3 再谈距离

Revisit Distance Metrics

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

- `metrics.pairwise.linear_kernel()` 计算线性核成对亲近度矩阵
- `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近度矩阵
- `metrics.pairwise.rbf_kernel()` 计算 RBF 核成对亲近度矩阵
- `metrics.pairwise.sigmoid_kernel()` 计算 sigmoid 核成对亲近度矩阵
- `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 $\boldsymbol{x}$ 和查询点 $\boldsymbol{q}$ 欧氏距离定义如下：

$$d(\boldsymbol{x},\boldsymbol{q}) = \|\boldsymbol{x}-\boldsymbol{q}\| = \sqrt{(\boldsymbol{x}-\boldsymbol{q})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{q})} \tag{1}$$

其中，$\boldsymbol{x}$ 和 $\boldsymbol{q}$ 为列向量。欧氏距离本质上就是 $\boldsymbol{x}-\boldsymbol{q}$ 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：

$$d(\boldsymbol{x},\boldsymbol{q}) = \sqrt{(x_1-q_1)^2+(x_2-q_2)^2+\ldots+(x_D-q_D)^2} \tag{2}$$

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

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

举个例子

如果查询点 $\boldsymbol{q}$ 有两个特征，并位于原点，即：

$$\boldsymbol{q}=\begin{bmatrix}0\\0\end{bmatrix} \tag{4}$$

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

$$\boldsymbol{x}^{(1)}=\begin{bmatrix}-5 & 0\end{bmatrix},\quad \boldsymbol{x}^{(3)}=\begin{bmatrix}4 & 3\end{bmatrix},\quad \boldsymbol{x}^{(3)}=\begin{bmatrix}3 & -4\end{bmatrix} \tag{5}$$

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

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

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。

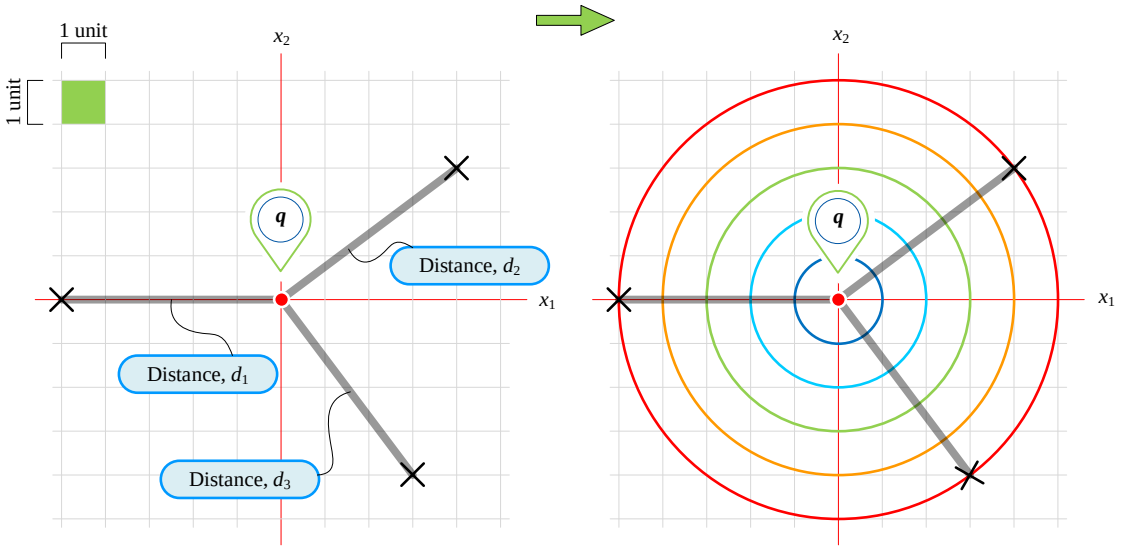
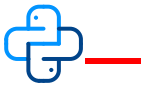


图 1. 3 特征 (D = 2) 欧几里得距离

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 $\boldsymbol{x}^{(1)}$、$\boldsymbol{x}^{(2)}$和 $\boldsymbol{x}^{(3)}$ 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。

$$d(\boldsymbol{x},\boldsymbol{q}) = \sqrt{(\boldsymbol{x}-\boldsymbol{q})^{\mathrm{T}}\boldsymbol{D}^{-1}\boldsymbol{D}^{-1}(\boldsymbol{x}-\boldsymbol{q})} \tag{7}$$

其中，$\boldsymbol{D}$ 为对角方阵，对角线元素为标准差，运算如下：

$$\boldsymbol{D} = \operatorname{diag}\left(\operatorname{diag}(\boldsymbol{\Sigma})\right)^{\frac{1}{2}} = \operatorname{diag}\left(\operatorname{diag}\begin{bmatrix} \sigma_1^2 & \rho_{1,2}\sigma_1\sigma_2 & \cdots & \rho_{1,D}\sigma_1\sigma_D \\ \rho_{1,2}\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2,D}\sigma_2\sigma_D \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1,D}\sigma_1\sigma_D & \rho_{2,D}\sigma_2\sigma_D & \cdots & \sigma_D^2 \end{bmatrix}^{\frac{1}{2}}\right) = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_D \end{bmatrix} \tag{8}$$

回忆丛书之前介绍过有关 diag() 函数的说明。如果 $\boldsymbol{A}$ 为方阵，diag($\boldsymbol{A}$) 函数提取对角线元素，结果为向量；如果 $\boldsymbol{a}$ 为向量，diag($\boldsymbol{a}$) 函数将向量 $\boldsymbol{a}$ 展开成对角方阵，方阵对角线元素为 $\boldsymbol{a}$ 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：

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

(9) 可以记做：

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

其中，z_j 为：

$$z_j = \frac{x_j - q_j}{\sigma_j} \tag{11}$$

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：

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

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。

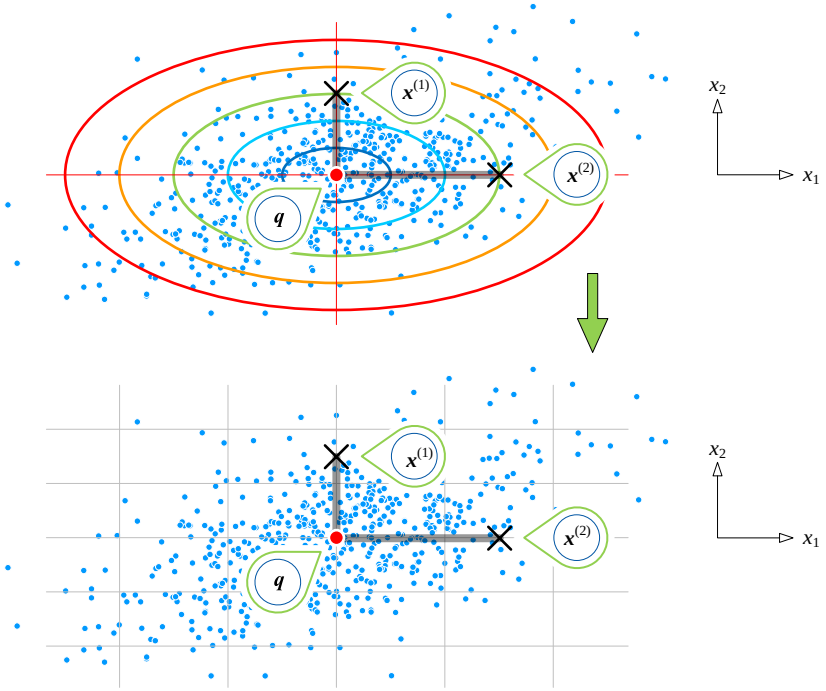


图 2. 2 特征 ($D = 2$) 标准化欧氏距离

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 $\boldsymbol{X}$ 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。

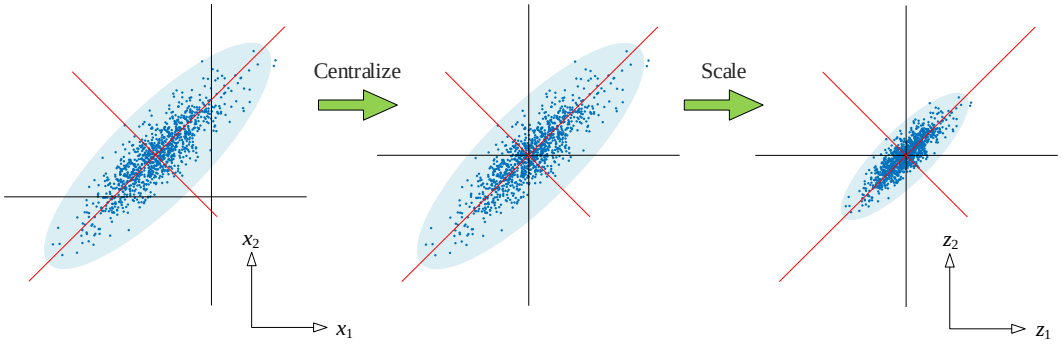


图 3. 标准化欧氏距离运算过程

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：

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

其中，$\boldsymbol{\Sigma}$ 为样本数据 $\boldsymbol{X}$ 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

$\boldsymbol{\Sigma}$ 特征值分解得到：

$$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}} \tag{14}$$

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

$$\boldsymbol{\Sigma}^{-1} = \left(\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}\right)^{-1} = \left(\boldsymbol{V}^{\mathrm{T}}\right)^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{V}^{-1} = \boldsymbol{V}\boldsymbol{\Lambda}^{-1}\boldsymbol{V}^{\mathrm{T}} \tag{15}$$

将 (15) 代入 (13) 得到：

$$d(\boldsymbol{x},\boldsymbol{q}) = \sqrt{\underbrace{\left[\underbrace{\boldsymbol{\Lambda}^{\frac{-1}{2}}}_{\text{Scale}}\underbrace{\boldsymbol{V}^{\mathrm{T}}}_{\text{Rotate}}\underbrace{\left(\boldsymbol{x}-\boldsymbol{q}\right)}_{\text{Centralize}}\right]^{\mathrm{T}}\left[\boldsymbol{\Lambda}^{\frac{-1}{2}}\boldsymbol{V}^{\mathrm{T}}\left(\boldsymbol{x}-\boldsymbol{q}\right)\right]}} \tag{16}$$

其中，$\boldsymbol{q}$ 列向量完成**中心化** (centralize)，$\boldsymbol{V}$ 矩阵完成**旋转** (rotate)，$\boldsymbol{\Lambda}$ 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。

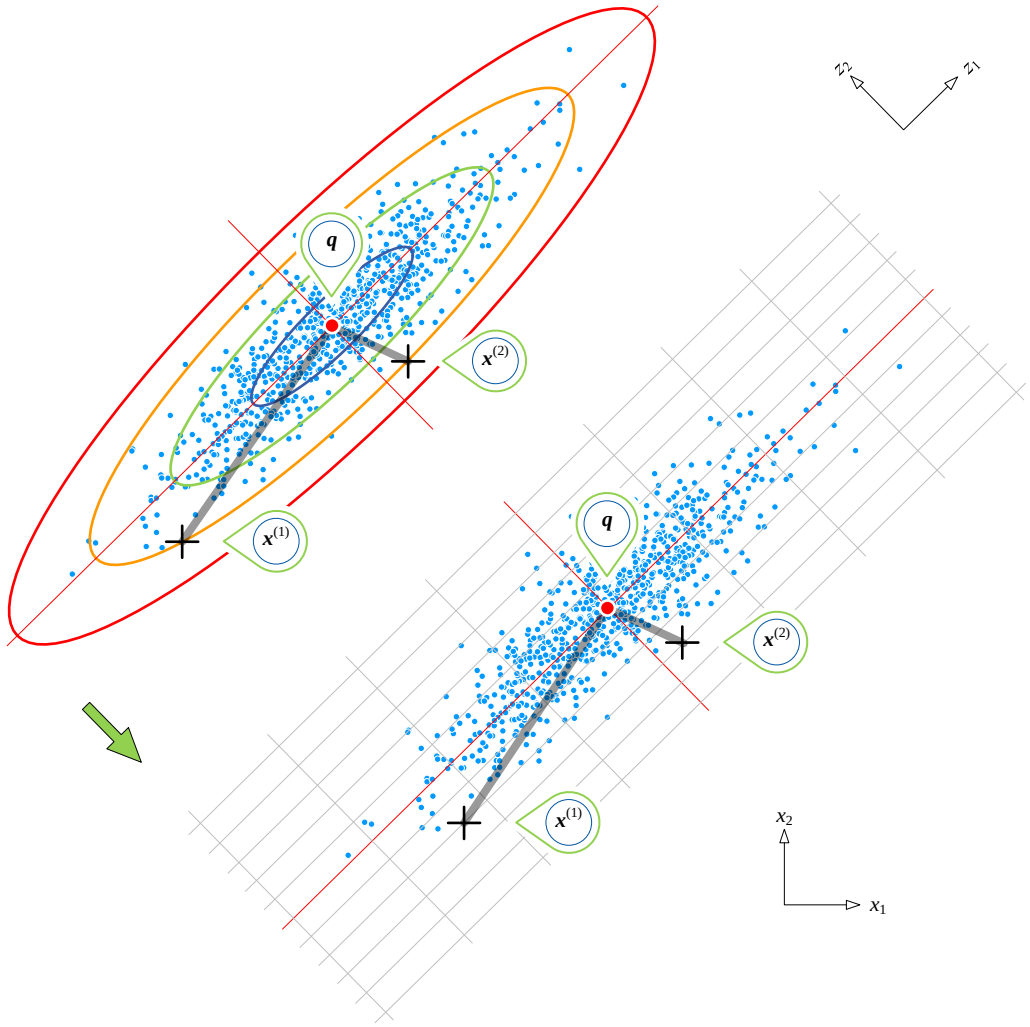
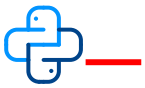


图 4. 2 特征 ($D = 2$) 马氏距离



代码 Bk7_Ch03_04.py 计算图 4 两个点的马氏距离。

下面，我们用具体数字举例讲解如何计算马氏距离。

给定质心 $\boldsymbol{\mu} = [0, 0]^T$。两个样本点的坐标分别为。

$$\boldsymbol{x}^{(1)} = \begin{bmatrix} -3.5 & -4 \end{bmatrix}^T, \quad \boldsymbol{x}^{(2)} = \begin{bmatrix} 2.75 & -1.5 \end{bmatrix}^T \tag{17}$$

计算得到 $\boldsymbol{x}^{(1)}$和 $\boldsymbol{x}^{(2)}$距离 $\boldsymbol{\mu}$ 之间欧氏距离 (L2 范数) 分别为 5.32 和 3.13。

方差协方差矩阵 $\boldsymbol{\Sigma}$ 取值如下。

$$\boldsymbol{\Sigma} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{18}$$

观察如上矩阵，可以发现 x_1 和 x_2 特征各自的方差均为 2，两者协方差为 1；计算得到 x_1 和 x_2 特征相关性为 0.5。根据 $\boldsymbol{\Sigma}$ 计算 $\boldsymbol{x}^{(1)}$和 $\boldsymbol{x}^{(2)}$距离 $\boldsymbol{\mu}$ 之间马氏距离为。

$$\begin{aligned} d_1 &= \sqrt{\left(\begin{bmatrix} -3.5 & -4 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix}\right) \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} -3.5 & -4 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix}\right)^T} \\ &= \sqrt{\begin{bmatrix} -3.5 & -4 \end{bmatrix} \cdot \frac{1}{3} \cdot \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} -3.5 & -4 \end{bmatrix}^T} = 3.08 \\ d_2 &= \sqrt{\left(\begin{bmatrix} 2.75 & -1.5 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix}\right) \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} 2.75 & -1.5 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix}\right)^T} \\ &= \sqrt{\begin{bmatrix} 2.75 & -1.5 \end{bmatrix} \cdot \frac{1}{3} \cdot \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 2.75 & -1.5 \end{bmatrix}^T} = 3.05 \end{aligned} \tag{19}$$

可以发现，$\boldsymbol{x}^{(1)}$和 $\boldsymbol{x}^{(2)}$ 和 $\boldsymbol{\mu}$ 之间马氏距离非常接近。

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：

$$d(\boldsymbol{x}, \boldsymbol{q}) = \|\boldsymbol{x} - \boldsymbol{q}\|_1 = \sum_{j=1}^{D} |x_j - q_j| \tag{20}$$

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 (20) 展开得到下式：

$$d(\boldsymbol{x},\boldsymbol{q}) = |x_1 - q_1| + |x_2 - q_2| + ... + |x_D - q_D| \tag{21}$$

特别地，当 $D = 2$ 时，城市街区距离为：

$$d(\boldsymbol{x},\boldsymbol{q}) = |x_1 - q_1| + |x_2 - q_2| \tag{22}$$

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，$\boldsymbol{x}^{(1)}$、$\boldsymbol{x}^{(2)}$和 $\boldsymbol{x}^{(3)}$ 和 $\boldsymbol{q}$ 欧氏距离均为 5，但是城市街区距离分别为 5、7 和 7。

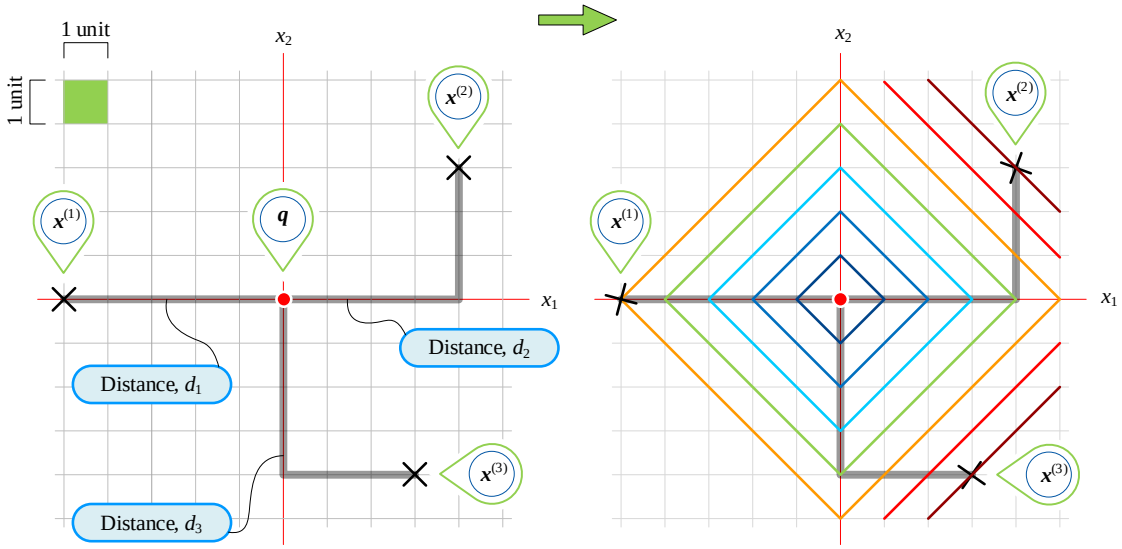


图 5. 2 特征 ($D = 2$) 城市街区距离

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^∞范数

切比雪夫距离 (Chebyshev distance)，具体如下。

$$d(\boldsymbol{x},\boldsymbol{q}) = \|\boldsymbol{x} - \boldsymbol{q}\|_\infty = \max_j \left\{ |x_j - q_j| \right\} \tag{23}$$

切比雪夫距离就是我们在《矩阵力量》第 3 章中介绍的 L^∞范数。

将 (23) 展开得到下式：

$$d(\boldsymbol{x},\boldsymbol{q}) = \max\left\{|x_1 - q_1|,\ |x_2 - q_2|,\ \ldots,\ |x_D - q_D|\right\} \tag{24}$$

特别地，当 $D = 2$ 时，切比雪夫距离为：

$$d(\boldsymbol{x},\boldsymbol{q}) = \max\left\{|x_1 - q_1|,\ |x_2 - q_2|\right\} \tag{25}$$

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，$\boldsymbol{x}^{(1)}$、$\boldsymbol{x}^{(2)}$和 $\boldsymbol{x}^{(3)}$ 和 $\boldsymbol{q}$ 欧氏距离相同，但是切比雪夫距离分别为 5、4 和 4。

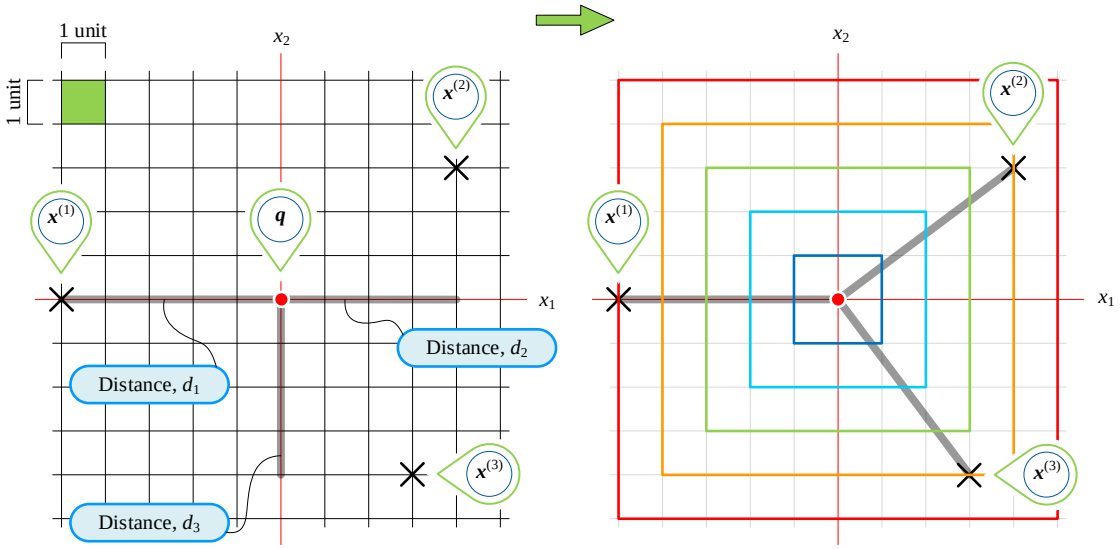


图 6. 2 特征 ($D = 2$) 切比雪夫距离

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：

$$d(\boldsymbol{x},\boldsymbol{q}) = \|\boldsymbol{x} - \boldsymbol{q}\|_p = \left(\sum_{j=1}^{D}\left|x_j - q_j\right|^p\right)^{1/p} \tag{26}$$

注意，$p \geqslant 1$。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；$p \to \infty$ 时，闵氏距离为切比雪夫距离。

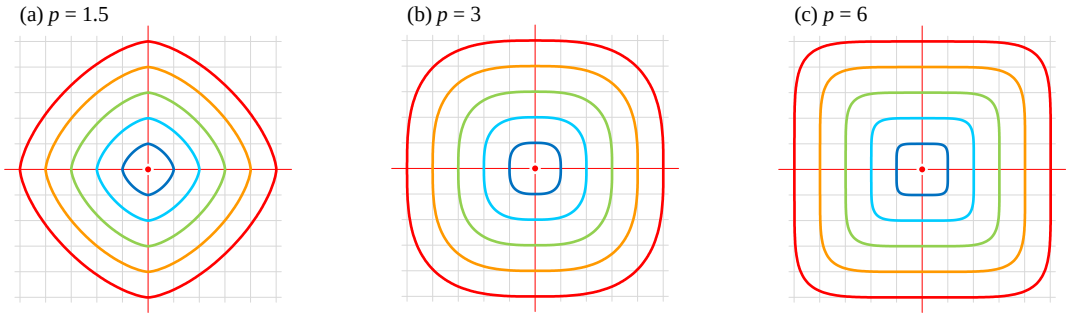


图 7. 闵氏距离 ($D = 2$)，p 取不同值

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。$\boldsymbol{x}$ 和 $\boldsymbol{q}$ 两个向量的余弦相似，具体定义如下：

$$k(\boldsymbol{x},\boldsymbol{q}) = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|} = \frac{\boldsymbol{x}\cdot\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|} \tag{27}$$

如图 8 所示，如果两个向量方向相同，则夹角 θ 余弦值 $\cos(\theta)$ 为 1；如果，两个向量方向完全相反，夹角 θ 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。

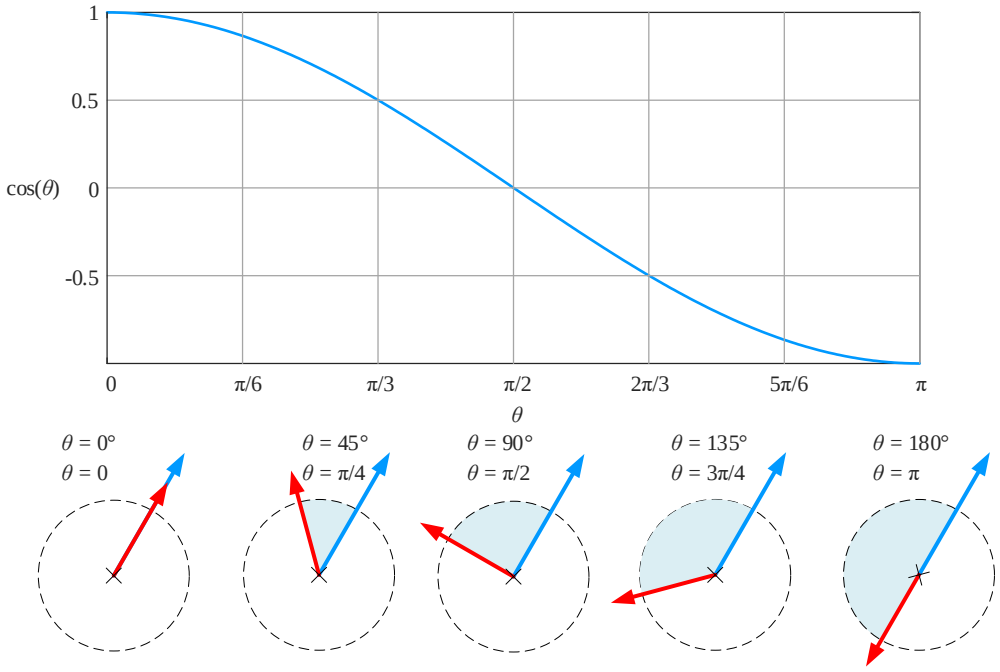


图 8. 余弦相似度

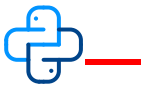
举个例子

给定如下两个向量具体值：

$$\boldsymbol{x}=\begin{bmatrix}8 & 2\end{bmatrix}^{\mathrm{T}},\quad \boldsymbol{q}=\begin{bmatrix}7 & 9\end{bmatrix}^{\mathrm{T}} \tag{28}$$

将 (28) 代入 (27) 得到：

$$k(\boldsymbol{x},\boldsymbol{q})=\frac{\boldsymbol{x}\cdot\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|}=\frac{8\times7+2\times9}{\sqrt{8^2+2^2}\times\sqrt{7^2+9^2}}=\frac{74}{\sqrt{68}\times\sqrt{130}}=0.7871 \tag{29}$$



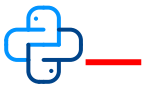
代码 Bk7_Ch03_07.py 得到和 (29) 一致结果。

余弦距离

余弦距离 (cosine distance) 的定义如下：

$$d(\boldsymbol{x},\boldsymbol{q})=1-k(\boldsymbol{x},\boldsymbol{q})=1-\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|}=1-\frac{\boldsymbol{x}\cdot\boldsymbol{q}}{\|\boldsymbol{x}\|\|\boldsymbol{q}\|} \tag{30}$$

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 (28) 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：

$$k(\boldsymbol{x},\boldsymbol{q}) = \frac{(\boldsymbol{x}-\bar{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{q}-\bar{\boldsymbol{q}})}{\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|\|\boldsymbol{q}-\bar{\boldsymbol{q}}\|} = \frac{(\boldsymbol{x}-\bar{\boldsymbol{x}})\cdot(\boldsymbol{q}-\bar{\boldsymbol{q}})}{\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|\|\boldsymbol{q}-\bar{\boldsymbol{q}}\|} \tag{31}$$

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

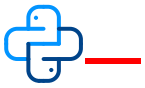
观察 (31)，发现相关系数相似度类似余弦相似度；稍有不同的是，相关系数相似度需要“中心化”向量。

还是以 (28) 为例，计算 $\boldsymbol{x}$ 和 $\boldsymbol{q}$ 两个向量的相关系数相似度。将 (28) 代入 (31) 可以得到：

$$\begin{aligned} k(\boldsymbol{x},\boldsymbol{q}) &= \frac{\left(\begin{bmatrix}8 & 2\end{bmatrix}^{\mathrm{T}} - \frac{8+2}{2}\right)\cdot\left(\begin{bmatrix}7 & 9\end{bmatrix}^{\mathrm{T}} - \frac{7+9}{2}\right)}{\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|\|\boldsymbol{q}-\bar{\boldsymbol{q}}\|} \\ &= \frac{\begin{bmatrix}3 & -3\end{bmatrix}^{\mathrm{T}}\cdot\begin{bmatrix}-1 & 1\end{bmatrix}^{\mathrm{T}}}{\left\|\begin{bmatrix}3 & -3\end{bmatrix}^{\mathrm{T}}\right\|\left\|\begin{bmatrix}-1 & 1\end{bmatrix}^{\mathrm{T}}\right\|} = \frac{-6}{6} = -1 \end{aligned} \tag{32}$$

从相关系数相似度可以计算得到相关系数距离：

$$d(\boldsymbol{x},\boldsymbol{q}) = 1 - k(\boldsymbol{x},\boldsymbol{q}) = 1 - \frac{(\boldsymbol{x}-\bar{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{q}-\bar{\boldsymbol{q}})}{\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|\|\boldsymbol{q}-\bar{\boldsymbol{q}}\|} = 1 - \frac{(\boldsymbol{x}-\bar{\boldsymbol{x}})\cdot(\boldsymbol{q}-\bar{\boldsymbol{q}})}{\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|\|\boldsymbol{q}-\bar{\boldsymbol{q}}\|} \tag{33}$$



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。

核函数亲近度

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

$$\kappa(\boldsymbol{x},\boldsymbol{q}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{q} = \boldsymbol{x}\cdot\boldsymbol{q} \tag{34}$$

对比 (27) 和 (34)，(27) 分母上 $\|\boldsymbol{x}\|$ 和 $\|\boldsymbol{q}\|$ 分别对 $\boldsymbol{x}$ 和 $\boldsymbol{q}$ 归一化。

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

将 (28) 代入 (34)，得到线性核亲近度为：

$$\kappa(\boldsymbol{x},\boldsymbol{q}) = 8\times 7 + 2\times 9 = 74 \tag{35}$$

多项式核 (linear kernel) 亲近度定义如下：

$$\kappa(\boldsymbol{x},\boldsymbol{q}) = \left(\gamma\boldsymbol{x}^{\mathrm{T}}\boldsymbol{q} + r\right)^{d} = \left(\gamma\boldsymbol{x}\cdot\boldsymbol{q} + r\right)^{d} \tag{36}$$

其中，d 为多项式核次数，γ 为系数，r 为常数。

多项式核亲近度函数为 `sklearn.metrics.pairwise.polynomial_kernel`。

Sigmoid 核 (sigmoid kernel) 亲近度定义如下：

$$\kappa(\boldsymbol{x},\boldsymbol{q}) = \tanh\left(\gamma\boldsymbol{x}^{\mathrm{T}}\boldsymbol{q} + r\right) = \tanh\left(\gamma\boldsymbol{x}\cdot\boldsymbol{q} + r\right) \tag{37}$$

Sigmoid 核亲近度函数为 `sklearn.metrics.pairwise.sigmoid_kernel`。

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

$$\kappa(\boldsymbol{x},\boldsymbol{q}) = \exp\left(-\gamma\|\boldsymbol{x}-\boldsymbol{q}\|^{2}\right) \tag{38}$$

注意到，(38) 中 $\|\boldsymbol{x}-\boldsymbol{q}\|^{2}$ 为欧氏距离的平方；因此，(38) 也可以写作：

$$\kappa(\boldsymbol{x},\boldsymbol{q}) = \exp\left(-\gamma d^{2}\right) \tag{39}$$

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

图 9 所示为，γ 取不同值时，高斯核亲近度随着欧氏距离 d 变化。聚类算法经常采用高斯核亲近度。

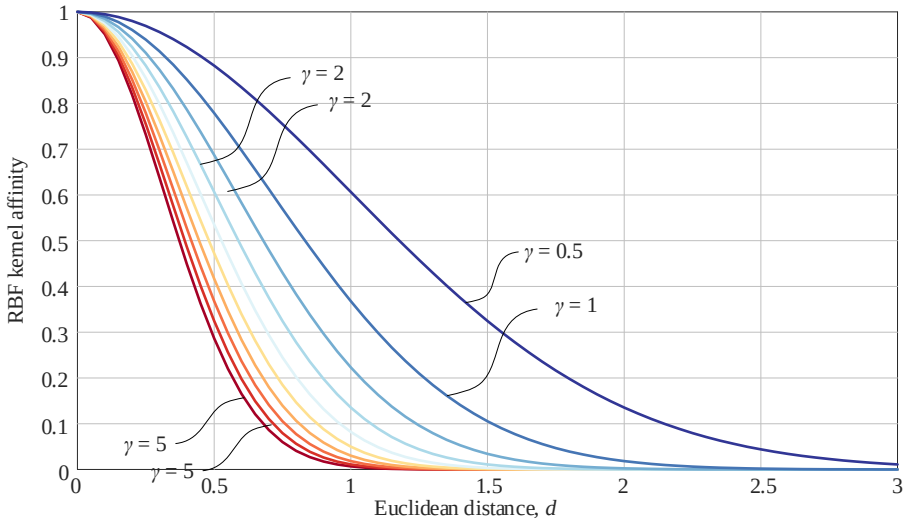


图 9. 高斯核亲近度随欧氏距离变化

拉普拉斯核 (Laplacian kernel) 亲近度，定义如下：

$$\kappa(\boldsymbol{x},\boldsymbol{q}) = \exp\left(-\gamma\left\|\boldsymbol{x}-\boldsymbol{q}\right\|_1\right) \tag{40}$$

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

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

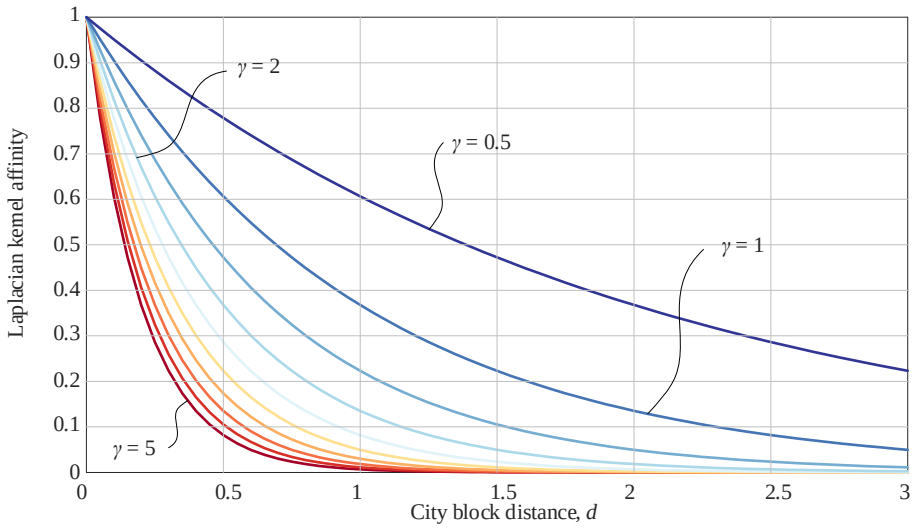


图 10. 拉普拉斯核亲近度随距离变化

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 $\boldsymbol{X}$ 每一列代表一个特征，而每一行代表一个样本数据点，比如：

$$\boldsymbol{X}_{n\times D}=\begin{bmatrix}\boldsymbol{x}^{(1)}\\ \boldsymbol{x}^{(2)}\\ \vdots\\ \boldsymbol{x}^{(n)}\end{bmatrix} \tag{41}$$

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

$$\boldsymbol{D}_{n\times n}=\begin{bmatrix}0 & d_{1,2} & d_{1,3} & \cdots & d_{1,n}\\ d_{2,1} & 0 & d_{2,3} & \cdots & d_{2,n}\\ d_{3,1} & d_{3,2} & 0 & \cdots & d_{3,n}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ d_{n,1} & d_{n,2} & d_{n,3} & \cdots & 0\end{bmatrix} \tag{42}$$

每个样本数据点和自身的距离为 0，因此 (42) 主对角线为 0。很显然矩阵 $\boldsymbol{D}$ 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。

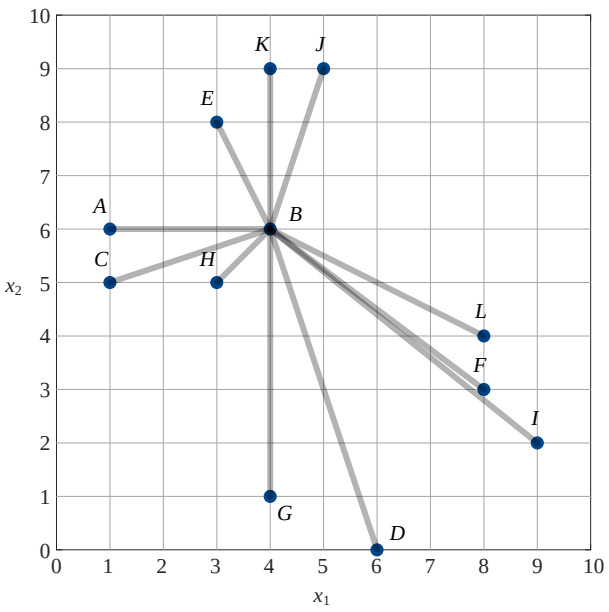


图 11. 样本数据散点图和成对距离

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：

$$C_n^2 = \frac{n(n-1)}{2} \tag{43}$$

也就是说，(42) 中不含对角线的下三角矩阵包含的信息足够使用。

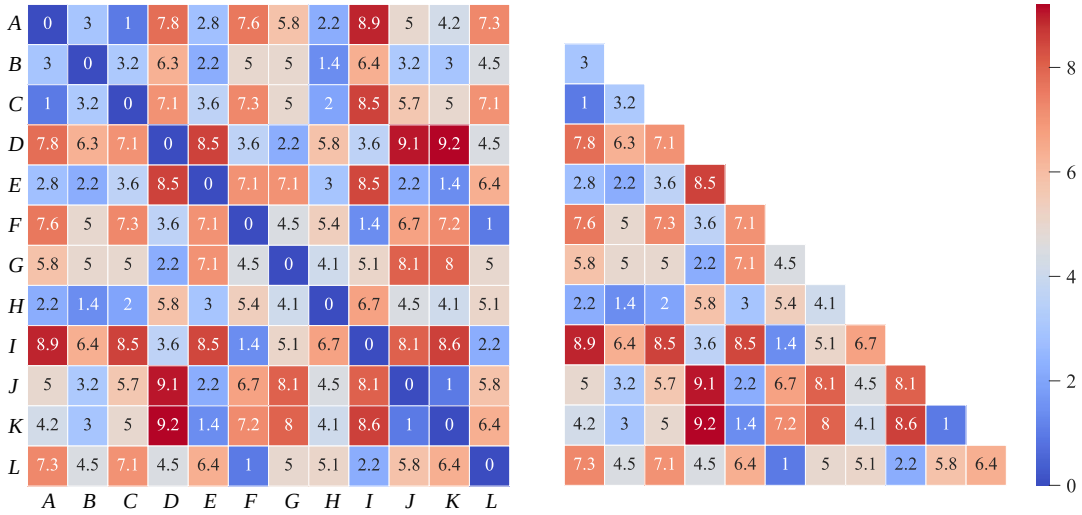


图 12. 样本数据成对距离矩阵热图

表 1 总结计算成对距离、亲近度矩阵常用函数。

表 1. 计算成对距离/亲近度矩阵常见函数

函数	描述
`metrics.pairwise.cosine_similarity()`	计算余弦相似度成对矩阵
`metrics.pairwise.cosine_distances()`	计算成对相似性距离矩阵
`metrics.pairwise.euclidean_distances()`	计算成对欧氏距离矩阵
`metrics.pairwise.laplacian_kernel()`	计算拉普拉斯核成对亲近度矩阵
`metrics.pairwise.linear_kernel()`	计算线性核成对亲近度矩阵
`metrics.pairwise.manhattan_distances()`	计算成对城市街区距离矩阵
`metrics.pairwise.polynomial_kernel()`	计算多项式核成对亲近度矩阵
`metrics.pairwise.rbf_kernel()`	计算 RBF 核成对亲近度矩阵
`metrics.pairwise.sigmoid_kernel()`	计算 sigmoid 核成对亲近度矩阵
`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.paired_cosine_distances(X,Q)`	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。

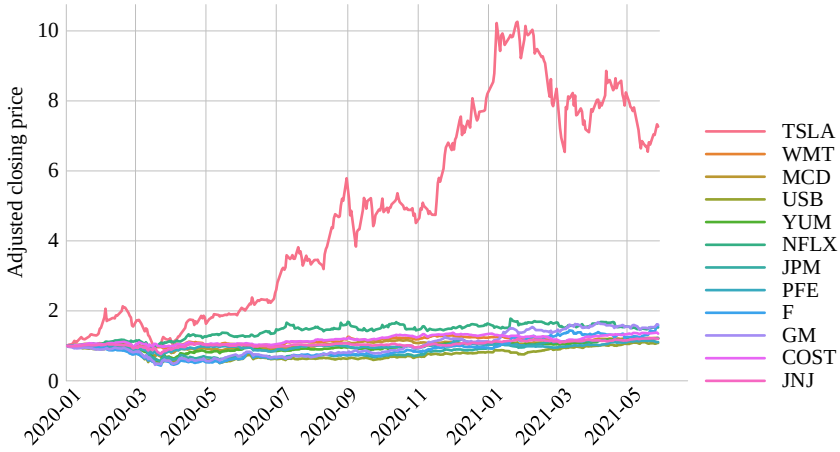


图 13. 12 只股票股价水平，初始股价归一化

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。

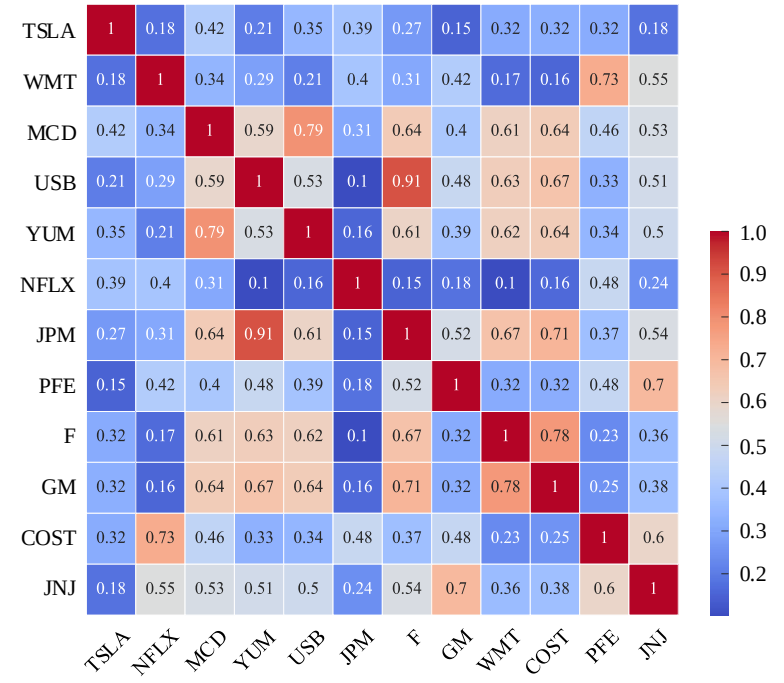


图 14. 12 只股票相关性热图

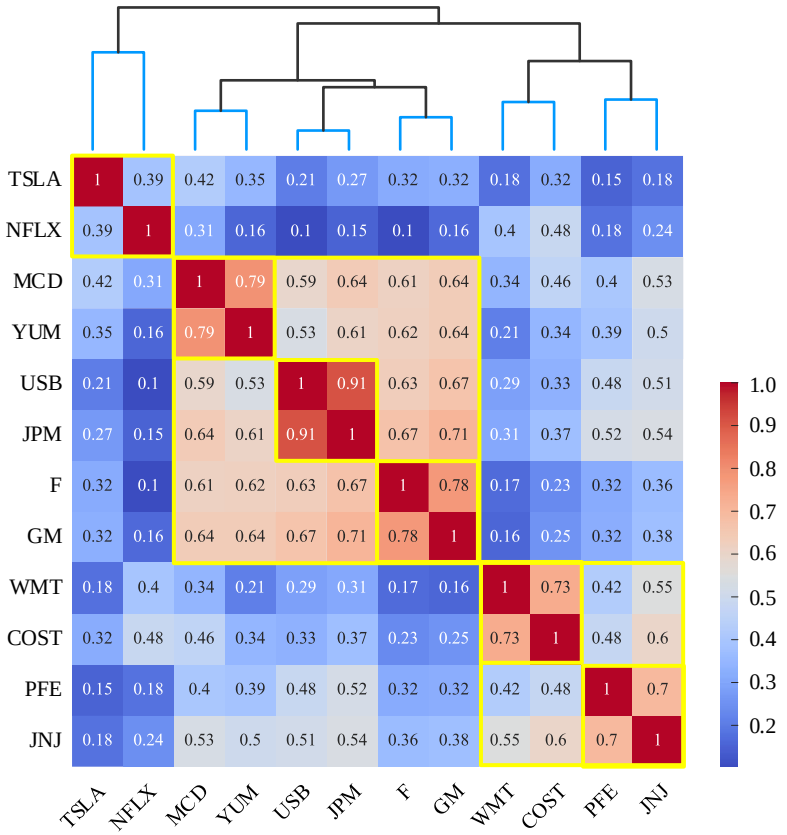
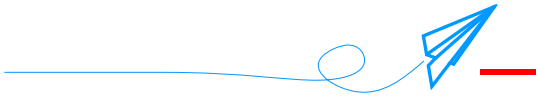


图 15. 根据树形图重组相关性热图

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。