第一章 聚类分析

将认识对象进行分类是人类认识世界的一种重要方法。如在生物学中，为了研究生物的演变，需要对生物进行分类，生物学家根据各种生物的特征，将它们归属于不同的界、门、纲、目、科、属、种之中。事实上，分门别类地对事物进行研究，要远比在一个混杂多变的集合中研究更清晰、明了和细致，这是因为同一类事物会具有更多的近似特性。

在企业的经营管理中，为了确定其目标市场，首先要进行市场细分。因为无论一个企业多么庞大和成功，它也无法满足整个市场的各种需求。而市场细分，可以帮助企业找到适合自己特色，并使企业具有竞争力的分市场，将其作为自己的重点开发目标。聚类分析作为一种定量方法，将从数据分析的角度，给出一个更准确、细致的分类工具。

聚类分析是一种将相似的对象归为同一组的常用方法，适用于从大量数据中寻找出一些潜在的、不同类型的固有结构，以便进行研究和理解。对样本进行分类称为 Q 型聚类（Qualitative Clustering），对指标进行分类称为 R 型聚类（Relational Clustering）。

1.1 Q 型聚类

1.1.1 样本的相似性度量

为了用数量化的方法对事物进行分类，就需要用数量化的方法描述事物之间的相似程度。若一群有待分类的样本点是 n 维的，则每个样本点可以看成是 $\mathbb{R}^n$ 空间中的一个点。相似程度的一种度量方式是用距离来度量。

在聚类分析中，对于定量变量，最常用的是 Minkowski 距离：

$$d_q(x,y)=\left[\sum_{k=1}^{n}|x_k-y_k|^q\right]^{\frac{1}{q}}$$

其中，$x=(x_1,x_2,\ldots,x_n)^T, y=(y_1,y_2,\ldots,y_n)^T$。下面分别给出 $q=1,2,\infty$ 三种距离的定义。

(1) 绝对值距离

$$d_1(x,y)=\sum_{k=1}^{n}|x_k-y_k|$$

(2) 欧式距离

$$d_2(x,y)=\left[\sum_{k=1}^{n}|x_k-y_k|^2\right]^{\frac{1}{2}}$$

(3) Chebyshev 距离

$$d_\infty(x,y)=\max_{1\le k\le n}|x_k-y_k|$$

在 Minkowski 距离中，最常用的是欧氏距离，它的主要优点是当坐标轴进行正交旋转时，欧氏距离是保持不变的。因此，如果对原坐标系进行平移和旋转变换，则变换后样本点间的距离和变换前完全相同。

在采用 Minkowski 距离时，一定要采用相同量纲的变量。如果变量的量纲不同，先进行数据的标准化处理，然后再计算距离。另外，还应尽可能地避免变量的多重相关性，以免信息重叠而片面强调某些变量的重要性。

为了克服 Minkowski 距离的这些缺点，引入如下的马氏距离：

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

其中，x,y 为总体 Z 样本观测值，Σ 为 Z 的协方差矩阵。

实际中 Σ 往往是不知道的，常常需要用样本协方差来估计。马氏距离对一切线性变换是不变的，故不受量纲的影响。

1.1.2 类与类间的相似性度量

给定两个样本类 G_1 和 G_2，有如下的定义。

(1) 最短距离法

$$D(G_1, G_2) = \min_{x_i \in G_1, y_j \in G_2} d(x_i, y_j)$$

(2) 最长距离法

$$D(G_1, G_2) = \max_{x_i \in G_1, y_j \in G_2} d(x_i, y_j)$$

(3) 重心法

$$D(G_1, G_2) = d(\overline{x}, \overline{y})$$

其中，$\overline{x}, \overline{y}$ 分别为 G_1 和 G_2 的重心。

(4) 类平均法

$$D(G_1, G_2) = \frac{1}{n_1 n_2} \sum_{x_i \in G_1} \sum_{y_j \in G_2} d(x_i, y_j)$$

其中，n_1, n_2 分别为 G_1, G_2 中的样本点个数。该值为 G_1, G_2 中中两两样本点距离的平均。

(5) 离差平方和法

$$D(G_1, G_2) = D_{12} - D_1 - D_2$$

其中，

$$D_1 = \sum_{x_i \in G_1} (x_i - \bar{x}_1)^T (x_i - \bar{x}_1), D_2 = \sum_{x_j \in G_2} (x_j - \bar{x}_2)^T (x_j - \bar{x}_2)$$

$$D_{12} = \sum_{x_k \in G_1 \bigcup G_2} (x_k - \bar{x})^T (x_k - \bar{x})$$

$\bar{x}_1, \bar{x}_2, \bar{x}$ 分别为 $G_1, G_2, G_1 \bigcup G_2$ 的平均。

离差平方和法最初是由 Ward 在 1936 年提出，后经 Orloci 等人 1976 年发展起来的，故又称为 Ward 方法。

1.2 最短距离法聚类

该法由 Florek 等人于 1951 年和 Sneath 于 1957 年引入，其基本思想是使用最短距离法来测量类与类之间的距离，又称为最近邻法。下面通过实例来给出该法聚类的步骤。

设有 5 个销售员 $w_1, w_2, \ldots, w_5$，他们的销售业绩由如下表的二维变量 (v_i, v_j) 所描述。

员工号	指标值 v1	指标值 v2
1	1	0
2	1	1
3	3	2
4	4	3
5	2	5

记第 i 个员工的销售业绩为：$(v_{i,1}, v_{i2})$。使用绝对值距离来测量点与点之间的距离，即：

$$d(w_i, w_j) = \sum_{k=1}^{2} |v_{ik} - v_{jk}|$$

进而可得距离矩阵：

$$A = \begin{pmatrix} 0 & 1 & 4 & 6 & 6 \\ & 0 & 3 & 5 & 5 \\ & & 0 & 2 & 4 \\ & & & 0 & 4 \\ & & & & 0 \end{pmatrix}$$

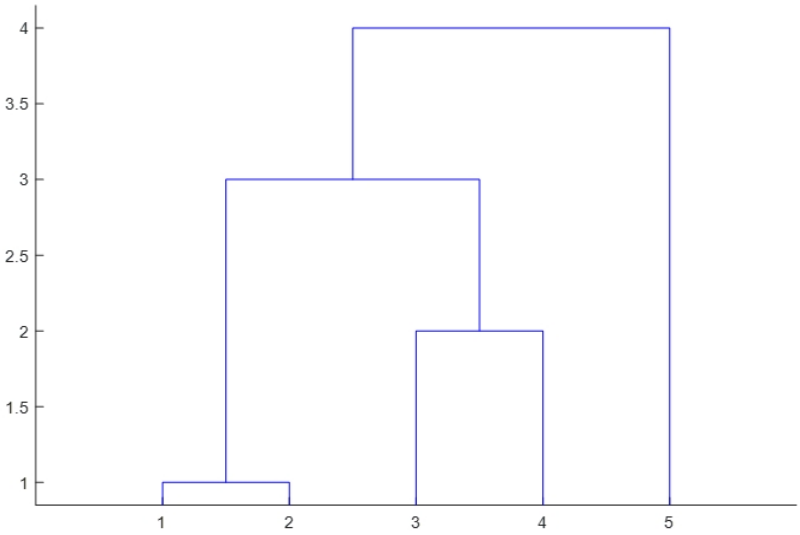
使用最短距离法来测量类与类之间的距离，即：

$$D(G_1, G_2) = \min_{x_i \in G_1, y_j \in G_2} d(x_i, y_j)$$

Step 1. 取新类的平台高度为 0，此时的分类：$H_1 = \{w_1, w_2, w_3, w_4, w_5\}$。
Step 2. 取新类的平台高度为 1，可将 w_1, w_2 合成一个新类 h_6，此时的分类：$H_2 = \{h_6, w_3, w_4, w_5\}$
Step 3. 取新类的平台高度为 2，可将 w_3, w_4 合成一个新类 h_7，此时的分类：$H_3 = \{h_6, h_7, w_5\}$。
Step 4. 取新类的平台高度为 3，可将 h_6, h_7 合成一个新类 h_8，此时的分类：$H_4 = \{h_8, w_5\}$。
Step 5. 取新类的平台高度为 4，可将 h_8 和 w_5 合成一个新类 h_9，此时的分类：$H_5 = \{h_9\}$。

待所有的样本点聚为一类，可以画出聚类图。可以看出，w_3 和 w_4 的业绩较好，而 w_1 和 w_2 的业绩较差。

```
clc
clear
close all
A = [1, 0; 1, 1; 3, 2; 4, 3; 2, 5];
y = pdist(A, 'cityblock');      % 求 a 的两两行向量间的绝对值距离
yc = squareform(y)              % 变换成距离方阵
z = linkage(y)                  % 产生等级聚类树
dendrogram(z)                   % 画聚类图
T = cluster(z, 'maxclust', 3)       % 把对象划分成 3 类
```



1.3 k-means 聚类算法

k-means 聚类是一种基于质心的聚类算法，其过程如下：

首先随机选取 k 个点作为质心；然后对于每个点，计算其到 k 个质心的距离，将该点归为距离最近的质心所在的簇；接着重新计算每个簇的质心；重复以上两步操作，直到质心不再发生变化或达到最大迭代次数。

Step 1. 随机选取 k 个点作为质心。
Step 2. 对于每个点，计算其到 k 个质心的距离，将该点归为距离最近的质心所在的簇。
Step 3. 重新计算每个簇的质心。
Step 4. 重复步骤 2 和 3，直到质心不再发生变化或达到最大迭代次数。

算法的适用场景：当簇与簇之间区别明显时，用该算法的效果较好。

算法的优点：计算简单、速度较快。

算法的缺点：对初始质心的选择较为敏感，容易陷入局部最优解；对于不是凸的数据集比较难收敛；算法容易受数据不均横、异常点的影响。

1.4 K-Means 聚类算法的改进

(1) k-means++：对 k-means 随机初始化质心的方法进行了优化，其目的时让选择的质心尽可能的分散。

(2) 二分 k-means：类似于决策树的思想，通过误差平方和设置阈值，然后进行划分。聚类的误差平方和能够衡量聚类性能，该值越小，聚类效果越好。

(3) elkan k-means：该算法利用了两边之和大于等于第三步，以及两边之差小于第三边的性质减少了不必要的距离计算。

(4) mini batch k-means：该法适用于样本量大样本的情形。该法不需要使用所有的数据样本，而是从不同类别的样本中抽取部分样本进行计算，减少了运行时间，但准确度也会随之下降。

(5) k-medoids：该法采用欧氏距离来衡量某个样本点到底是属于哪个类簇，适合于小样本的情形。与 k-means 算法的区别在于中心点的选取：k-means 算法选取当前簇中所有数据点的平均值为中心点，对异常点会非常敏感；而 k-medoids 算法是从当前簇中选取到其他所有（当前簇中的）点的距离之和最小的点作为中心点。

(6) k-center：寻找 k 个半径越小越好的 center 以覆盖所有的点。

(7) k-modes：属性相同为 0，不同为 1，并所有相加。使用一个簇的每个属性出现频率最大的那个属性值作为代表簇的属性值。

(8) kernel k-means：参照 SVM 中核函数的思想，将所有样本映射到另外一个特征空间中再进行聚类。

1.5 层次聚类算法

层次聚类算法是一种自底向上或自顶向下的聚类方法，其过程如下。

Step 1. 对于每个样本，将其视为一个独立的簇。
Step 2. 计算两两样本之间的相似度或距离，根据相似度或距离构建一个树形结构，即聚类树。
Step 3. 不断合并聚类树中距离最小的两个簇，直至所有样本被合并为一个簇或达到某个预设的簇的数量。

算法的应用场景：（1）适合于小型数据集的聚类；（2）不清楚数据集可以聚成几类的情况下，层次聚类可以在不同梯度水平上对数据进行探测，从而能发现类之间的层次关系。

算法的优点：不需要事先确定聚类的数目，且可视化效果好。

算法的缺点：计算复杂度高，适用于样本量较小的情况。

1.6 DBSCAN 聚类算法

DBSCAN 聚类算法是一种基于密度的聚类方法，其过程如下。

Step 1. 对于每个样本，计算其在指定半径 r 内的样本数量，将密度大于某个阈值的样本视为核心样本。
Step 2. 将所有核心样本连接起来，构成一个簇。
Step 3. 对于所有不是核心样本但与核心样本距离在 r 范围内的样本，将其归为与其最近的核心样本所在的簇。
Step 4. 重复以上步骤直到所有样本被归类。

算法的优点：能够处理任意形状的簇，并且能够识别噪声数据。
算法的缺点：对距离度量的选择敏感。

1.7 R 型聚类

在系统分析或评估过程中，通常会尽可能多地考虑相关因素而选取指标。这样会导致变量过多，变量间的相关度高，从而给系统分析与建模带来很大的不便。为了找出影响系统的主要因素，需要研究变量间的相似关系，按照变量的相似关系把它们聚合成若干类。

变量相似性度量：常用的变量相似性度量有如下两种。

(1) 相关系数

$$r_{ij}=\frac{\sum_{k=1}^{n}(x_{ik}-\bar{x}_i)(x_{jk}-\bar{x}_j)}{\left[\sum_{k=1}^{n}(x_{ik}-\bar{x}_i)^2\sum_{k=1}^{n}(x_{jk}-\bar{x}_j)^2\right]^{\frac{1}{2}}}$$

(2) 夹角余弦

$$\cos\theta=\frac{\sum_{k=1}^{n}x_{ik}x_{jk}}{\left(\sum_{k=1}^{n}x_{ik}^2\sum_{k=1}^{n}x_{jk}^2\right)^{\frac{1}{2}}}$$

在对变量进行聚类分析时，大多数情形下用相关系数矩阵 $R=(r_{ij})_{n\times n}$。

两种定义下，$|r_{ij}|$ 越接近 1，x_i 与 x_j 越相关或越相似；值 $|r_{ij}|$ 越接近零，相似性越弱。

变量聚类法：采用了与系统聚类法相同的思路和过程。在变量聚类问题中，常用的有最长距离法、最短距离法等。给定两个样本类 G_1 和 G_2，有如下的定义。

(1) 最短距离法

$$R(G_1,G_2)=\min_{x_i\in G_1,y_j\in G_2}d_{ij}$$

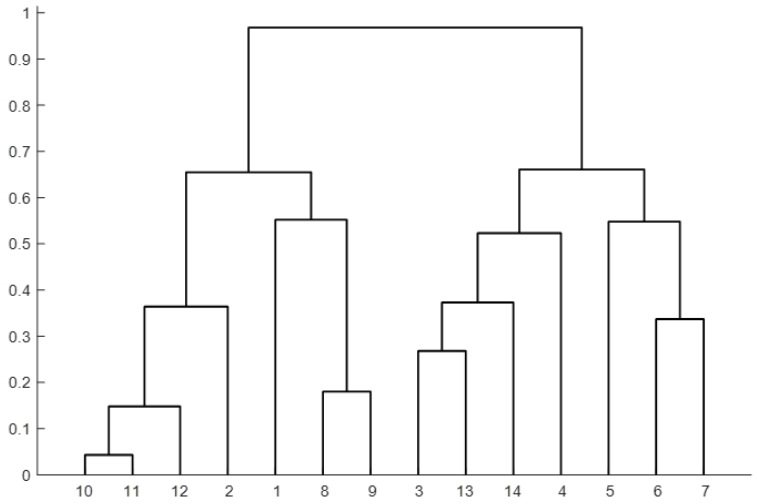
其中，$d_{ij}=1-|r_{ij}|$ 或 $d_{ij}^2=1-r_{ij}^2$。

(2) 最长距离法

$$R(G_1,G_2)=\max_{x_i\in G_1,y_j\in G_2}d_{ij}$$

例题 1.1 在服装标准制定中，对某地成人的 14 个指标进行了统计，得到了各因素之间的相关系数表（见附件）。用最长距离法对这 14 个变量进行系统聚类。

聚类的结果如下。



代码如下。

```
clc
clear
close all
A=xlsread('Data_Julei_R');
A(isnan(A))=0;
D=1-abs(A);      % 进行数据变换, 把相关系数转化为距离
D=tril(D);           % 提出矩阵的下三角部分
B=nonzeros(D);       % 去掉 d 中的零元素
B=B';
z=linkage(B, 'complete');       % 按最长距离法聚类
y=cluster(z, 'maxclust',2)       % 把变量划分成两类
h=dendrogram(z);       % 画聚类图
```