《机器学习》课程系列

k 均值聚类*

k-Means Clustering

武汉纺织大学数学与计算机学院

杜小勤

2020/10/03

Contents

*本系列文档属于讲义性质，仅用于学习目的。Last updated on: November 4, 2020。

1 无监督学习

在机器学习算法中，有一类算法，可以从无标注 (Labelling) 或无标签 (Label) 的数据集中，学习到数据集的统计规律与内在结构特性，我们将这类学习算法称为无监督学习 (Unsupervised Learning)，它是机器学习的重要组成部分。与之对应的是监督学习 (Supervised Learning)，它在学习时需要带标签的数据集。

还有一类学习算法，被称为强化学习。虽然它在学习时也不需要带标签的数据，但是该算法强调的是在与环境的交互中进行学习——需要环境提供有效的反馈信息或强化信号，告知学习智能体的哪些行为受欢迎 (反馈一个正奖励信号)，哪些行为不受欢迎 (反馈一个负奖励信号)。正奖励信号与负奖励信号统称为强化信号或回报。智能体在与环境的交互中，不断地收集这些回报，并在眼前利益与长远利益之间取得某种平衡[1]，以构建出适应环境的“最佳”行为策略。

无监督学习中的基本问题，包括聚类、降维、话题分析以及图分析等，所使用的技术手段通常有聚类、降维与概率估计等。无监督学习可以用作数据分析或者监督学习的预处理阶段。

与监督学习一样，无监督学习也包括三要素，即模型、策略与算法。

在不同的无监督学习任务中，模型表现为不同的形式。例如，聚类模型表现为类别映射函数，它将输入映射到类别上，或者换句话说，按照某种相似度或距离准则将输入数据划分成若干类别，即对输入数据执行通常意义上的“物以类聚，人以群分”操作。对于降维模型，它将高维输入向量，降维为低维向量，使得向量的表示更加紧凑。在基于概率模型的无监督学习中，可以使用混合概率模型，例如高斯混合模型 (Gaussian Mixture Model, GMM)，也可以使用概率图模型，例如有向概率图模型和无向概率图模型。

策略一般表现为某种优化函数或损失函数的最优化。例如，在聚类问题中，将

[1]该问题的正式术语为“利用与探索”(Exploitation-Exploration) 的平衡问题，这是一个两难选择问题，需要利用各种策略进行平衡。从某种意义而言，强化学习领域的主要研究任务就是处理利用与探索之间的平衡。

所有样本与其所属类别的聚类中心之间的总距离定义为损失函数，然后对其执行最小化求解，则可以得到“最优”模型。在降维任务中，将所有样本从高维空间降维到低维空间中产生的总信息损失定义为损失函数。而在概率模型中，可以将所有样本的生成数据概率的似然函数定义为优化函数，然后执行最优化求解[2]。

算法一般表现为使用各种优化迭代方法对优化函数进行优化求解。常见的方法包括梯度下降法、牛顿法、拟牛顿法等。

2 相似度准则

聚类 (Clustering) 指的是，按照某个特定标准，例如距离或相似度准则，将一个数据集分割成不同的类 (Class) 或簇 (Cluster)，使得同一个簇内的数据对象之间的相似性尽可能地大，而簇间数据对象之间的相似性尽可能地小，即不相似性尽可能地大。这意味着，聚类后同类数据对象尽可能地聚集到一起，不同类数据对象尽量地相互分离。

如图2-1所示，展示了一个 2 维数据集的 k 均值聚类结果，相似性准则为欧氏距离。在该图中，标记为红 (实心点)、绿 (+ 标记点) 与蓝 (Y 标记点) 的数据样本点，依据欧氏距离的相似性准则，分别形成了 3 个簇：每个簇内，样本点因相对距离近而形成一簇；每个簇间，样本点因相对距离较远而相互分离形成不同的簇。一般而言，位于不同簇内的样本点，其性质是不同的，而簇内的样本点，往往具有相似的性质。

从上述示例可以看出，借助于聚类的结果，可以为每个数据样本点赋予簇或类别标签。例如，分别为红色实心样本点、绿色“+”标记样本点、蓝色“Y”标记样本点赋予标签“0”、“1”和“2”。因此，我们能够将聚类算法应用于监督学习的数据预处理与准备阶段，以生成供监督学习算法使用的带标签的数据集。

聚类的方法有很多，包括层次化聚类、k 均值聚类、高斯混合模型、基于密度的方法 (例如 DBScan 等)、基于谱聚类的方法 (例如 Normalized Cuts 等) 等方法。本章讨论 k 均值聚类方法。

对于聚类等方法而言，数据点以及簇之间的相似度是一个非常重要的研究论

[2]一般情况下，将似然函数转换为负对数似然函数 (即损失函数)，从而将最大化问题转换为损失函数的最小化问题。实际上，取对数的意义在于，将累乘项转换为累加项，可以避免许多极小量的累乘所造成的精度损失，也有利于使用解析方法对问题进行求解。

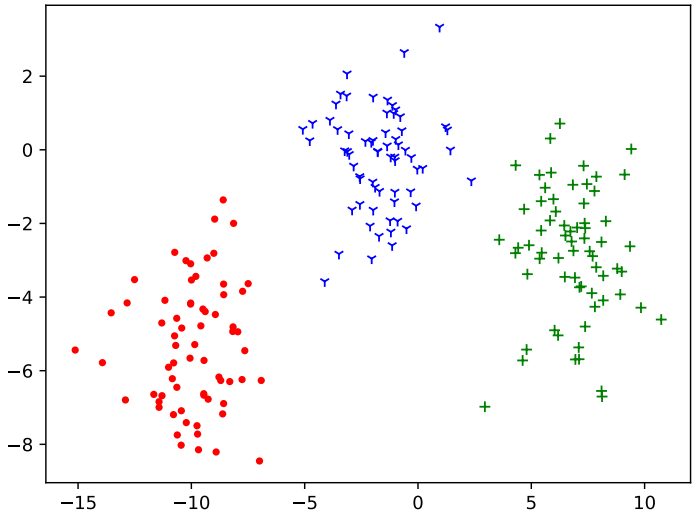


图 2-1: 一个 2 维数据集的 k 均值聚类结果

题，下面将分别予以介绍。

2.1 数据点之间的相似度

从前面的讨论可以看出，在聚类算法中，数据点以及簇之间的相似度度量准则是一个非常重要的概念，它将直接影响到聚类的效果。从某个角度来说，聚类性能的好坏是由相似度度量准则决定的，而相似度的选取取决于问题的性质。

下面，首先概述数据点之间的各种相似度准则，然后列举各种簇间相似度度量准则。在本节，我们将对欧氏距离与马氏距离展开比较研究，以展示马氏距离的特性。

2.1.1 Minkowski 距离

Minkowski 距离 (Minkowski Distance) 或闵氏距离，又被称为 L_p 距离 (L_p Distance)，它比欧氏距离更加具有一般性。

在闵氏距离准则下，将数据点看作向量空间中的向量 (点)，且以向量空间中的距离表示数据点之间的相似度——闵氏距离越小，相似度越大；反之，闵氏距离越大，相似度越小。

设特征空间 $\mathcal{X}$ 是 m 维实数向量空间 $\mathbf{R}^m$，$\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 为该空间中的 2 个实例

点，其 L_p 距离被定义为：

$$L_p(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{l=1}^{m} |x_i^{(l)} - x_j^{(l)}|^p \right)^{\frac{1}{p}} \tag{1}$$

其中，m 表示实例点 $\boldsymbol{x}$ 的维度，$p \geqslant 1$。具体地，常用的距离准则有：

- $p = 1$：

$$L_1(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{l=1}^{m} |x_i^{(l)} - x_j^{(l)}| \tag{2}$$

  被称为曼哈顿距离 (Manhattan Distance)。

- $p = 2$：

$$L_2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{l=1}^{m} |x_i^{(l)} - x_j^{(l)}|^2 \right)^{\frac{1}{2}} \tag{3}$$

  被称为欧氏距离 (Euclidean Distance)。

- $p = +\infty$：

$$L_{+\infty}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max_l |x_i^{(l)} - x_j^{(l)}| \tag{4}$$

  被称为切比雪夫距离 (Chebyshev Distance)。上式表明，该度量准则将坐标距离中的最大值作为距离。

图2-2展示了上述三种 p 值所对应的图形：二维空间中，$L_p(\boldsymbol{x}, \boldsymbol{0}) = 1$ 的点 $\boldsymbol{x}$ 所组成的图形。

从图2-2可以看出，欧氏距离 ($p = 2$) 具有各向同性 (Isotropism) 或均质性 (Homogeneity) 的特点，即它在所有方向上具有相同的度量性质，或者说，它不具有方向性。因此，欧氏距离准则适用于数据点均匀分布的情形。

如图2-3所示，展示了一个簇及其数据点的分布状况，虚圆线内的所有红色实心点属于该簇。从图中可以看出，该簇内的数据点在各个方向上的分布较为均匀。在簇外，存在 2 个蓝色实心点 p_1 与 p_2，它们到簇中心 c(绿色实心点) 的距离相同。显然，我们可以依据某个设定的条件，将点 p_1 与点 p_2 同时赋予该簇或不赋予该簇，即点 p_1 与点 p_2 应该具有相似的性质。之所以能够做出上述判断，主要原因在于点 p_1 与点 p_2 到簇中心 c 的距离相等，且该簇的数据分布具有均质性。

然而，如果簇的数据分布不具有均质性，那么我们还能否做出上述判断呢？如图2-4所示，展示了一个簇，其数据分布具有明显的偏向性。此时，点 p_1 与点 p_2

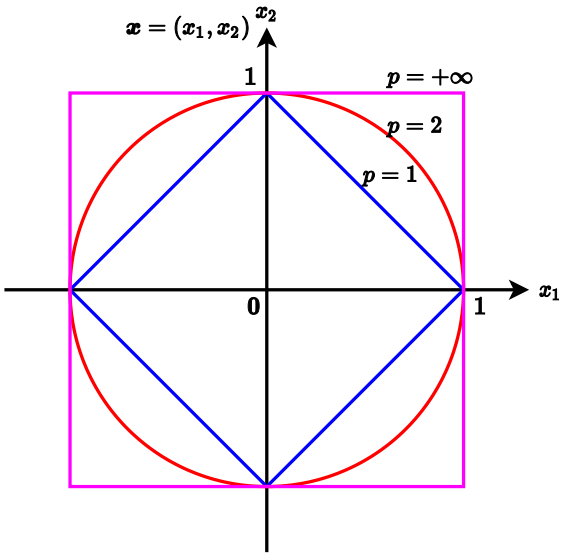


图 2-2: $L_p(\boldsymbol{x},\mathbf{0})=1$ 的点 $\boldsymbol{x}$ 所组成的图形：$p=1$、$p=2$、$p=+\infty$

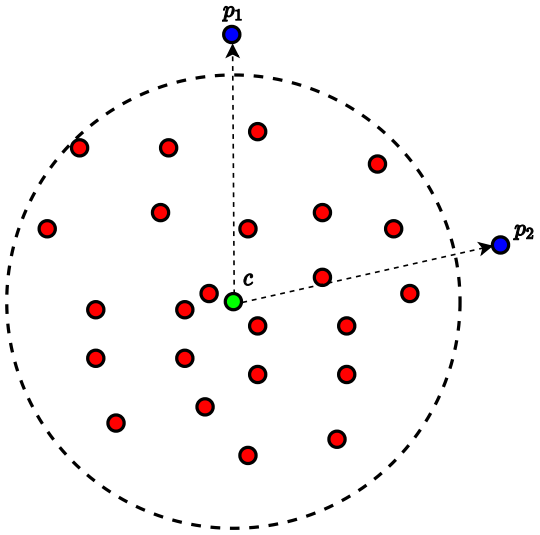


图 2-3: 均质性的数据分布

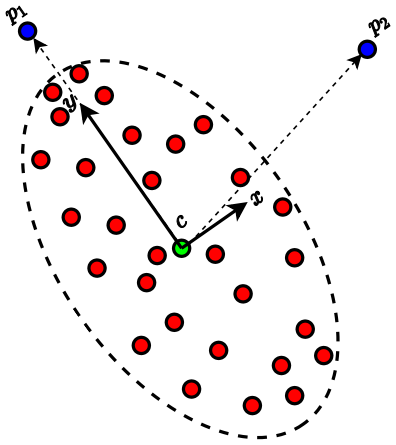


图 2-4: 非均质性的数据分布

到簇中心 c 的距离仍然相等。但是，从图中可以看出，点 p_1 位于数据分布的主轴 y 的方向上，而点 p_2 位于数据分布的次轴 x 的方向上——虽然点 p_1 与点 p_2 到簇中心 c 的距离相等，但是点 p_1 更靠近簇内的数据点，而点 p_2 更远离簇内的数据点。因此，我们有理由认为，点 p_1 与点 p_2 应该具有不同的性质，或者说，点 p_1 与点 p_2 应该属于不同的簇。至少，点 p_1 属于该簇的可能性要高于点 p_2 属于该簇的可能性。

综上所述，欧氏距离不能正确地处理非均质性的数据分布情形，其相似度度量准则对于此类情况是不准确的。为此，我们需要结合数据的分布特征，提出一种更为有效的距离度量准则。Mahalanobis 距离能够解决上述问题。

2.1.2 Mahalanobis 距离

Mahalanobis 距离 (Mahalanobis Distance，简称马氏距离)，也是一种常用的相似度度量准则，由 P. C. Mahalanobis 于 1936 年提出。它显式地考虑了数据集中各数据点及其分量之间的相关性，并消除了各分量之间尺度的不一致性，因而能够很好地处理非均质数据分布的相似度度量问题。

给定一个数据集 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$，其中 $\boldsymbol{x}_i \in \mathbf{R}^m$ 为 m 维向量，数据集

的协方差矩阵记作 $\boldsymbol{\Sigma}$，则数据点 $\boldsymbol{x}_i$ 与数据点 $\boldsymbol{x}_j$ 之间的马氏距离被定义为：

$$d_{ij} = \left((\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j) \right)^{\frac{1}{2}} \tag{5}$$

其中：

$$\boldsymbol{x}_i = \left(x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(m)} \right)^T, \boldsymbol{x}_j = \left(x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(m)} \right)^T \tag{6}$$

显然，当协方差逆矩阵 $\boldsymbol{\Sigma}^{-1}$ 退化为单位矩阵 $\boldsymbol{I}$ 时，马氏距离就退化为欧氏距离。此时，在数据集的各个方向上，数据点的分布具有均质性，各分量相互独立且各分量的方差为 1。

对于一般的协方差逆矩阵 $\boldsymbol{\Sigma}^{-1}$，我们自然会产生这样的疑问：“既然欧氏距离是马氏距离的特例，那么马氏距离应该具有欧氏距离所不具备的特性，或者马氏距离能够解决欧氏距离所不能解决的问题，或者马氏距离所表达的几何意义是什么?”。针对上述疑问，我们仍然从图2-4所展示的示例出发，展开讨论。

为了弄清楚协方差逆矩阵 $\boldsymbol{\Sigma}^{-1}$ 所表达的含义，我们考虑对其进行对角化处理。由于 $\boldsymbol{\Sigma}^{-1}$ 为实对称矩阵，则必有正交矩阵 $\boldsymbol{P}$，使得下式成立[3]：

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{P} \boldsymbol{\Lambda} \boldsymbol{P}^T \tag{7}$$

其中，$\boldsymbol{\Lambda}$ 为协方差逆矩阵 $\boldsymbol{\Sigma}^{-1}$ 的特征值组成的对角矩阵，$\boldsymbol{P}$ 为相应的特征列向量组成的正交矩阵，且 $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$ 成立。上式即为协方差逆矩阵 $\boldsymbol{\Sigma}^{-1}$ 的对角化公式。

将公式 (7) 代入马氏距离计算公式 (5)，令 $\boldsymbol{x}_j = \boldsymbol{c}$，并利用公式 $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{\Lambda}^{\frac{1}{2}}$ 以及 $\boldsymbol{\Lambda}^{\frac{1}{2}}$ 为对角矩阵的事实，可以得到：

$$\begin{aligned} d_{ic} &= \left((\boldsymbol{x}_i - \boldsymbol{c})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{c}) \right)^{\frac{1}{2}} \\ &= \left((\boldsymbol{x}_i - \boldsymbol{c})^T \boldsymbol{P} \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{P}^T (\boldsymbol{x}_i - \boldsymbol{c}) \right)^{\frac{1}{2}} \\ &= \left(\left(\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{P}^T (\boldsymbol{x}_i - \boldsymbol{c}) \right)^T \left(\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{P}^T (\boldsymbol{x}_i - \boldsymbol{c}) \right) \right)^{\frac{1}{2}} \end{aligned} \tag{8}$$

其中，$\boldsymbol{c}$ 为数据集的中心。考察向量 $\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{P}^T (\boldsymbol{x}_i - \boldsymbol{c})$，其意义非常明确[4]——首先对向量 $\boldsymbol{x}_i$ 执行中心化操作，即 $(\boldsymbol{x}_i - \boldsymbol{c})$；然后应用正交矩阵 $\boldsymbol{P}$ 的逆变换 (因为 $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$)，对中心化后的向量 $(\boldsymbol{x}_i - \boldsymbol{c})$ 执行旋转操作，使之与标准的坐标轴重合，变换后的各分量是相互独立的；最后对每个坐标分量执行自适应的缩放操作，

[3]相关讨论，请阅读《机器学习》课程系列之“高斯分布”。

[4]相关讨论，请阅读《机器学习》课程系列之“高斯分布”。

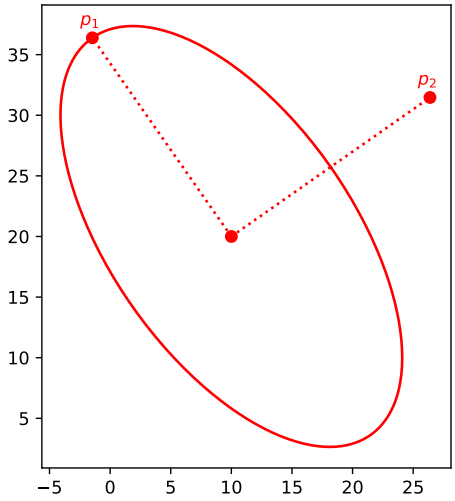


图 2-5: 变换前的图形

按比例消除各坐标分量的缩放因子长短的影响，使得所有分量的长度都一致。经过上述变换之后，数据点的分布将由非均质形态转变为均质形态。

我们可以使用附录 (6.1) 的程序对上述变换的结果进行验证与展示。图2-5展示了变换前的数据分布轮廓。为简便及演示效果的考虑，我们将数据点直接分布在椭圆曲线上，并且也将 p_1 置于椭圆曲线上，而 p_2 为椭圆外的一点，且 p_1 到椭圆中心的距离与 p_2 到椭圆中心的距离相等。

图2-6展示了执行变换 $\boldsymbol{P}^T(\boldsymbol{x}_i - \boldsymbol{c})$ 后的结果。注意到，此时椭圆的中心位于 $(0,0)$ 处，且长短轴已与坐标轴对齐。但是，椭圆的形状并没有发生变化。这意味着，p_1 到椭圆中心的距离与 p_2 到椭圆中心的距离仍然相等。

图2-7展示了执行变换 $\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{P}^T(\boldsymbol{x}_i - \boldsymbol{c})$ 后的结果。此时，数据分布的轮廓已经由椭圆变为了标准圆形。这表明，p_1 到椭圆中心的距离与 p_2 到椭圆中心的距离不再相等，且前者小于后者，即椭圆长轴上的距离实际上缩小了，而椭圆短轴上的距离却放大了。这种缩放效果正是我们所需要的——马氏距离准则综合考虑了数据集的分布状况，在欧氏距离准则下原本看似相似的点 p_2 与点 p_1，在马氏距离准则下显露了“原形”，它们不再相似！实际情况也的确如此，与 p_1 相比，p_2 并不更接近数据集。

图2-8展示了某个数据集的马氏距离变换效果。

从上述讨论可以看出，马氏距离是度量数据集中数据点之间距离的一种有效方法，它将数据集的数据分布特征考虑在内，能够有效地解决欧氏距离准则中存

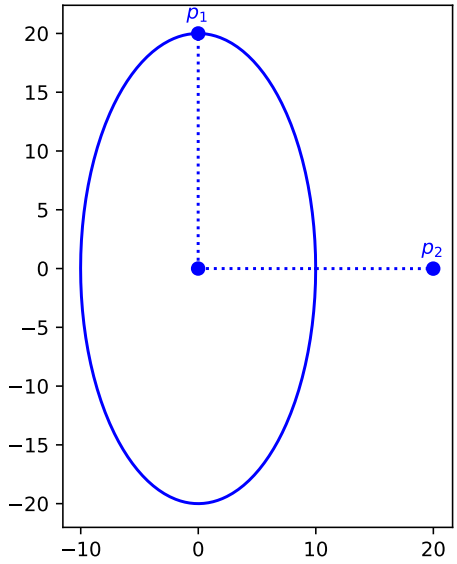


图 2-6: 执行变换 $\boldsymbol{P}^T(\boldsymbol{x}_i - \boldsymbol{c})$ 后的图形

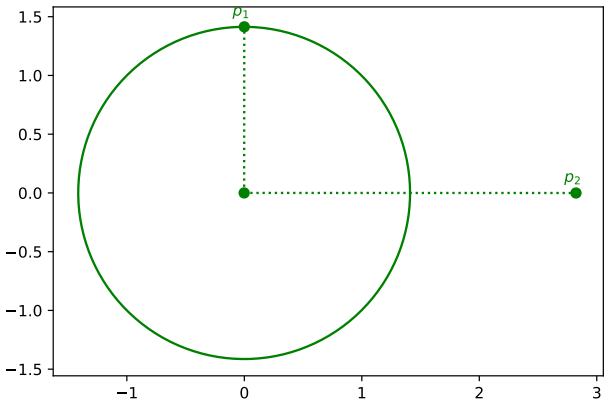
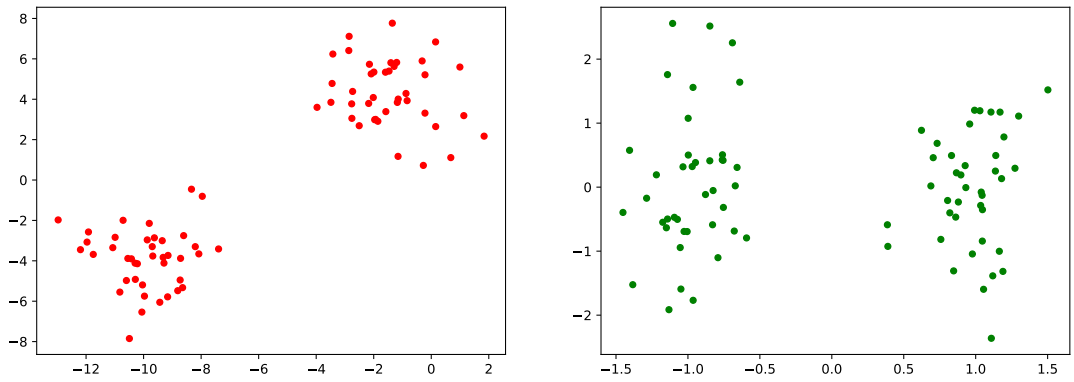


图 2-7: 执行变换 $\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{P}^T(\boldsymbol{x}_i - \boldsymbol{c})$ 后的图形



(a) 变换前 (b) 变换后

图 2-8: 某个数据集的马氏距离变换效果

在的问题。正因为如此，马氏距离也能够用于异常检测 (Anomaly Detection) 应用以及类别间样本点数目非常不均衡的数据集的分类任务中，详情请参考文献 [6]。

我们也可以从权重的角度来解释马氏距离与欧氏距离，以加深对这 2 种距离准则的理解。仔细观察图2-7，易知，在马氏距离准则下，原本在欧氏距离准则下相等的距离，因被赋予不同的权重，而不再相等。实际上，这些权重就组成了协方差逆矩阵 $\boldsymbol{\Sigma}^{-1}$——在数据集的各分量与标准坐标轴对齐之后，主要分量被赋予的权重变小，其效果为距离被缩小，而次要分量被赋予的权重变大，其效果为距离被放大。反过来，在欧氏距离准则下，无论数据分布的分量主次或重要性如何，分量被赋予的权重完全一样，它们组成了单位矩阵 $\boldsymbol{I}$。因此，欧氏距离准则适用于分量具有同等重要性且相互独立的场合，或者，在满足该条件的场合中，欧氏距离准则才具有较为完美的表现。

需要注意的是，在马氏距离 (公式 (5)) 中，如果数据点为标量形式，且用数据集均值 μ 替换 x_j、方差 σ^2 替换协方差矩阵 $\boldsymbol{\Sigma}$，则该公式的标量形式为：

$$d_i = \frac{x_i - \mu}{\sigma} \tag{9}$$

其中，均值 μ 和标准差 σ 分别由下面的公式计算 (n 为数据集中样本点的个数)：

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i,\ \sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{10}$$

公式 (9) 被称为标准化 (Standardization)、Z-Score 归一化 (Z-Score Normalization) 或 Standard Score 计算公式。其作用是，将数据集转换为均值 0、方差 1 且不受

表 2-1: 使用不同单位的相同数据

Area(sq.ft)	Price($1000's)	Area(acre)	Price($M)
2400	156000	0.0550944	156
1950	126750	0.0447642	126.75
2100	105000	0.0482076	105
1200	78000	0.0275472	78
2000	130000	0.045912	130
900	54000	0.0206604	54

表 2-2: 示例数据的标准化结果

Area(sq.ft)	Price($1000's)	Area(acre)	Price($M)
1.21550331	1.40035732	1.21550331	1.40035732
0.36307242	0.54179763	0.36307242	0.54179763
0.64721605	-0.09661854	0.64721605	-0.09661854
-1.05764574	-0.88913517	-1.05764574	-0.88913517
0.45778696	0.63719315	0.45778696	0.63719315
-1.625933	-1.5935944	-1.625933	-1.5935944

量纲影响的数据集。与其它几种特征缩放的预处理方法相比[5]，该方法适用于数据点的最大值和最小值未知的情况，或有超出取值范围的离群数据的情况。该方法可以消除因量纲不同带来的特征数据范围的差异所产生的影响。

例如，在表2-1中，每一行列出了同一房屋的面积与价格，左边 2 列与右边 2 列的差别仅仅在于所使用单位的不同。可以看出，左边 2 列数据的大小值要远远高于右边 2 列的数据。此外，仅单纯地从每列数据的变化范围上看，左边 2 列数据的变化范围也要比右边 2 列数据的变化范围大很多。因此，可以预见，如果不对这些数据进行预处理，那么分别使用它们进行学习或训练，将会获得截然不同的结果。这显然是不合理的，也是不利于学习与训练的。

如果对表中数据进行标准化处理，结果则如表2-2所示。此时，左右 2 列数据完全一致，显然符合“对同一对象进行描述应该得到同样的结果”这一要求。

[5]在一些文献中，可能由于中文翻译等原因，存在着“标准化 (Standardization)”与“归一化 (Normalization)”等术语滥用的情形。无论使用哪个术语，对于数据集的预处理而言，它们都属于特征缩放 (Feature Scaling) 方法。其它常见的特征缩放方法包括：Min-Max 归一化 (Min-Max Normalization)，也被称为 0-1 归一化 (0-1 Normalization) 或线性函数归一化或离差归一化；均值归一化 (Mean Normalization)；Sigmoid/Softmax 归一化等。各种特征缩放方法的作用也不尽相同。但是，总体上来说，特征缩放的作用表现为：消除特征的数据范围不同所造成的影响；加快梯度下降算法的收敛速度。关于特征缩放的具体解释，详见文献 [7]-[11]。

从距离计算的角度分析表2-1和表2-2，也可以得出类似的结论。

直觉上，标准化公式 (9) 的意义在于，它将数据点 x_i 偏离均值 μ 的偏差 $x_i-\mu$ 转换为标准偏差 σ 的倍数，即 x_i 偏离均值 μ “多少个”标准偏差，从而将绝对量转变为相对量，因而消除了单位量纲的影响[6]。这使得不同单位量纲的变量之间具有可比性，为它们之间关系的研究创造了有利条件。例如，通过计算标准化后的面积 (Area) 与价格 (Price) 这 2 个变量之间的协方差，可以得到它们之间准确的相关性关系。

2.1.3 夹角余弦

2 个数据点 (向量) 之间的相似度，也可以使用它们的夹角余弦 (Cosine) 来表示[7]：夹角越小，余弦值越大，数据点越相似；反之，夹角越大，余弦值越小，数据点越不相似。其定义为：

$$\cos\theta = \frac{\boldsymbol{x}_i^T\boldsymbol{x}_j}{\|\boldsymbol{x}_i\|\|\boldsymbol{x}_j\|} = \frac{\boldsymbol{x}_i^T}{\|\boldsymbol{x}_i\|}\frac{\boldsymbol{x}_j}{\|\boldsymbol{x}_j\|} \tag{11}$$

其中，θ 为向量 $\boldsymbol{x}_i$ 与向量 $\boldsymbol{x}_j$ 之间的夹角；$\|\boldsymbol{x}\|$ 表示向量 $\boldsymbol{x}$ 的模长；$\boldsymbol{x}_i^T\boldsymbol{x}_j$ 表示向量 $\boldsymbol{x}_i$ 与向量 $\boldsymbol{x}_j$ 之间的点积；$\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$ 所表达的意义为向量 $\boldsymbol{x}$ 的单位化，这相当于消除其长度的影响。

从夹角余弦的计算公式可知，2 个向量点的相似度并不取决于它们之间的距离，也不取决于它们自身的模长，而是取决于它们之间的夹角。图2-9展示了两向量之间的夹角余弦相似度示例。在该示例中，如果将欧氏距离作为相似度准则，则点 P_2 较之点 P_3 与点 P_1 更相似；而如果将夹角余弦作为相似度准则，则点 P_3 较之点 P_2 与点 P_1 更相似。因此，相似度准则的选取是非常重要的。

夹角余弦相似度准则可以消除向量自身模长带来的影响，这对于诸如长度差异较大的文本主题相似度判断的应用而言是非常有优势的。例如，在信息检索中，每个词项被赋予不同的维度，而一个文档由一个向量表示，其各个维度上的值对应于该词项在文档中出现的频率。如果单纯地使用文本中各个常用词的词频，并将欧氏距离作为文本相似度的判断依据，那么 2 个较长的文本之间的欧氏距离可

[6]标准化后的随机变量，均值为 0，方差为 1。

[7]在计算机图形学领域，两向量的夹角余弦经常被用作判断它们之间方向异同的依据以及由此产生的各种应用 (例如三角片的正面与背面的判断、迎光面与背光面的判断等)——如果夹角余弦大于 0，则两向量方向大致相同；如果夹角余弦等于 0，则两向量垂直；如果夹角余弦小于 0，则两向量方向相反。

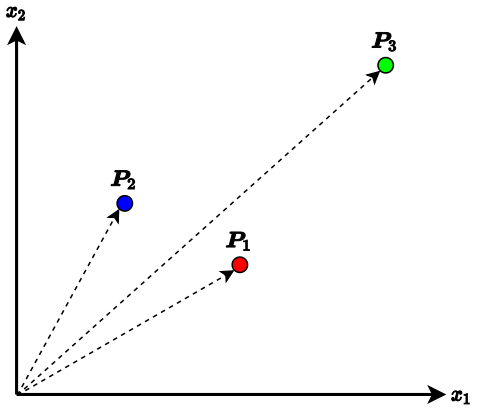


图 2-9: 两向量之间的夹角余弦相似度

能更接近 (常用词的词频更接近)，而 2 个长度差异较大的文本，即使它们的主题非常接近，其欧氏距离可能会更大 (常用词的词频差异大)。但是，如果使用夹角余弦作为文本的相似度判断依据，则 2 个长度差异较大但主题更接近的文本更加相似——夹角余弦将选择忽略掉常用词的词频差异大这一因素，可以给出两篇文档在主题方面的相似度。因此，可以将夹角余弦公式看作文本长度的规整化方法。在信息检索的应用中，由于词频不为负数，因而两个文档的余弦相似性的范围为 $[0, 1]$。

2.1.4 相关系数

相关系数 (Correlation Coefficient) 是一种用于度量随机变量之间某种类型的相关性的度量准则。相关系数的取值范围为 $[-1, +1]$：值为 ± 1 时，表示 2 个变量之间存在强相关性；值为 0 时，表示 2 个变量之间不具有相关性。

针对不同的应用，存在几种相关系数，例如：皮尔逊相关系数 (Pearson Correlation Coefficient，PCC)，也被称为皮尔逊乘积矩相关系数 (Pearson Product-moment Correlation Coefficient，PPMCC) 或 2 元变量相关性 (Bivariate Correlation)，记为 r 或 R；类内相关性 (Intraclass Correlation, ICC) 等。皮尔逊相关系数用于度量 2 个变量之间的线性相关性，是最著名而常用的一种相关系数。一般情况下，如果没有特别说明，相关系数指的就是皮尔逊相关系数。在不引起歧义的情况下，将其简称为相关系数，下面将对其进行详细介绍。

在给出皮尔逊相关系数的正式定义之前，我们先来直观地感受一下其所表达的含义。对于皮尔逊相关系数，值为 ± 1 时，表示 2 个变量完全线性相关 (即呈现

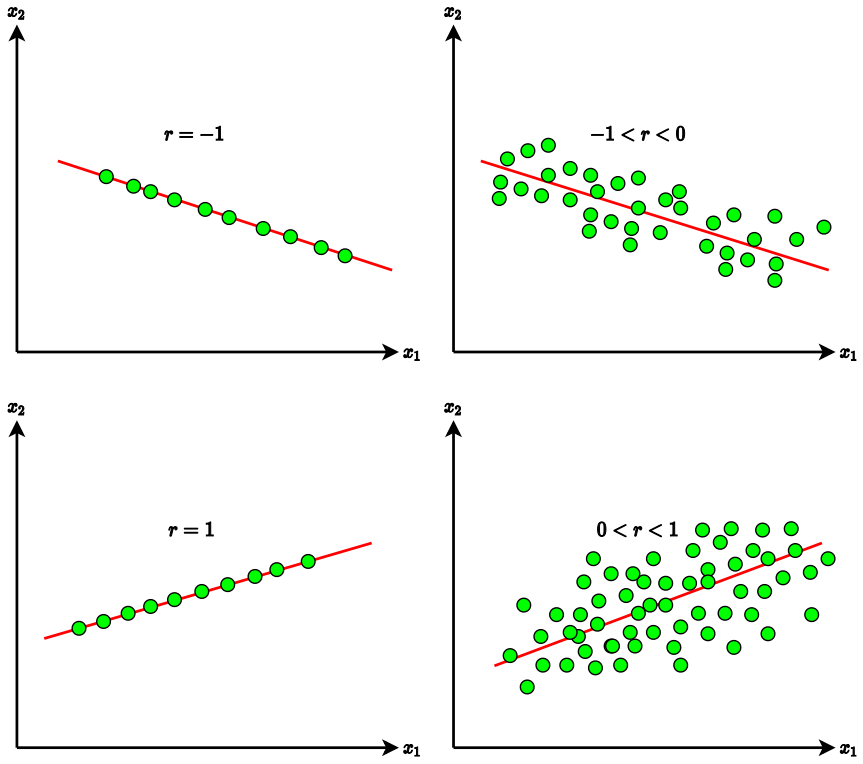


图 2-10: 不同 r 值及其对应的样本点散列图

线性关系)——+1 表示完全正线性相关，−1 表示完全负线性相关；值为 0 时，表示 2 个变量完全不线性相关。图2-10与图2-11展示了几种不同 r 值及其对应的样本点散列图。

需要注意的是，皮尔逊相关系数反映的是 2 个随机变量之间的线性相关程度，且在完全线性相关 (2 个变量表现为直线关系) 的情况下，相关系数与直线斜率并没有直接的关系[8]。

本质上，相关系数反映了 2 个变量之间的协变化 (Co-Varying) 关系——它们是否一起变化，或者各自独立变化而不产生关联；而协方差是反映变量之间协变化关系的一个重要工具。

为了充分地理解 (皮尔逊) 相关系数背后的原理，我们将利用一个样本数据集，从协方差的角度来讨论 2 个变量之间的线性相关关系。表2-3列出了 1955 年 14 个国家几个方面的主要指标数据。

[8]在直线既非水平亦非垂直的情况下，不论斜率如何，相关系数总为 +1 或 −1。对于水平直线或垂直直线，相关系数没有定义，因为总有 1 个变量的方差为 0。根据下文给出的皮尔逊相关系数定义，因方差项出现在分母而直接导致定义无效。

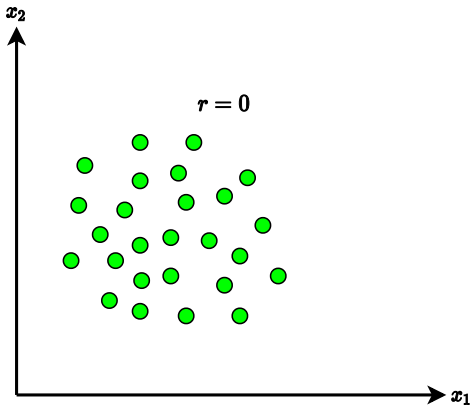


图 2-11: $r=0$ 所对应的样本点散列图

表 2-3: 1955 年 14 个国家几个方面的主要指标数据

Country	GNP per capita($)	Trade($) Millions	Power	Stability	Freedom of Group Opposition	Foreign Conflict	Agreement with US or UN	Defense Budget($) Millions	% GNP for Defense	Acceptance of Int. Law
Brazil	91	2729	7	0	2	0	69.1	148	2.8	0
Burma	51	407	4	0	1	0	-9.5	74	6.9	0
China	58	349	11	0	0	1	-41.7	3054	8.7	0
Cuba	359	1169	3	0	1	0	64.3	53	2.4	0
Egypt	134	923	5	1	1	1	-15.4	158	6.0	1
India	70	2689	10	0	2	0	-28.6	410	1.9	1
Indonesia	129	1601	8	0	1	0	-21.4	267	6.7	0
Israel	515	415	2	1	2	1	42.9	33	2.7	1
Jordan	70	83	1	0	1	1	8.3	29	25.7	0
Netherlands	707	5395	6	1	2	0	52.3	468	6.1	1
Poland	468	1852	9	0	0	1	-41.7	220	1.5	0
USSR	749	6530	13	1	0	1	-41.7	34000	20.4	0
UK	998	18677	12	1	2	1	69.0	3934	7.8	0
US	2334	26836	14	1	2	1	100.0	40641	12.2	1

表 2-4: 各列数据的均值与标准差

MEAN	480.93	4975.36	7.50	0.43	1.21	0.57	14.71	5963.50	7.99	0.36
STD.	593.90	7641.74	4.03	0.49	0.77	0.49	47.89	12915.03	6.87	0.48

对表中各列数据进行横向比较，可以发现，各列中的数据，无论数量大小，还是数量单位，都普遍存在较大的差异。为了建立直观认识，我们对各列数据的均值与标准差进行计算，并将结果列于表2-4中 (由于文本宽度限制，精确到小数点后 2 位)。

从表2-4可以看出，均值与标准差较大的列有第 1-2 列、第 8 列，而较小的列有第 3-7 列、第 9-10 列。造成这个现象的原因为，这些列所对应的 (特征) 变量是对不同类型的指标的反映，而这些指标本身就具有不同的单位，且大小差异性较大。

显然，在计算协方差之前，需要利用前述的标准化公式对数据进行预处理——首先进行中心化，然后除以标准差，从而将绝对量转变为相对量，以消除单位量纲的影响，使得不同单位量纲的变量之间具有可比性。对原始数据进行标准化处理之后，再利用协方差公式来研究各变量之间的线性相关性。标准化结果如表2-5所示 (由于文本宽度限制，精确到小数点后 2 位)。可以看出，标准化后的各列数据被限制于一个相对较小的范围之内，数据变化的幅度并不大。

为了描述方便，对于本例中的数据集，令 $n = 14$ 表示样本数据点的个数，向量 $\boldsymbol{x}_i$ 表示第 i 个样本数据点 ($i = [1, n]$)，令 $m = 10$ 表示每个样本数据点中特征或分量的个数，则 x_{ij} 表示第 i 个样本数据点的第 j 个特征 ($j = [1, m]$)。于是，上述标准化公式可表示如下：

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{12}$$

其中，z_{ij} 表示 x_{ij} 标准化后的值；均值 μ_j 和标准差 σ_j 分别由下面的公式计算：

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij},\ \sigma_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \mu_j)^2} \tag{13}$$

在对样本数据集进行标准化之后，我们就得到了每个样本数据点的各个特征或分量的标准化值 (Standardized Score)。于是，基于这些标准化值 z_{ij}，我们就可

表 2-5: 标准化后的主要指标数据

Country	GNP per capita($)	Trade($) Millions	Power	Stability	Freedom of Group Opposition	Foreign Conflict	Agreement with US or UN	Defense Budget($) Millions	% GNP for Defense	Acceptance of Int. Law
Brazil	-0.66	-0.29	-0.12	-0.87	1.02	-1.15	1.14	-0.45	-0.75	-0.75
Burma	-0.72	-0.60	-0.87	-0.87	-0.28	-1.15	-0.51	-0.46	-0.16	-0.75
China	-0.71	-0.61	0.87	-0.87	-1.57	0.87	-1.18	-0.23	0.10	-0.75
Cuba	-0.21	-0.50	-1.12	-0.87	-0.28	-1.15	1.04	-0.46	-0.81	-0.75
Egypt	-0.58	-0.53	-0.62	1.15	-0.28	0.87	-0.63	-0.45	-0.29	1.34
India	-0.69	-0.30	0.62	-0.87	1.02	-1.15	-0.90	-0.43	-0.89	1.34
Indonesia	-0.59	-0.44	0.12	-0.87	-0.28	-1.15	-0.75	-0.44	-0.19	-0.75
Israel	0.06	-0.60	-1.36	1.15	1.02	0.87	0.59	-0.46	-0.77	1.34
Jordan	-0.69	-0.64	-1.61	-0.87	-0.28	0.87	-0.13	-0.46	2.58	-0.75
Netherlands	0.38	0.05	-0.37	1.15	1.02	-1.15	0.79	-0.43	-0.27	1.34
Poland	-0.02	-0.41	0.37	-0.87	-1.57	0.87	-1.18	-0.44	-0.94	-0.75
USSR	0.45	0.20	1.36	1.15	-1.57	0.87	-1.18	2.17	1.81	-0.75
UK	0.87	1.79	1.12	1.15	1.02	0.87	1.13	-0.16	-0.03	-0.75
US	3.12	2.86	1.61	1.15	1.02	0.87	1.78	2.69	0.61	1.34

以使用协方差来刻画任意 2 个特征或分量之间的线性相关关系：

$$\begin{aligned} r_{jk} &= \sigma(z_{*j}, z_{*k}) = \operatorname{cov}(z_{*j}, z_{*k}) \\ &= \frac{1}{n-1}\sum_{i=1}^{n}(z_{ij}-0)(z_{ik}-0) \\ &= \frac{1}{n-1}\sum_{i=1}^{n} z_{ij}z_{ik} \end{aligned} \tag{14}$$

其中，r_{jk} 表示样本数据集的第 j 个特征与第 k 个特征的协方差[9]；z_{*j} 和 z_{*k} 分别表示标准化后样本数据集的第 j 个特征与第 k 个特征；$\sigma(z_{*j}, z_{*k})$ 表示特征 z_{*j} 与特征 z_{*k} 之间的协方差，也可以表示为 $\operatorname{cov}(z_{*j}, z_{*k})$。

进一步地，为得到 r_{jk} 与原始未标准化的数据集之间的关系，将标准化公式

[9]从下文可以看出，r_{jk} 实际上就是第 j 个特征与第 k 个特征的相关系数。另外，公式中出现的 0，表示 z_{ij} 和 z_{ik} 的均值——标准化后的数据，均值为 0，方差为 1。

(12) 代入公式 (14) 中，得到：

$$\begin{aligned} r_{jk} &= \frac{1}{n-1}\sum_{i=1}^{n} z_{ij}z_{ik} \\ &= \frac{1}{n-1}\sum_{i=1}^{n} \frac{x_{ij}-\mu_j}{\sigma_j}\frac{x_{ik}-\mu_k}{\sigma_k} \\ &= \frac{1}{n-1}\frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sigma_j\sigma_k} \\ &= \frac{\operatorname{cov}(x_{*j},x_{*k})}{\sigma_j\sigma_k} \end{aligned} \tag{15}$$

实际上，上式给出了皮尔逊相关系数的定义[10]。显然，皮尔逊相关系数就是标准化数据 z_{ij} 与 z_{ik} 之间的协方差。从这个角度而言，皮尔逊相关系数是一种特殊的协方差，其能够反映 2 个变量之间的线性相关关系也就不足为奇了。

对于公式 (15)，将标准差的计算公式代入，可进一步变形为如下的等价形式：

$$\begin{aligned} r_{jk} &= \frac{1}{n-1}\frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sigma_j\sigma_k} \\ &= \frac{1}{n-1}\frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ik}-\mu_k)^2}} \\ &= \frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik}-\mu_k)^2}} \end{aligned} \tag{17}$$

[10] 当将皮尔逊相关系数应用于总体时，常使用希腊字母 ρ 表示。本部分的讨论，主要集中于样本数据集，因而使用 r 来表示。对于总体相关系数，根据协方差与方差的定义，还可以得到其它的等价形式：

$$\begin{aligned} \rho_{jk} &= \frac{\operatorname{cov}(x_{*j},x_{*k})}{\sigma_j\sigma_k} = \frac{\mathbb{E}\left[(x_{*j}-\mu_j)(x_{*k}-\mu_k)\right]}{\sigma_j\sigma_k} \\ &= \frac{\mathbb{E}\left[x_{*j}x_{*k}\right]-\mathbb{E}\left[x_{*j}\right]\mathbb{E}\left[x_{*k}\right]}{\sqrt{\mathbb{E}\left[x_{*j}^2\right]-\left(\mathbb{E}\left[x_{*j}\right]\right)^2}\sqrt{\mathbb{E}\left[x_{*k}^2\right]-\left(\mathbb{E}\left[x_{*k}\right]\right)^2}} \end{aligned} \tag{16}$$

对上式展开，并利用均值计算公式，也可得到如下的等价形式：

$$\begin{aligned} r_{jk} &= \frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik}-\mu_k)^2}} \\ &= \frac{\sum_{i=1}^{n}x_{ij}x_{ik}-\sum_{i=1}^{n}\mu_k x_{ij}-\sum_{i=1}^{n}\mu_j x_{ik}+\sum_{i=1}^{n}\mu_j\mu_k}{\sqrt{\sum_{i=1}^{n}x_{ij}^2-2\sum_{i=1}^{n}x_{ij}\mu_j+\sum_{i=1}^{n}\mu_j^2}\sqrt{\sum_{i=1}^{n}x_{ik}^2-2\sum_{i=1}^{n}x_{ik}\mu_k+\sum_{i=1}^{n}\mu_k^2}} \\ &= \frac{\sum_{i=1}^{n}x_{ij}x_{ik}-n\mu_j\mu_k}{\sqrt{\sum_{i=1}^{n}x_{ij}^2-n\mu_j^2}\sqrt{\sum_{i=1}^{n}x_{ik}^2-n\mu_k^2}} \end{aligned} \tag{18}$$

或者：

$$\begin{aligned} r_{jk} &= \frac{\sum_{i=1}^{n}x_{ij}x_{ik}-n\mu_j\mu_k}{\sqrt{\sum_{i=1}^{n}x_{ij}^2-n\mu_j^2}\sqrt{\sum_{i=1}^{n}x_{ik}^2-n\mu_k^2}} \\ &= \frac{\sum_{i=1}^{n}x_{ij}x_{ik}-\frac{1}{n}\sum_{i=1}^{n}x_{ij}\sum_{i=1}^{n}x_{ik}}{\sqrt{\sum_{i=1}^{n}x_{ij}^2-\frac{1}{n}\left(\sum_{i=1}^{n}x_{ij}\right)^2}\sqrt{\sum_{i=1}^{n}x_{ik}^2-\frac{1}{n}\left(\sum_{i=1}^{n}x_{ik}\right)^2}} \\ &= \frac{n\sum_{i=1}^{n}x_{ij}x_{ik}-\sum_{i=1}^{n}x_{ij}\sum_{i=1}^{n}x_{ik}}{\sqrt{n\sum_{i=1}^{n}x_{ij}^2-\left(\sum_{i=1}^{n}x_{ij}\right)^2}\sqrt{n\sum_{i=1}^{n}x_{ik}^2-\left(\sum_{i=1}^{n}x_{ik}\right)^2}} \end{aligned} \tag{19}$$

利用皮尔逊相关系数的等价定义形式 (公式 (17))，可以较为方便地解释 x_{*j} 与 x_{*k} 完全线性相关时 r_{jk} 的取值结果。因 x_{*j} 与 x_{*k} 满足线性关系，不妨令：

$$x_{*k} = \alpha x_{*j} + \beta \tag{20}$$

其中 α 为斜率且 $\alpha \neq 0$，β 为截距。将上式代入公式 (17) 中，并利用公式

$\mu_k = \frac{1}{n}\sum_{i=1}^{n} x_{ik} = \frac{1}{n}\sum_{i=1}^{n}(\alpha x_{ij} + \beta) = \alpha\mu_j + \beta$，得到：

$$\begin{aligned} r_{jk} &= \frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik}-\mu_k)^2}} \\ &= \frac{\alpha\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ij}-\mu_j)}{|\alpha|\sqrt{\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}} = \pm 1 \end{aligned} \tag{21}$$

可以看出，在完全线性相关时，r_{jk} 的取值依 α 的值而定：$\alpha > 0$，$r_{jk} = 1$；$\alpha < 0$，$r_{jk} = -1$。

如果将 $x_{ij} - \mu_j$ 和 $x_{ik} - \mu_k$ 分别看作 n 维向量 $\boldsymbol{x}_j - \boldsymbol{\mu}_j$ 和 n 维向量 $\boldsymbol{x}_k - \boldsymbol{\mu}_k$ 的一个分量，那么皮尔逊相关系数的等价定义形式 (公式 (17))，可以表示为：

$$\begin{aligned} r_{jk} &= \frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik}-\mu_k)^2}} \\ &= \frac{\left(\boldsymbol{x}_j - \boldsymbol{\mu}_j\right)^T\left(\boldsymbol{x}_k - \boldsymbol{\mu}_k\right)}{\|\boldsymbol{x}_j - \boldsymbol{\mu}_j\|\|\boldsymbol{x}_k - \boldsymbol{\mu}_k\|} \end{aligned} \tag{22}$$

其中，$\boldsymbol{x}_j = (x_{1j}, x_{2j}, \cdots, x_{nj})^T$，$\boldsymbol{\mu}_j = (\mu_j, \mu_j, \cdots, \mu_j)^T$，$\boldsymbol{x}_k = (x_{1k}, x_{2k}, \cdots, x_{nk})^T$，$\boldsymbol{\mu}_k = (\mu_k, \mu_k, \cdots, \mu_k)^T$。

实际上，公式 (22) 就是 n 维向量 $\boldsymbol{x}_j - \boldsymbol{\mu}_j$ 与 n 维向量 $\boldsymbol{x}_k - \boldsymbol{\mu}_k$ 之间的夹角余弦 (参见公式 (11))，即：

$$r_{jk} = \cos\theta \tag{23}$$

其中 θ 为 n 维向量 $\boldsymbol{x}_j - \boldsymbol{\mu}_j$ 与 n 维向量 $\boldsymbol{x}_k - \boldsymbol{\mu}_k$ 之间的夹角。需要注意的是，$\boldsymbol{x}_j$ 与 $\boldsymbol{x}_k$ 分别表示所有样本数据点的第 j 列特征与第 k 列特征组成的向量，而不是表示第 j 个和第 k 个样本数据点。在此前提下，皮尔逊相关系数 r_{jk} 可以解释为，对 n 维向量 $\boldsymbol{x}_j$ 与 n 维向量 $\boldsymbol{x}_k$ 进行中心化 (分别得到 $\boldsymbol{x}_j - \boldsymbol{\mu}_j$ 与 $\boldsymbol{x}_k - \boldsymbol{\mu}_k$) 之后的两向量之间的夹角余弦。

显然，如果 $\boldsymbol{x}_j - \boldsymbol{\mu}_j$ 与 $\boldsymbol{x}_k - \boldsymbol{\mu}_k$ 中的每个分量均符合一致的线性关系，则 $\theta = 0°$ 或 $\theta = 180°$，此时 $r_{jk} = 1$ 或 $r_{jk} = -1$。而在其余情况下，$-1 < r_{jk} < 1$。其几何解释与示意图，可参见图2-10与图2-11。

如果我们单独对变量 x_{ij} 和 x_{ik} 进行如下的线性变换：

$$\begin{cases} x_{ij} \Rightarrow ax_{ij} + b \\ x_{ik} \Rightarrow cx_{ik} + d \end{cases} \tag{24}$$

其中，$a > 0$ 与 $c > 0$ 且 a、b、c、d 均为常量，则它们之间的相关系数并不会发生变化。由上式可计算出新的均值：

$$\begin{cases} \mu_j \Rightarrow a\mu_j + b \\ \mu_k \Rightarrow c\mu_k + d \end{cases} \tag{25}$$

将新变量及新均值代入公式 (17)，可得：

$$\begin{aligned} r_{jk}^{new} &= \frac{\sum_{i=1}^{n} a(x_{ij} - \mu_j)c(x_{ik} - \mu_k)}{\sqrt{\sum_{i=1}^{n} a^2(x_{ij} - \mu_j)^2}\sqrt{\sum_{i=1}^{n} c^2(x_{ik} - \mu_k)^2}} \\ &= \frac{\sum_{i=1}^{n}(x_{ij} - \mu_j)(x_{ik} - \mu_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \mu_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik} - \mu_k)^2}} = r_{jk}^{old} \end{aligned} \tag{26}$$

这表明，如果对变量进行各自独立的 (正) 缩放与独立的平移，并不会改变它们的相关系数，这是皮尔逊相关系数的一个重要性质。

对于样本数据集而言，皮尔逊相关系数 r 的平方 r^2 是决定系数 (Coefficient of Determination) 的一种特殊情况。一般地，决定系数 r^2 用于度量因变量的变化中可由自变量进行解释的部分所占的比例，并以此来判断统计模型的解释力。例如，对于简单的线性回归模型而言，决定系数为样本观测值与拟合估算值之间的相关系数的平方。当加入其他回归自变量后，决定系数相应地变为多重相关系数的平方。决定系数可用作回归模型中拟合效果的评价标准，被称为拟合优度 (Goodness-of-Fit)，其值越接近 1，拟合效果越好。

下面，从基于最小二乘法的线性回归模型的角度，来讨论相关系数与决定系数之间的关系及决定系数所表达的含义。

假设 x_{*j} 为自变量，x_{*k} 为因变量，且假设使用最小二乘法对 x_{*k} 进行线性拟合，即对于每个自变量 x_{*j}，使用线性拟合公式 $\hat{x}_{*k} = \alpha x_{*j} + \beta$ 计算 x_{*k} 的相应估算值 $\hat{x}_{*k}$，其中 α 与 β 是需要依据最小二乘法确定的参数 (它们分别是直线的斜率与截距)[11]。

[11]下面，我们不会讨论如何确定参数 α 与 β，而只是借助于与求解这 2 个参数类似的方法来证明我们所需要的结论。关于参数 α 与 β 的求解方法与结果的讨论，请阅读文献 [20]。

首先，列出最小二乘法的目标函数。一般地，可以使用残差平方和 (Residual Sum of Squares, RSS) 或误差平方和作为优化准则：

$$RSS = \sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik})^2 = \sum_{i=1}^{n} (x_{ik} - \alpha x_{ij} - \beta)^2 \tag{27}$$

对该目标函数按 β 求取偏导数，并令其等于 0，得到：

$$\begin{aligned} &\frac{\partial RSS}{\partial \beta} = -2 \sum_{i=1}^{n} (x_{ik} - \alpha x_{ij} - \beta) = 0 \quad \Rightarrow \\ &\beta = \mu_k - \alpha \mu_j \end{aligned} \tag{28}$$

对该目标函数按 α 求取偏导数，令其等于 0，并将 β 代入，得到：

$$\begin{aligned} &\frac{\partial RSS}{\partial \alpha} = -2 \sum_{i=1}^{n} (x_{ik} - \alpha x_{ij} - \beta) x_{ij} = 0 \quad \Rightarrow \\ &\sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik}) \frac{(\hat{x}_{ik} - \beta)}{\alpha} = 0 \quad \Rightarrow \\ &\sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik}) (\hat{x}_{ik} - \mu_k + \alpha \mu_j) = 0 \quad \Rightarrow \\ &\sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik}) (\hat{x}_{ik} - \mu_k) + \sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik}) \alpha \mu_j = 0 \quad \Rightarrow \\ &\sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik}) (\hat{x}_{ik} - \mu_k) = 0 \end{aligned} \tag{29}$$

上述结论非常有用，我们将其用于接下来的公式推导中。

进一步地，从下面的公式出发，对其进行分解并应用上述结论，将得到：

$$\begin{aligned} \sum_{i=1}^{n} (x_{ik} - \mu_k)^2 &= \sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik} + \hat{x}_{ik} - \mu_k)^2 \\ &= \sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik})^2 + 2 \sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik}) (\hat{x}_{ik} - \mu_k) + \sum_{i=1}^{n} (\hat{x}_{ik} - \mu_k)^2 \\ &= \sum_{i=1}^{n} (x_{ik} - \hat{x}_{ik})^2 + \sum_{i=1}^{n} (\hat{x}_{ik} - \mu_k)^2 \end{aligned} \tag{30}$$

上式表达的意义是什么？为了获得一个较为直观的认识，我们可以将样本数据点绘制于直角坐标系中，其中 x_{*j} 表示横坐标，x_{*k} 表示纵坐标，示意图如图2-12所示。

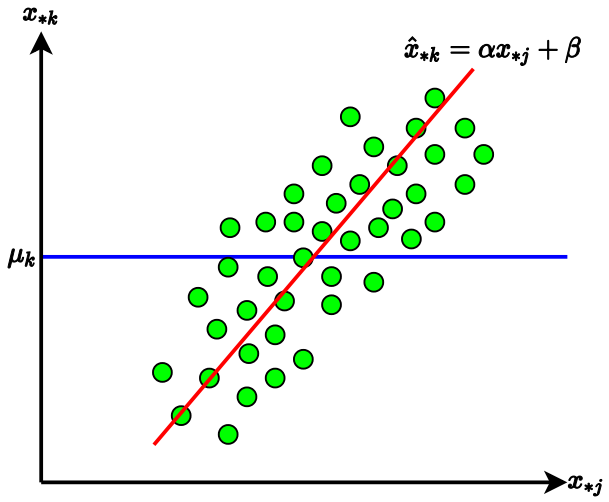


图 2-12: 简单的平均值拟合与最小二乘法拟合

在图2-12中，蓝色直线表示对样本数据集的最简单拟合——平均值拟合，即 $\mu_k = \frac{1}{n}\sum_{i=1}^{n} x_{ik}$；红色直线表示对样本数据集在高斯分布假设下的最优拟合——最小二乘法拟合，即 $\hat{x}_{*k} = \alpha x_{*j} + \beta$。

在此解释下，公式 (30) 所表达的含义为，使用最简单拟合所产生的误差平方和 (即总误差)，可分解为 2 部分：使用最优拟合所产生的误差平方和；最优拟合与最简单拟合之间的误差平方和[12]。

继续对公式 (30) 进行变形，可得：

$$\begin{aligned} &\sum_{i=1}^{n}(x_{ik}-\mu_k)^2 = \sum_{i=1}^{n}(x_{ik}-\hat{x}_{ik})^2 + \sum_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2 \quad \Rightarrow \\ &1 = \frac{\sum_{i=1}^{n}(x_{ik}-\hat{x}_{ik})^2}{\sum_{i=1}^{n}(x_{ik}-\mu_k)^2} + \frac{\sum_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}{\sum_{i=1}^{n}(x_{ik}-\mu_k)^2} \end{aligned} \tag{31}$$

进一步地，为了描述方便，定义总 (误差) 平方和 (Total Sum of Squares) 为：

$$SS_{tot} = \sum_{i=1}^{n}(x_{ik}-\mu_k)^2 \tag{32}$$

定义残差平方和 (Residual Sum of Squares) 为：

$$SS_{res} = \sum_{i=1}^{n}(x_{ik}-\hat{x}_{ik})^2 \tag{33}$$

[12]实际上，如果在公式 (30) 的两端同时乘以因子 $\frac{1}{n-1}$，则可以将该式解释为相应的方差之和。

在上述定义下，公式 (31) 可表示为：

$$\frac{\sum\limits_{i=1}^{n}\left(\hat{x}_{ik}-\mu_k\right)^2}{\sum\limits_{i=1}^{n}\left(x_{ik}-\mu_k\right)^2}=1-\frac{SS_{res}}{SS_{tot}} \tag{34}$$

对于上式右端，当 $SS_{res}=0$ 时，最优拟合直线完全拟合了样本数据集，所有的样本数据点都分布在最优拟合直线上，即每个拟合估算值 $\hat{x}_{ik}$ 与其对应的数据观测值 x_{ik} 完全相符，或者，拟合估算值 $\hat{x}_{*k}$ 与其对应的数据观测值 x_{*k} 完全线性相关；另一方面，此时，x_{*k} 与 x_{*j} 也是完全线性相关的。此时，上式左端项的值为 1。这似乎预示着该公式与相关系数之间存在着某种关联。情况的确如此。

下面考察数据观测值 x_{*k} 与拟合估算值 $\hat{x}_{*k}$ 之间的皮尔逊相关系数 $r_{k\hat{k}}$，使用等价公式 (17) 的形式，可得[13]：

$$\begin{aligned}
r_{k\hat{k}} &= \frac{\sum\limits_{i=1}^{n}(x_{ik}-\mu_k)(\hat{x}_{ik}-\mu_k)}{\sqrt{\sum\limits_{i=1}^{n}(x_{ik}-\mu_k)^2}\sqrt{\sum\limits_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}} \\
&= \frac{\sum\limits_{i=1}^{n}(x_{ik}-\hat{x}_{ik}+\hat{x}_{ik}-\mu_k)(\hat{x}_{ik}-\mu_k)}{\sqrt{\sum\limits_{i=1}^{n}(x_{ik}-\mu_k)^2}\sqrt{\sum\limits_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}} \\
&= \frac{\sum\limits_{i=1}^{n}(x_{ik}-\hat{x}_{ik})(\hat{x}_{ik}-\mu_k)+\sum\limits_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}{\sqrt{\sum\limits_{i=1}^{n}(x_{ik}-\mu_k)^2}\sqrt{\sum\limits_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}} \\
&= \frac{\sum\limits_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}{\sqrt{\sum\limits_{i=1}^{n}(x_{ik}-\mu_k)^2}\sqrt{\sum\limits_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}} = \sqrt{\frac{\sum\limits_{i=1}^{n}(\hat{x}_{ik}-\mu_k)^2}{\sum\limits_{i=1}^{n}(x_{ik}-\mu_k)^2}}
\end{aligned} \tag{35}$$

综合公式 (34) 和公式 (35)，得到：

$$r_{k\hat{k}}^2=\frac{\sum\limits_{i=1}^{n}\left(\hat{x}_{ik}-\mu_k\right)^2}{\sum\limits_{i=1}^{n}\left(x_{ik}-\mu_k\right)^2}=1-\frac{SS_{res}}{SS_{tot}} \tag{36}$$

[13]假设噪声 ϵ 服从均值 0 的高斯分布，则数据观测值 x_{*k} 与拟合估算值 $\hat{x}_{*k}$ 之间具有如下关系：$x_{*k}=\hat{x}_{*k}+\epsilon=\alpha x_{*j}+\beta+\epsilon$，则 $\mathbb{E}\left[x_{*k}\right]=\mathbb{E}\left[\hat{x}_{*k}\right]$。因此，数据观测值 x_{*k} 与拟合估算值 $\hat{x}_{*k}$ 之间的皮尔逊相关系数 $r_{k\hat{k}}$，是期望均值意义下计算得到的。

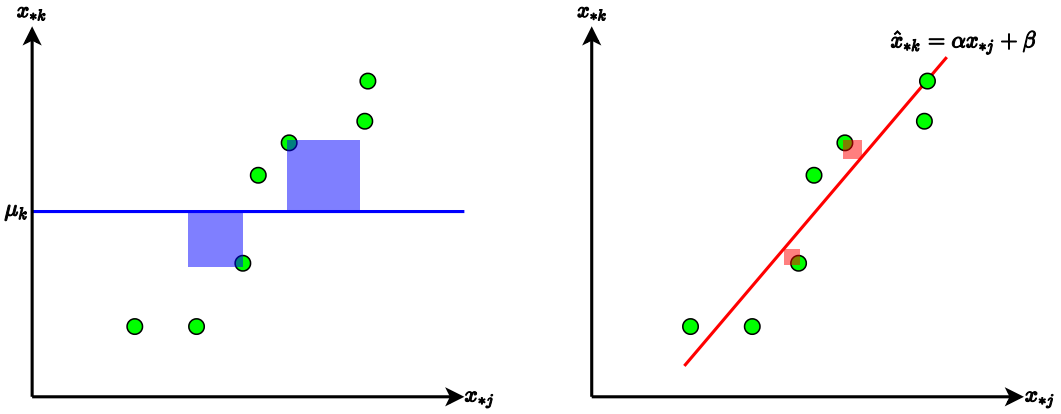


图 2-13: 平均值拟合与最小二乘法拟合的残差平方示意图

至此，我们得到了决定系数或回归意义下的拟合优度。当 $SS_{res} = 0$ 时，拟合优度取得最大值 1。此时，拟合估算值 $\hat{x}_{*k}$ 与其对应的数据观测值 x_{*k} 完全线性相关。换言之，决定系数或拟合优度反映了拟合质量。一般地，从另一个角度看，$r^2_{k\hat{k}}$ 也反映了因变量 x_{*k} 的拟合质量中能够由自变量 x_{*j} 进行解释的比例。于是，定义回归平方和 (Regression Sum of Squares)[14] 为：

$$SS_{reg} = \sum_{i=1}^{n} (\hat{x}_{ik} - \mu_k)^2 \tag{37}$$

则决定系数也可以写为：

$$r^2_{k\hat{k}} = \frac{SS_{reg}}{SS_{tot}} = 1 - \frac{SS_{res}}{SS_{tot}} \tag{38}$$

其意义为，当 SS_{tot} 全部由 SS_{reg} 组成时，拟合优度取得最大值。图2-13展示了平均值拟合与最小二乘法拟合的残差平方示意图。左图中，蓝色方形表示使用平均值拟合时样本数据点的残差平方；右图中，红色方形表示使用最小二乘法拟合时样本数据点的残差平方。所有红色方形的面积之和越小，拟合优度的值越接近于 1。为清晰起见，其余红色方形未绘制出。

2.2 簇之间的相似度

簇或类是由样本数据集中的样本点组成的。对于硬聚类 (Hard Clustering) 而言，一个样本点只能属于一个簇，或者，不同簇之间的交集为空。与硬聚类相反，

[14] 回归平方和也被称为解释平方和 (Explained Sum of Squares)。

软聚类允许一个样本点属于多个簇，或者允许不同簇之间的交集不为空。在本章，我们只考虑硬聚类。

下面，给出簇的常用特征，它们从不同的角度刻画了簇：

- 中心或均值

$$\boldsymbol{\mu}_{C_i} = \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{x}_j \tag{39}$$

其中，$\boldsymbol{x}_j \in C_i$ 表示簇 C_i 中的样本数据点 (m 维向量)，n 表示簇 C_i 中样本数据点的个数。

- 直径

$$d_{C_i} = \max\{d_{jk}|\boldsymbol{x}_j, \boldsymbol{x}_k \in C_i\} \tag{40}$$

其中，d_{jk} 表示簇 C_i 中样本数据点 $\boldsymbol{x}_j$ 与 $\boldsymbol{x}_k$ 之间的距离。簇的直径被定义为所有 d_{jk} 中的最大值。样本数据点的距离准则，可以依据问题的性质进行定义。

- 散布矩阵与协方差矩阵

簇的散布矩阵 (Scatter Matrix) 定义为：

$$\boldsymbol{S}_{C_i} = \sum_{j=1}^{n}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_{C_i}\right)\left(\boldsymbol{x}_j - \boldsymbol{\mu}_{C_i}\right)^T \tag{41}$$

其中，$\boldsymbol{x}_j \in C_i$ 表示簇 C_i 中的样本数据点 (m 维向量)，n 表示簇 C_i 中样本数据点的个数，$\boldsymbol{S}_{C_i}$ 为半正定 $m \times m$ 矩阵。

针对样本数据集，无偏差协方差矩阵被定义为归一化的散布矩阵 (Normalized Scatter Matrix)：

$$\boldsymbol{\Sigma}_{C_i} = \frac{1}{n-1}\boldsymbol{S}_{C_i} = \frac{1}{n-1}\sum_{j=1}^{n}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_{C_i}\right)\left(\boldsymbol{x}_j - \boldsymbol{\mu}_{C_i}\right)^T \tag{42}$$

如果样本数据点 $\boldsymbol{x}_j$ 服从多元正态分布，那么在最大似然估算中，协方差矩阵可以定义为：

$$\boldsymbol{\Sigma}_{C_i} = \frac{1}{n}\boldsymbol{S}_{C_i} = \frac{1}{n}\sum_{j=1}^{n}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_{C_i}\right)\left(\boldsymbol{x}_j - \boldsymbol{\mu}_{C_i}\right)^T \tag{43}$$

在一些聚类算法中，例如层次聚类算法等，还需要考虑簇间距离或相似度。簇间距离也被称为连接 (Linkage)，也存在着多种定义形式：

- 最短距离或单连接 (Single Linkage)

$$d\left(C_i, C_j\right) = \min\{d_{kl} | \boldsymbol{x}_k \in C_i, \boldsymbol{x}_l \in C_j\} \tag{44}$$

其中，C_i 与 C_j 为簇。

- 最长距离或完全连接 (Complete Linkage)

$$d\left(C_i, C_j\right) = \max\{d_{kl} | \boldsymbol{x}_k \in C_i, \boldsymbol{x}_l \in C_j\} \tag{45}$$

其中，C_i 与 C_j 为簇。

- 中心距离

$$d\left(C_i, C_j\right) = d_{\boldsymbol{\mu}_{C_i}\boldsymbol{\mu}_{C_j}} \tag{46}$$

其中，C_i 与 C_j 为簇；$d_{\boldsymbol{\mu}_{C_i}\boldsymbol{\mu}_{C_j}}$ 表示 2 个聚类中心 $\boldsymbol{\mu}_{C_i}$ 与 $\boldsymbol{\mu}_{C_j}$ 之间的距离。

- 平均距离

$$d\left(C_i, C_j\right) = \frac{1}{n_i n_j} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} d_{kl} \tag{47}$$

其中，n_i 与 n_j 分别表示簇 C_i 与簇 C_j 中样本数据点的个数；d_{kl} 表示 $\boldsymbol{x}_k \in C_i$ 与 $\boldsymbol{x}_l \in C_j$ 之间的距离。

3 k 均值算法的推导

“k 均值 (k-Means)”一词首先由 James MacQueen 于 1967 年提出，而相关的思想则由 Hugo Steinhaus 于 1956 年提出。标准的 k 均值算法由贝尔实验室的 Stuart Lloyd 于 1957 年提出，但正式作为论文发表却是在 1982 年。Edward W. Forgy 于 1965 年发表了本质上与 Lloyd 所提出的 k 均值一样的算法。因此，k 均值算法有时候也被称为 Lloyd-Forgy 算法。

k 均值是一种直接对样本数据集进行划分的聚类算法。由于数据集缺少标签或类别信息，算法需要依据欧氏距离准则，将相似的样本数据点尽可能地划分到同一个数据子集中而形成一个簇，且使得不相似的样本数据点尽可能地相互分离。一般地，算法执行完毕后，样本数据点将形成 K 个具有不同性质的簇，每个样本数据点到其所属簇的中心的距离最小[15]。

[15]为公式描述的方便，使用 K 表示簇或类别的个数，而不是使用 k。k 均值算法的划分方法将样本空间分割成许多的 Voronoi 单元 (Voronoi Cell)。关于 Voronoi 单元、Voronoi 图及相关性质与应用的讨论，请参考文献 [25]-[26]。

需要注意的是，k 均值算法中的 K 值需要事先指定，而且不同的初始簇中心会得到不同的聚类结果，算法在迭代过程中易于陷入局部最优，不能保证收敛到全局最优。

下面，给出 k 均值算法的推导过程。

假设样本数据集 $\boldsymbol{X}$ 由 N 个数据点 $\boldsymbol{x}_n(n=[1,2,\cdots,N])$ 组成，其中 $\boldsymbol{x}_n$ 为 M 维 (列) 向量。k 均值算法的目标为，将 N 个数据点划分到 $K(K<N)$ 个不同的簇或类别 $C_k(k=[1,2,\cdots,K])$ 中 (硬聚类)。换句话说，K 个簇 C_k 是对样本数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$ 的一个硬划分：$C_k\bigcap C_l=\emptyset(k\neq l)$ 和 $\bigcup\limits_{k=1}^{K}C_k=\boldsymbol{X}$。

直观地，每一个簇 C_k 应该有一个中心——簇中心或聚类中心，使用符号 $\boldsymbol{\mu}_k$ 来表示[16]，且簇内每个样本数据点到簇中心的距离要小于它们各自到其它簇中心的距离。因此，k 均值算法的目标是：

- 为每个样本数据点找到其所属的簇或类别；
- 为每个簇找到一组均值 $\{\boldsymbol{\mu}_k\}$，使得每个样本数据点到其簇中心的距离的平方和最小；

为描述方便，引入二值指示变量 r_{nk}，用来表示样本数据点 $\boldsymbol{x}_n$ 是否属于簇或类别 k：$r_{nk}=1$，属于簇 k；否则，不属于簇 k。实际上，$\boldsymbol{r}_n=(r_{n1},r_{n2},\cdots,r_{nk})$ 就构成了一组类别表示向量，被称为 One-Hot 向量或 1-of-K 向量。对任一 $\boldsymbol{x}_n$，它满足如下关系：

$$\sum_{k=1}^{K}r_{nk}=1 \tag{48}$$

根据上述讨论，我们定义 k 均值算法的优化目标为：

$$L=\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2 \tag{49}$$

从上式可以看出，目标函数包含 2 类待优化的参数，即 r_{nk} 与 $\boldsymbol{\mu}_k$。r_{nk} 表示样本数据点 $\boldsymbol{x}_n$ 是否属于类别 C_k，而 $\boldsymbol{\mu}_k$ 为类别中心，由该类别内的所有样本数据点共同确定。此外，在算法的迭代过程中，r_{nk} 与 $\boldsymbol{\mu}_k$ 均处于动态调整中，直至算法收敛而稳定为止[17]。因此，最终的分类结果是由 r_{nk} 与 $\boldsymbol{\mu}_k$ 一起决定的。也正

[16]从下文的推导结果来看，簇中心等于该簇内所有样本数据点的均值。这也是该算法被称为 k 均值算法的原因。

[17]从下文分析可以看出，k 均值算法可以看作高斯混合模型 EM 算法的一个特例。关于 EM 算法及高斯混合模型，请阅读《机器学习》课程系列之“EM 算法”。

是由于这个原因，k 均值算法不一定能够找到全局最优解——在迭代的过程中，易于陷入局部最优。

由于存在 2 个待优化的变量，不易同时进行。此处，可以采取一种分阶段优化并反复迭代的方式，以得到最优参数或次优参数。

具体地，每次迭代包括 2 个连续的优化阶段——分别对应于 r_{nk} 与 $\boldsymbol{\mu}_k$ 的优化。在第 1 阶段，固定 $\boldsymbol{\mu}_k$ 值，对 r_{nk} 进行优化。例如，在算法伊始，可以随机初始化每个类别的 $\boldsymbol{\mu}_k$ 值；而在算法的后续每次迭代中，固定住 $\boldsymbol{\mu}_k$ 的值，即意味着使用上一阶段得到的 $\boldsymbol{\mu}_k$ 值。在第 2 阶段，固定 r_{nk} 值，即意味着使用上一阶段得到的 r_{nk} 值，对 $\boldsymbol{\mu}_k$ 进行优化。上述过程，周而复始，循环往复，直至算法收敛或满足要求为止。

首先，固定 $\boldsymbol{\mu}_k$，对 r_{nk} 进行优化。观察到 One-Hot 向量 $\boldsymbol{r}_n=(r_{n1},r_{n2},\cdots,r_{nk})$ 依赖于各自的样本数据点 $\boldsymbol{x}_n$，即各个 $\boldsymbol{r}_n$ 之间是相互独立的，因而可以单独地对每个样本数据点 $\boldsymbol{x}_n$ 的 $\boldsymbol{r}_n$ 取值进行优化：

$$\min_{\{\boldsymbol{r}_1,\cdots,\boldsymbol{r}_N\}} L=\min_{\{\boldsymbol{r}_1,\cdots,\boldsymbol{r}_N\}}\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2=\sum_{n=1}^{N}\min_{\boldsymbol{r}_n}\sum_{k=1}^{K}r_{nk}\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2 \tag{50}$$

于是，得到：

$$\min_{\boldsymbol{r}_n}\sum_{k=1}^{K}r_{nk}\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2 \quad\Leftrightarrow\quad r_{nk}=\begin{cases}1,\ if\ k=\arg\min\limits_{j=1,\cdots,K}\|\boldsymbol{x}_n-\boldsymbol{\mu}_j\|^2\\ 0,\ else\end{cases} \tag{51}$$

上式表达的含义为，在第 1 阶段，对 r_{nk} 进行优化时，可以简单地将每个样本数据点 $\boldsymbol{x}_n$ 与离自己最近的聚类中心 $\boldsymbol{\mu}_k$ 聚为同一类，并设置 $r_{nk}=1$，而 $r_{nl}=0(l\neq k)$。

然后，固定 r_{nk}，对 $\boldsymbol{\mu}_k$ 进行优化。其过程为，求解偏导数 $\frac{\partial L}{\partial \boldsymbol{\mu}_k}$，令其等于 $\mathbf{0}$，最后可解出 $\boldsymbol{\mu}_k$。由于 $\frac{\partial L}{\partial \boldsymbol{\mu}_k}$ 为标量对向量求导，需要使用迹技巧[18]。

为此，对目标函数 L 进行变形：

$$\begin{aligned}L&=\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2\\&=\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\left(\boldsymbol{x}_n-\boldsymbol{\mu}_k\right)^T\left(\boldsymbol{x}_n-\boldsymbol{\mu}_k\right)\\&=\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\left(\boldsymbol{x}_n^T\boldsymbol{x}_n-2\boldsymbol{x}_n^T\boldsymbol{\mu}_k+\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k\right)\end{aligned} \tag{52}$$

[18]关于标量对向量及矩阵的求导，请阅读《机器学习》课程系列之“判别函数的线性分类基础”。

于是，得到：

$$\begin{aligned} dL &= \mathrm{tr}(dL) \\ &= \mathrm{tr}\left\{\sum_{n=1}^{N} r_{nk}\left(-2\boldsymbol{x}_n^T d\boldsymbol{\mu}_k + d\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T d\boldsymbol{\mu}_k\right)\right\} \\ &= \mathrm{tr}\left\{\sum_{n=1}^{N} r_{nk}\left(-2\boldsymbol{x}_n^T d\boldsymbol{\mu}_k + 2\boldsymbol{\mu}_k^T d\boldsymbol{\mu}_k\right)\right\} \end{aligned} \tag{53}$$

则：

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} r_{nk}\left(-2\boldsymbol{x}_n + 2\boldsymbol{\mu}_k\right) \tag{54}$$

令上式等于 $\mathbf{0}$，可得：

$$\sum_{n=1}^{N} r_{nk}\left(-2\boldsymbol{x}_n + 2\boldsymbol{\mu}_k\right) = \mathbf{0} \tag{55}$$

解出 $\boldsymbol{\mu}_k$，得到：

$$\boldsymbol{\mu}_k = \frac{\sum\limits_{n=1}^{N} r_{nk}\boldsymbol{x}_n}{\sum\limits_{n=1}^{N} r_{nk}} \tag{56}$$

上式的意义也非常明确，簇 C_k 的聚类中心 $\boldsymbol{\mu}_k$ 等于簇内所有样本数据点的均值。

上面的 2 个阶段，完成了 k 均值算法的一次迭代。接着，重新为每个样本数据点分配簇 (优化 r_{nk}) 以及重新计算聚类中心 (优化 $\boldsymbol{\mu}_k$)，便完成了新一轮迭代。上述过程，一直迭代下去，直至簇的分配稳定或达到某个终止条件为止。在算法的每一次迭代后，目标函数 L 的值都减少了，因而算法的收敛性得到了保证。但是，k 均值算法与 EM 算法一样，无法保证找到全局最优解。

在实际应用中，k 均值算法经常被用作初始化高斯混合模型的参数。从算法执行效率的角度而言，直接实现的 k 均值算法效率相当低，尤其是对于大规模数据集而言。原因在于，每次迭代的第 1 阶段，都需要计算每个样本数据点与聚类中心的欧氏距离。为此，研究者们已经提出了一些加速方法。例如，某些方法将数据集预处理成树状结构，使得相邻的数据点位于同一棵子树上；而另一些算法，利用距离的三角不等式，从而避免了一些不必要的计算。此外，研究者们还提出了增量 k 均值算法，以在线增量的方式更新相关参数。

值得注意的是，k 均值算法以欧氏距离作为相似度准则，这不仅限制了算法能够处理的数据变量的类型，也使得聚类中心对于异常点不具有鲁棒性。于是，一些使用其它相似度准则的算法也已经被提出，例如 k 中心点 (k-Medoids) 算法。

下面给出 k 均值算法。

算法 **3.1**（k 均值算法）

def kMeans($\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$):

1. Initialize $\boldsymbol{\mu}_k$, $k = 1, \cdots, K$
2. Repeat:
3. Assign each sample data point to its closest cluster center:

$$r_{nk} = \begin{cases} 1, \ if \ k = \arg\min\limits_{j=1,\cdots,K} \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0, \ else \end{cases}$$

4. Update each cluster center by computing the mean of all sample points
5. assigned to it:

$$\boldsymbol{\mu}_k = \frac{\sum\limits_{n=1}^{N} r_{nk}\boldsymbol{x}_n}{\sum\limits_{n=1}^{N} r_{nk}}$$

6. Until converged

4 与高斯混合模型 EM 算法的关系

在“EM 算法”一章中，我们介绍了 EM 算法及高斯混合模型。实际上，k 均值算法可以看作高斯混合模型 EM 算法的一种极限情况。

首先，简单回顾一下高斯混合模型。它是一种具有如下形式的概率模型：

$$P(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_k \phi(\boldsymbol{x};\boldsymbol{\theta}_k) \tag{57}$$

其中，$\alpha_k (\alpha_k \geqslant 0)$ 为权重系数，满足条件 $\sum\limits_{k=1}^{K} \alpha_k = 1$，$K$ 表示基模型或分模型的个数；$\phi(\boldsymbol{x};\boldsymbol{\theta}_k)$ 是高斯分布，$\boldsymbol{\theta}_k = (\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$：

$$\phi(\boldsymbol{x};\boldsymbol{\theta}_k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{d}{2}}} \frac{1}{|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)\right\} \tag{58}$$

其中，d 维向量 $\boldsymbol{\mu}_k$ 被称为均值向量，$d \times d$ 维矩阵 $\boldsymbol{\Sigma}_k$ 被称为协方差矩阵，$|\boldsymbol{\Sigma}_k|$ 表示 $\boldsymbol{\Sigma}_k$ 的行列式；$\boldsymbol{\theta} = (\alpha_1, \alpha_2, \ldots, \alpha_K, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$。

现在，假设所有的高斯分布分量 $\phi(\boldsymbol{x};\boldsymbol{\theta}_k)$ 的协方差矩阵 $\boldsymbol{\Sigma}_k=\epsilon\boldsymbol{I}$(其中 $\boldsymbol{I}$ 为单位矩阵，$k=[1,\cdots,K]$)，且假设 ϵ 为常数。于是，$\phi(\boldsymbol{x};\boldsymbol{\theta}_k)$ 可以简化为：

$$\begin{aligned}\phi(\boldsymbol{x};\boldsymbol{\theta}_k)&=\frac{1}{(2\pi\epsilon)^{\frac{d}{2}}}\exp\left\{-\frac{1}{2\epsilon}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T(\boldsymbol{x}-\boldsymbol{\mu}_k)\right\}\\&=\frac{1}{(2\pi\epsilon)^{\frac{d}{2}}}\exp\left\{-\frac{1}{2\epsilon}\|\boldsymbol{x}-\boldsymbol{\mu}_k\|^2\right\}\end{aligned}\tag{59}$$

于是，我们可以对上述特殊的高斯混合模型应用 EM 算法[19]。在 E 步，对于一个特定的数据点 $\boldsymbol{x}_n$，其属于类别 k 的概率为：

$$\begin{aligned}P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t)&=\frac{\alpha_k^t\phi(\boldsymbol{x}_n;\boldsymbol{\theta}_k^t)}{\sum\limits_{j=1}^{K}\alpha_j^t\phi(\boldsymbol{x}_n;\boldsymbol{\theta}_j^t)}\\&=\frac{\alpha_k^t\exp\left\{-\frac{\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2}{2\epsilon}\right\}}{\sum\limits_{j=1}^{K}\alpha_j^t\exp\left\{-\frac{\|\boldsymbol{x}_n-\boldsymbol{\mu}_j\|^2}{2\epsilon}\right\}}\end{aligned}\tag{60}$$

在一般情况下，利用上式及 M 步的几个更新公式，我们可以对样本数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$ 进行软聚类，因为 $P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$ 的取值范围为 $[0,1]$。

但是，在考虑 $\epsilon\to 0$ 的极限情形下，$P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$ 不再从 $[0,1]$ 范围中取值，而是从 2 个端点取值 (0 或 1)，即算法对样本数据集 $\boldsymbol{X}$ 进行硬聚类[20]：

$$P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\to r_{nk},\epsilon\to 0\tag{61}$$

其中，r_{nk} 的意义与前述 k 均值中的意义相同。具体地，当 $\epsilon\to 0$ 时，在公式 (60) 中，只要数据点 $\boldsymbol{x}_n$ 在簇 C_j 内，那么 $\|\boldsymbol{x}_n-\boldsymbol{\mu}_j\|^2$ 就是值最小的项，它将以最慢的速度趋近于 0，而其它非最小值项都以较快的速度趋近于 0。因此，综合考虑分子分母情况，$P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$ 的取值存在 2 种情形：因数据点 $\boldsymbol{x}_n$ 在簇 C_k 内而取值为 1，或因数据点 $\boldsymbol{x}_n$ 不在簇 C_k 内而取值为 0。这意味着，在极限 $\epsilon\to 0$ 的情况下，

[19]具体的公式推导，请阅读《机器学习》课程系列之“EM 算法”。

[20]考虑一个协方差矩阵为 $\boldsymbol{\Sigma}=\epsilon\boldsymbol{I}$ 且 $\epsilon\to 0$ 的高斯分布，其各个维度上的方差将逐步减少并趋于 0。这样，其函数图像就形成一个类似于脉冲信号的超曲面图。此时，该分布在除了均值以外其它所有点的函数值均等于 0，但其在整个定义域上的积分仍然为 1。这种极限情形下形成的函数，可以被认为是狄拉克 δ 函数 (Dirac Delta Function) 的近似。但是，需要注意的是，这样的函数是通过极限方式构造的，因而严格来说并不是狄拉克 δ 函数本身。然而，在一些数学计算中，可以将这样的函数作为狄拉克 δ 函数来使用，并参与计算。

$P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \to r_{nk}$，且按如下的方式取值：

$$r_{nk} = \begin{cases} 1,\ if\ k = \arg\min\limits_{j=1,\cdots,K} \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0,\ else \end{cases} \tag{62}$$

因此，在极限 $\epsilon \to 0$ 情形时，高斯混合模型 EM 算法将每个样本数据点分配到最近的聚类中心，因而生成了一个硬聚类。这与 k 均值算法中对每个样本数据点的 r_{nk} 的解释完全一致。

另一方面，在高斯混合模型 EM 算法的 M 步，每个高斯分布的权重参数 α_k^{t+1} 按如下方式进行更新：

$$\alpha_k^{t+1} = \frac{\sum\limits_{n=1}^{N} P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t)}{N} \tag{63}$$

当 $\epsilon \to 0$ 时，可以将 α_k^{t+1} 解释为簇 C_k 中样本数据点占整个样本数据点的比例。然而，需要注意的是，每个高斯分布的权重参数 α_k^{t+1} 将不再起作用——考察公式 (60)，不在簇 C_k 内的数据点 $\boldsymbol{x}_n$，因较快地趋于 0 而使得自己的权重参数失去作用；而在簇 C_k 内的数据点 $\boldsymbol{x}_n$，也因较慢地趋于 0 而使得该公式的分子分母项相同而相互约去。

在 M 步，聚类中心 $\boldsymbol{\mu}_k^{t+1}$ 的更新公式为：

$$\boldsymbol{\mu}_k^{t+1} = \frac{\sum\limits_{i=1}^{N} P(k|\boldsymbol{x}_i;\boldsymbol{\theta}^t)\boldsymbol{x}_i}{\sum\limits_{i=1}^{N} P(k|\boldsymbol{x}_i;\boldsymbol{\theta}^t)} \tag{64}$$

当 $\epsilon \to 0$ 时，由于 $P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \to r_{nk}$(取值 0 或 1)，则可以将聚类中心 $\boldsymbol{\mu}_k^{t+1}$ 解释为簇 C_k 中所有样本数据点的均值。这也与 k 均值算法中对聚类中心的解释完全一致。

此外，考察高斯混合模型 EM 算法中的 Q 函数：

$$\begin{aligned} &Q(\boldsymbol{\theta}^{t+1}, \boldsymbol{\theta}^t) \\ &\propto \sum_{n=1}^{N}\sum_{k=1}^{K} P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \left[\log \alpha_k^{t+1} - \frac{1}{2}\log|\boldsymbol{\Sigma}_k^{t+1}| - \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{t+1})^T(\boldsymbol{\Sigma}_k^{t+1})^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{t+1})\right] \end{aligned} \tag{65}$$

将 $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}$ 代入，同时去掉不起作用的权重参数 α_k^{t+1} 及其它常数项[21]，则对 Q 函

[21] 因假设 $\boldsymbol{\Sigma}_k$ 不随时间变化，可忽略掉时间步 t 因素。另外，常数项对优化不起作用。

数取反，可以得到：

$$-Q(\boldsymbol{\theta}^{t+1},\boldsymbol{\theta}^{t}) \propto \sum_{n=1}^{N}\sum_{k=1}^{K} P(k|\boldsymbol{x}_n;\boldsymbol{\theta}^{t})\|\boldsymbol{x}_n-\boldsymbol{\mu}_k^{t+1}\|^2 \tag{66}$$

当 $\epsilon \to 0$ 时，上式可以写为：

$$-Q(\boldsymbol{\theta}^{t+1},\boldsymbol{\theta}^{t}) \propto \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\boldsymbol{x}_n-\boldsymbol{\mu}_k^{t+1}\|^2 \tag{67}$$

可以发现，上式与 k 均值算法的目标函数 (公式 (49)) 完全一致。

综上所述，k 均值算法是高斯混合模型 EM 算法的特例——在前者的每次迭代中，第 1 阶段与第 2 阶段分别对应 EM 算法中的 E 步与 M 步。

从前面的讨论可以看出，基本的 k 均值算法并没有估计簇的协方差，而只是对簇的均值进行了估计。关于协方差估计的 k 均值，请参考椭圆 k 均值 (Elliptical k-Means) 算法。

5 练习

1. 使用鸢尾花数据集实现并测试 k 均值算法；

6 附录

6.1 Mahalanobis 距离变换的演示

```
# -*- coding: utf-8 -*-
"""
Created on Fri Oct  2 22:59:29 2020
@author: duxiaoqin
@Functions:
    (1)Ellipse demo for standardization visualization:
z=(x-mean)/sigma;
"""
from math import *
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets.samples_generator import make_blobs

def get_ellipse(e_x, e_y, a, b, e_angle):
    angles_circle = np.arange(0, 2 * np.pi, 0.01)
    x = []
    y = []
    for angles in angles_circle:
        or_x = a * cos(angles)
        or_y = b * sin(angles)
        length_or = sqrt(or_x * or_x + or_y * or_y)
```

```
23          or_theta = atan2(or_y, or_x)
24          new_theta = or_theta + e_angle/180*np.pi
25          new_x = e_x + length_or * cos(new_theta)
26          new_y = e_y + length_or * sin(new_theta)
27          x.append(new_x)
28          y.append(new_y)
29      return x, y
30
31  XCENTER, YCENTER = 10, 20
32  X0, X1 = get_ellipse(XCENTER, YCENTER, 10, 20, 35)
33  N_SAMPLES = len(X0)
34  X0 = np.array(X0)
35  X1 = np.array(X1)
36  X = np.c_[X0, X1]
37  print(X.shape)
38
39  # 数据点均值: 按维度/特征计算
40  MEAN = np.mean(X, axis = 0)
41  print('MEAN', MEAN)
42  TEST_MEAN0, TEST_MEAN1 = np.sum(X[:, 0])/N_SAMPLES, np.sum(X[:,
    1])/N_SAMPLES
43  print('TEST MEAN', TEST_MEAN0, TEST_MEAN1)
44
45  # 数据点标准偏差: 按维度/特征计算
46  STD = np.std(X, axis = 0, ddof = 1)
47  print('STD', STD)
48  TEST_STD0 = np.sqrt(np.sum((X[:, 0] - TEST_MEAN0)**2) / (N_SAMPLES -
    1))
49  TEST_STD1 = np.sqrt(np.sum((X[:, 1] - TEST_MEAN1)**2) / (N_SAMPLES -
    1))
50  print('TEST STD', TEST_STD0, TEST_STD1)
51
52  # 数据点协方差矩阵: 按维度/特征计算
53  COV = np.cov(X.T)
54  print('COV', COV)
55  COV00 = np.sum((X[:, 0] - TEST_MEAN0)**2) / (N_SAMPLES - 1)
56  COV11 = np.sum((X[:, 1] - TEST_MEAN1)**2) / (N_SAMPLES - 1)
57  COV01 = np.sum((X[:, 0] - TEST_MEAN0)*(X[:, 1] - TEST_MEAN1)) /
    (N_SAMPLES - 1)
58  print('COV00', COV00)
59  print('COV01', COV01)
60  print('COV11', COV11)
61
62  # 数据点的标准偏差: 按维度计算, 应该与前面的结果相同
63  print('STD of COV00', np.sqrt(COV00))
64  #print('STD of COV01', np.sqrt(COV01))
65  print('STD of COV11', np.sqrt(COV11))
66
67  # 计算协方差矩阵的逆矩阵的特征值与特征向量
68  # 请参考《机器学习》课程系列之“高斯分布”: n 元高斯分布的几何解释
69  INV_COV = np.linalg.inv(COV)
70  EIGENVALUES, EIGENVECTORS = np.linalg.eig(INV_COV)
71  print('Eigenvalues', EIGENVALUES)
72  print('Eigenvectors\n', EIGENVECTORS)
73  LAMBDA = np.diag(EIGENVALUES)
74  print('Diagonal of Eigenvalues\n', LAMBDA)
75  print('INV_COV\n', INV_COV)
76  TEST_INV_COV = np.dot(EIGENVECTORS, np.dot(LAMBDA, EIGENVECTORS.T))
77  print('TEST_INV_COV\n', TEST_INV_COV)
78
79  # 对数据集 X 进行去相关性 + 中心化 + 缩放操作 => 某种形式的标准化操作
80  # 请参考《机器学习》课程系列之“高斯分布”: n 元高斯分布的几何解释
81  # 中心化 + 旋转
82  X_NEW = np.dot(EIGENVECTORS.T, (X - MEAN).T).T
83  # 中心化 + 旋转 + 缩放
```

```
X_NEW_NEW = np.dot(np.sqrt(LAMBDA), np.dot(EIGENVECTORS.T, (X - MEAN).T)).T

# 几个特殊的标记点
MARKER0 = np.array([XCENTER, YCENTER]).reshape(-1, 1)
MARKER1 = np.array([X[157, 0], X[157, 1]]).reshape(-1, 1)
ANGLE = -np.pi/2
ROTATE = np.array([[cos(ANGLE), -sin(ANGLE)],[sin(ANGLE), cos(ANGLE)]])
MARKER2 = np.dot(ROTATE, MARKER1-MARKER0)+MARKER0

# 绘制原图及标记点
plt.figure(1)
plt.plot(X[:, 0], X[:, 1], c = 'red')
plt.scatter(MARKER0[0, 0], MARKER0[1, 0], s = 40, c = 'red', marker = 'o')
plt.scatter(MARKER1[0, 0], MARKER1[1, 0], s = 40, c = 'red', marker = 'o')
plt.annotate('$p_{1}$', xy = (MARKER1[0, 0], MARKER1[1, 0]), xytext = (MARKER1[0, 0]-1, MARKER1[1, 0]+1), color = 'red')
plt.scatter(MARKER2[0, 0], MARKER2[1, 0], s = 40, c = 'red', marker = 'o')
plt.annotate('$p_{2}$', xy = (MARKER2[0, 0], MARKER2[1, 0]), xytext = (MARKER2[0, 0]-1, MARKER2[1, 0]+1), color = 'red')
plt.plot([MARKER0[0, 0], MARKER1[0, 0]], [MARKER0[1, 0], MARKER1[1, 0]], linestyle=':', c = 'red')
plt.plot([MARKER0[0, 0], MARKER2[0, 0]], [MARKER0[1, 0], MARKER2[1, 0]], linestyle=':', c = 'red')
plt.gca().set_aspect('equal', adjustable='box')

# 中心化 + 旋转
MARKER0_NEW = np.dot(EIGENVECTORS.T, MARKER0 - MEAN.reshape(-1, 1))
MARKER1_NEW = np.dot(EIGENVECTORS.T, MARKER1 - MEAN.reshape(-1, 1))
MARKER2_NEW = np.dot(EIGENVECTORS.T, MARKER2 - MEAN.reshape(-1, 1))
plt.figure(2)
plt.plot(X_NEW[:, 0], X_NEW[:, 1], c = 'blue')
plt.scatter(MARKER0_NEW[0, 0], MARKER0_NEW[1, 0], s = 40, c = 'blue', marker = 'o')
plt.scatter(MARKER1_NEW[0, 0], MARKER1_NEW[1, 0], s = 40, c = 'blue', marker = 'o')
plt.annotate('$p_{1}$', xy = (MARKER1_NEW[0, 0], MARKER1_NEW[1, 0]), xytext = (MARKER1_NEW[0, 0]-1, MARKER1_NEW[1, 0]+1.2), color = 'blue')
plt.scatter(MARKER2_NEW[0, 0], MARKER2_NEW[1, 0], s = 40, c = 'blue', marker = 'o')
plt.annotate('$p_{2}$', xy = (MARKER2_NEW[0, 0], MARKER2_NEW[1, 0]), xytext = (MARKER2_NEW[0, 0]-1, MARKER2_NEW[1, 0]+1.2), color = 'blue')
plt.plot([MARKER0_NEW[0, 0], MARKER1_NEW[0, 0]], [MARKER0_NEW[1, 0], MARKER1_NEW[1, 0]], linestyle=':', c = 'blue')
plt.plot([MARKER0_NEW[0, 0], MARKER2_NEW[0, 0]], [MARKER0_NEW[1, 0], MARKER2_NEW[1, 0]], linestyle=':', c = 'blue')
plt.gca().set_aspect('equal', adjustable='box')

# 中心化 + 旋转 + 缩放
MARKER0_NEW_NEW = np.dot(np.sqrt(LAMBDA), np.dot(EIGENVECTORS.T, MARKER0 - MEAN.reshape(-1, 1)))
MARKER1_NEW_NEW = np.dot(np.sqrt(LAMBDA), np.dot(EIGENVECTORS.T, MARKER1 - MEAN.reshape(-1, 1)))
MARKER2_NEW_NEW = np.dot(np.sqrt(LAMBDA), np.dot(EIGENVECTORS.T, MARKER2 - MEAN.reshape(-1, 1)))
plt.figure(3)
plt.plot(X_NEW_NEW[:, 0], X_NEW_NEW[:, 1], c = 'green')
```

```
126 plt.scatter(MARKER0_NEW_NEW[0, 0], MARKER0_NEW_NEW[1, 0], s = 40, c =
    'green', marker = 'o')
127 plt.scatter(MARKER1_NEW_NEW[0, 0], MARKER1_NEW_NEW[1, 0], s = 40, c =
    'green', marker = 'o')
128 plt.annotate('$p_{1}$', xy = (MARKER1_NEW_NEW[0, 0],
    MARKER1_NEW_NEW[1, 0]), xytext = (MARKER1_NEW_NEW[0, 0]-0.1,
    MARKER1_NEW_NEW[1, 0]+0.09), color = 'green')
129 plt.scatter(MARKER2_NEW_NEW[0, 0], MARKER2_NEW_NEW[1, 0], s = 40, c =
    'green', marker = 'o')
130 plt.annotate('$p_{2}$', xy = (MARKER2_NEW_NEW[0, 0],
    MARKER2_NEW_NEW[1, 0]), xytext = (MARKER2_NEW_NEW[0, 0]-0.1,
    MARKER2_NEW_NEW[1, 0]+0.09), color = 'green')
131 plt.plot([MARKER0_NEW_NEW[0, 0], MARKER1_NEW_NEW[0, 0]],
    [MARKER0_NEW_NEW[1, 0], MARKER1_NEW_NEW[1, 0]], linestyle=':', c =
    'green')
132 plt.plot([MARKER0_NEW_NEW[0, 0], MARKER2_NEW_NEW[0, 0]],
    [MARKER0_NEW_NEW[1, 0], MARKER2_NEW_NEW[1, 0]], linestyle=':', c =
    'green')
133 plt.gca().set_aspect('equal', adjustable='box')
134 plt.show()
```

7 参考文献

1. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, The MIT Press, 2012.

3. 李航。《统计学习方法》，清华大学出版社，2019 年 5 月第 2 版。

4. 周志华。《机器学习》，清华大学出版社，2016 年 1 月第 1 版。

5. 常用聚类算法。https://zhuanlan.zhihu.com/p/104355127.

6. Mahalonobis Distance - Understanding the math with examples (python). https://www.machinelearningplus.com/statistics/mahalanobis-distance/.

7. Feature Scaling. https://en.wikipedia.org/wiki/Feature_scaling.

8. Standard Score. https://en.wikipedia.org/wiki/Standard_score.

9. Normalization(statistics). https://en.wikipedia.org/wiki/Normalization_(statistics).

10. 标准化和归一化的区别？. https://www.zhihu.com/question/20467170.

11. 标准化和归一化:透彻理解数据变换. https://blog.csdn.net/weixin_36604953/article/details/102652160.

12. Cosine Similarity -Understanding the math and how it works (with python codes). https://www.machinelearningplus.com/nlp/cosine-similarity/.

13. 余弦相似性. https://zh.wikipedia.org/wiki/余弦相似性.

14. Cosine Similarity. https://en.wikipedia.org/wiki/Cosine_similarity.

15. Understanding Correlation. https://www.hawaii.edu/powerkills/uc.htm.

16. Pearson correlation coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient.

17. Correlation coefficient. https://en.wikipedia.org/wiki/Correlation_coefficient.

18. Coefficient of determination. https://en.wikipedia.org/wiki/Coefficient_of_determination.

19. 决定系数. https://zh.wikipedia.org/wiki/决定系数。

20. Simple linear regression. https://en.wikipedia.org/wiki/Simple_linear_regression.

21. Residual sum of squares. https://en.wikipedia.org/wiki/Residual_sum_of_squares.

22. Scatter Matrix. https://en.wikipedia.org/wiki/Scatter_matrix.

23. Sample mean and covariance. https://en.wikipedia.org/wiki/Sample_mean_and_covariance.

24. k-means clustering. https://en.wikipedia.org/wiki/K-means_clustering.

25. Voronoi diagram. https://en.wikipedia.org/wiki/Voronoi_diagram.

26. Voronoi 图。https://dsa.cs.tsinghua.edu.cn/ deng/cg/project/2000f/2000f-g.pdf.

27. 狄拉克 δ 函数. https://baike.baidu.com/item/狄拉克 δ 函数/5760582.