热图所示。相关系数相当于亲近度，相关系数越高，说明股票涨跌趋势越相似。利用树形图，我们可以清楚看到各种股票之间的关联。

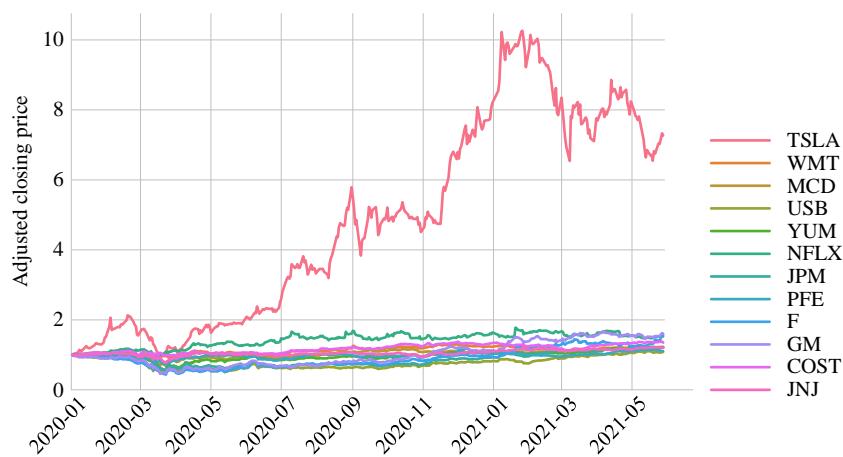


图 13. 12 只股票股价水平，初始股价归一化

PFE 和 JNJ 同属医疗，WMT 和 COST 同属零售，F 和 GM 同属汽车，USB 和 JPM 同属金融，MCD 和 YUM 同属餐饮；因此，它们之间相关性高并不足为奇。但是，本应该离汽车更近的 TSLA，却展现出和 NFLX 更高的相似性。

图 15 给出的树形图，直观地表达样本数据之间的距离/亲密度关系。树形图纵坐标高度表达不同数据之间的距离。

USB 和 JPM 之间相关性系数最高，因此 USB 和 JPM 距离最近，所以在树形图中首先将这两个节点相连，形成一个新的节点。然后，MCD 和 YUM 形成一个节点，F 和 GM 形成一个节点... 依据这种方式，树形自下而上不断聚拢。有关树形图的原理，本书将在层次聚类一章中讲解。

图 15 树形图将股票按照相似度重新排列顺序。图 15 热图发生有意思的变化，热图中出现一个个色彩相近“方块”。每一个“方块”实际上代表着一类相似的数据点。因此，树形图很好揭示股票之间的相似性关系，这便是**聚类** (clustering) 算法的一种思路。

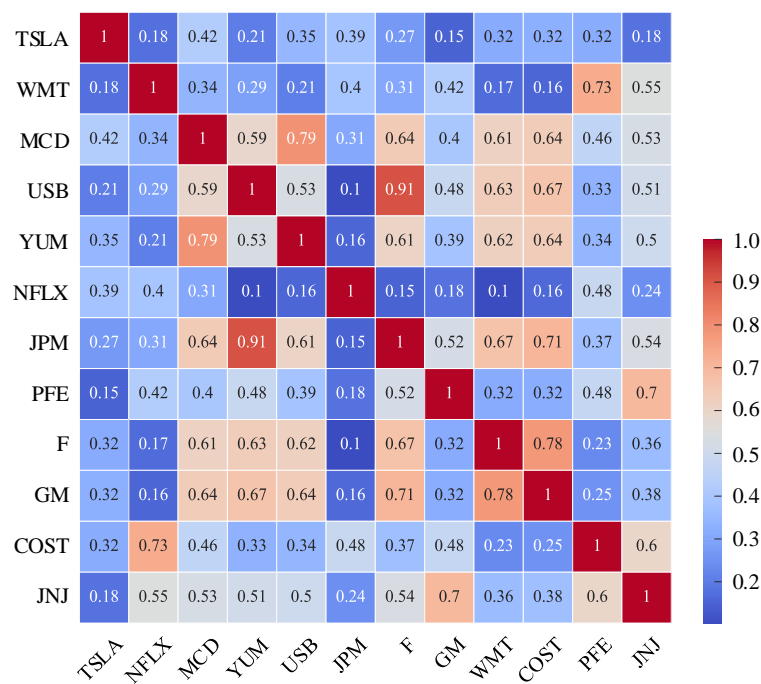


图 14.12 只股票相关性热图

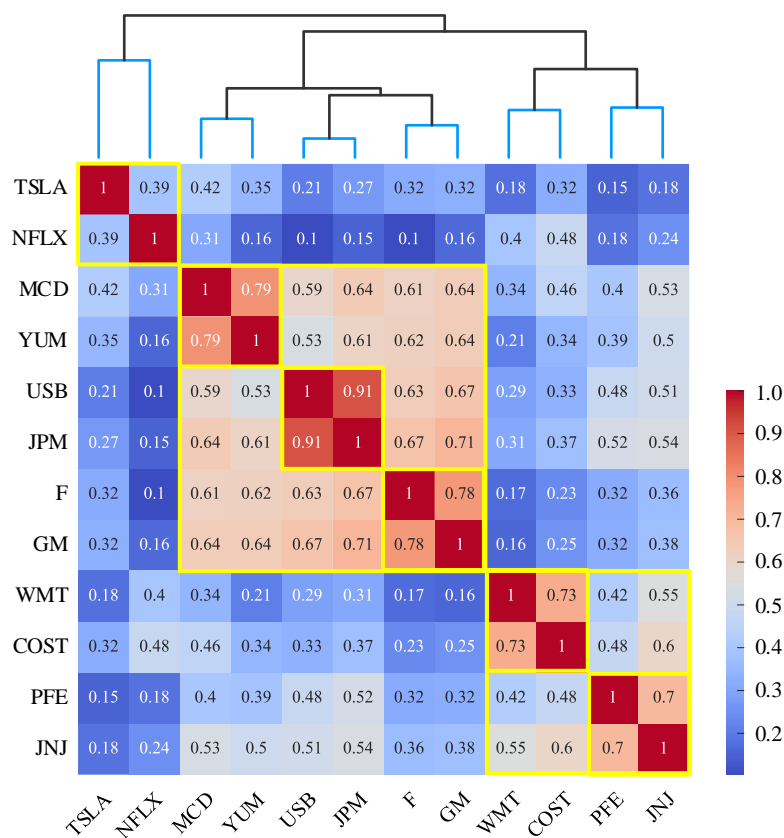


图 15. 根据树形图重组相关性热图

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

版权归清华大学出版社所有，请勿商用，引用请注明出处。

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

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

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



代码 Bk7\_Ch03\_11.py 绘制图 13、图 14 和图 15。



本章主要介绍了几种常见的距离度量和亲近度。机器学习中的距离，并不简单指的是“两点一线”，需要具体问题具体分析。特别希望读者能够结合丛书之前讲解的有关椭圆、矩阵转化和统计相关内容，强化对马氏距离的理解。此外，“远亲不如近邻”，两个点距离越近，两个点的“亲近度”或“相似度”也就越高。