机器学习讲义 (L19) 聚类

授课教师：王贝伦 / 助教：张嘉琦，黄旭，谈笑，徐浩卿

在之前的学习中，我们学习了主成分分析（PCA）这种降维方法，在这一章节中，我们将学习与降维方法同为无监督学习方法的聚类方法。

聚类是针对给定的样本，依据它们特征的相似度或距离，将其归并到若干个“类”或“簇”的数据分析问题。一个类是给定样本集合的一个子集。直观上，相似的样本聚集在相同的类，不相似的样本分散在不同的类。这里，样本之间的相似度或者距离起着重要作用，我们会在第二节介绍这部分内容。

聚类算法很多，基本可以分成两类：层次聚类（hierarchical clustering）和划分聚类（partitional clustering）。划分聚类通常以随机划分开始整个算法过程，随后进行迭代，如 k 均值聚类（k-means clustering）和基于混合模型的聚类（Mixture-Model based clustering）。层次聚类有聚合（自下而上）和分裂（自上而下）两种方式，在这次讲义内容中我们将着重介绍层次聚类中的聚合方法。

1 聚类方法的简单介绍

聚类（clustering）是将样本集合中相似的样本（实例）分配到不同的类。聚类时，样本通常是欧式空间中的向量，类别不是事先给定，而是从数据中自动发现，但类别的个数通常是事先给定的。样本之间的相似度或者距离由应用决定。

我们通常说的聚类一般是指硬聚类（hard clustering），即一个样本只能属于一个类；但是实际上一个样本也可以属于多个类，这就是软聚类（soft clustering）。

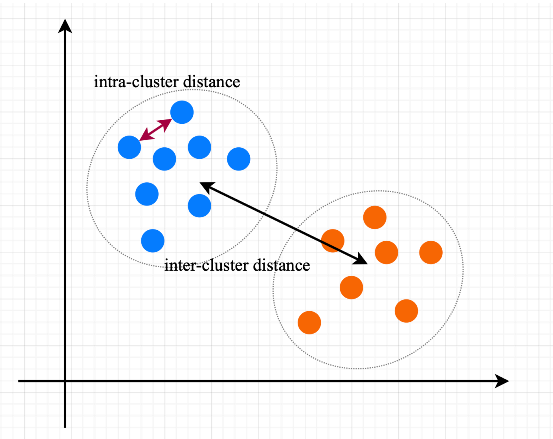

Figure 1: 聚类

如上图所示，在一个聚类内样本与样本之间的距离被称为类内距离（intra-cluster distance），而聚类与聚类之间的距离被称为类间距离（inter-cluster distance）。我们在划分聚类的时候都希望类间距离更大，类内距离更小，这样的模型确信度更高，这是我们学习聚类模型时追求的目标。

1.1 聚类算法的应用

聚类方法作为无监督学习最为常见的方法之一，适用范围非常广泛。如下图中的页面导航，这就是聚类方法中层次聚类的一种应用。

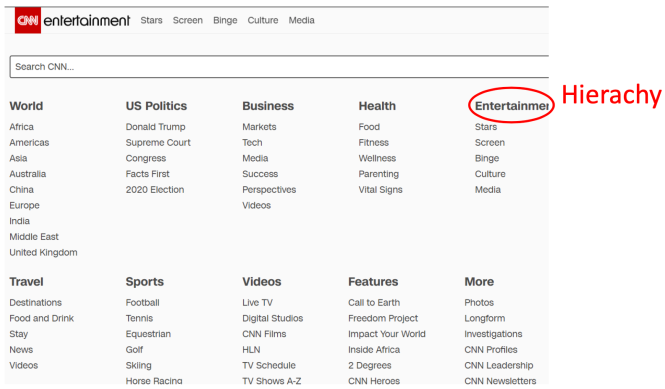

Figure 2: 页面导航

再比如我们在搜索引擎上搜索一个词时，会跳出来很多包含这个词的页面，这是聚类方法通过相似度的计算，把与这个词可能最相关的页面统统聚集起来呈现出来。

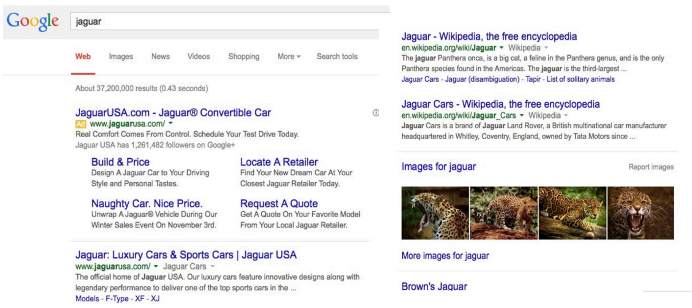

Figure 3: 搜索引擎的结果

2 相似度或距离

当我们在对样本进行分类前，首先我们要思考的时，我们是根据什么样本进行分类的。对于以下这个样本集，我们可以根据职业分类也可以根据性别分类，我们需要一个指标去衡量样本之间的相似度。这就引出了这一节的内容——相似度或距离。

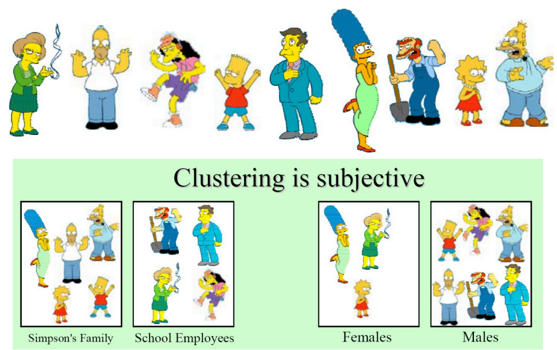

Figure 4: 如何划分《辛普森》动画人物？

聚类的核心概念是相似度（similarity）或距离（distance），有多种相似度或距离的定义。因为相似度直接影响聚类的结果，所以

其选择是聚类的根本问题。具体哪种相似度更合适取决于应用问题的特性。

2.1 闵可夫斯基距离（Minkowski distance）

在聚类中，可以将样本集合看做是向量空间中点的集合，以该空间的距离表示样本之间的相似度。常用的距离有 Minkowski 距离，特别是欧氏距离。Minkowski 距离越大相似度越小，距离越小相似度越大。

定义：闵可夫斯基距离

对于样本集合 X 中的两点 x 和 y，$x=(x_1,x_2,...,x_p), y=(y_1,y_2,...,y_p)$，这两点的 Minkowski 距离定义为

$$d(x,y)=(\sum_{i=1}^{p}|x_i-y_i|^r)^{\frac{1}{r}}, \quad r\geq 1 \tag{1}$$

当 $r=2$ 时称为欧式距离（Euclidean distance），即

$$d(x,y)=(\sum_{i=1}^{p}|x_i-y_i|^2)^{\frac{1}{2}} \tag{2}$$

当 $r=1$ 时称为曼哈顿距离（Manhattan distance），即

$$d(x,y)=\sum_{i=1}^{p}|x_i-y_i| \tag{3}$$

当 $r=\infty$ 时称为切比雪夫距离（Chebyshev distance），取各个坐标数值差的绝对值的最大值，即

$$d(x,y)=\max_i|x_i-y_i| \tag{4}$$

下面举一个例子说明这三种距离。

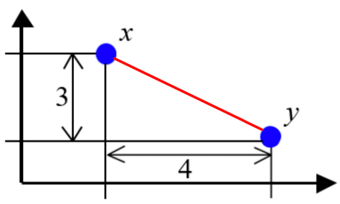

Figure 5: 例图

- 欧式距离：$d(x,y)=\sqrt{3^2+4^2}=5$
- 曼哈顿距离：$d(x,y)=4+3=7$
- 切比雪夫距离：$d(x,y)=max\{4,3\}=4$

2.2 相关系数

样本之间的相似度也可以用相关系数（correlation coefficient）来表示。相关系数的绝对值越接近于 1，表示样本越相似；越接近 0，表示样本越不相似。

定义：相关系数

样本 x 与样本 y 之间的相关系数定义为

$$r_{xy}=\frac{\sum_{i=1}^{p}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i-\bar{x})^2\sum_{i=1}^{p}(y_i-\bar{y})^2}} \tag{5}$$

其中

$$\bar{x}=\frac{1}{p}\sum_{i=1}^{p}x_i, \quad \bar{y}=\frac{1}{p}\sum_{i=1}^{p}y_i$$

这之前的课程内容中我们已经着重介绍过相关系数了，在这里不多做介绍。

2.3 夹角余弦（Cosine distance）

样本之间的相似度也可以用夹角余弦来表示。夹角余弦越接近 1，表示样本越相似；越接近于 0，表示样本越不相似。

定义：夹角余弦

样本 x 与样本 y 之间的夹角余弦定义为

$$s(x,y)=\frac{\sum_{i=1}^{p}x_iy_i}{\sqrt{\sum_{i=1}^{p}x_i^2\sum_{i=1}^{p}y_i^2}} \tag{6}$$

这个相似度衡量方法相对来说比较少见，但在信息检索和文本挖掘中有所应用。

3 层次聚类

层次结构在生活中十分常见。例如学校的人员管理结构、书籍的目录章节、计算机中的文件逻辑结构等，都是呈层次结构或称树状结构。在无监督学习中，也可以按层次结构的思路进行聚类。其基本思想是：距离较近的点应该属于同一类。

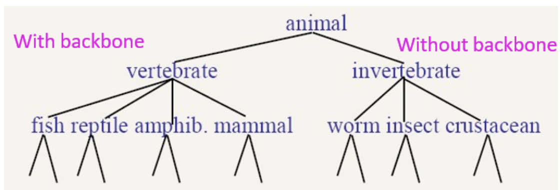

Figure 6: 层次聚类示意

我们下面介绍一种非常简单的层次聚类方法。

3.1 自底向上构建聚类

构建层次聚类的方式有自底向上、自顶向下两种，本节主要说明自底向上构建法。所谓自底向上，是指最初把每个点都认为是一个单独聚类，将两个最近的聚类聚合为一个更高层的聚类，重复这一操作直到最后所有点都聚为一类，这样就构建好了一颗树。接下来我们举个例子说明：

假设有 5 个数据点，我们可以计算出他们两两之间的距离。以欧式距离为例，如果我们得到如下的距离矩阵：

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
0 & & & & \\
2 & 0 & & & \\
6 & 3 & 0 & & \\
10 & 9 & 7 & 0 & \\
9 & 8 & 5 & 4 & 0
\end{bmatrix}
\end{array}
$$

Figure 7: 距离矩阵

除了自己与自己的距离固定为 0 以外，我们可以找出一对距离最近的点，那么这一对就合并为一个聚类。在本例中，显然 1、2 两个点应该合并为聚类 (1,2)。

接下来，我们应该再次计算各点的距离，选择距离最近的一对合并。这里引申出一个问题，2 个聚类之间的距离如何定义？常见的有 3 种定义方法：

- Single Link：两个不同聚类点的最短距离
- Complete Link：两个不同聚类点的最远距离
- Average：两个聚类中每对点距离的平均值

例如，用 Single Link 计算聚类的距离：

$$
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 3 & 0 & & \\
4 & 9 & 7 & 0 & \\
5 & 8 & 5 & 4 & 0
\end{array}
$$

Figure 8: Single Link 示意

其中，第一列的数字实际上是取 Figure 7中前两列各行的最小值，而 Average 则取均值。

按上述规则继续迭代，最后距离矩阵变为 1×1，构建结束。

$$
\begin{array}{c|ccc}
 & (1,2,3) & 4 & 5 \\
\hline
(1,2,3) & 0 & & \\
4 & 7 & 0 & \\
5 & 5 & 4 & 0
\end{array}
\qquad
\begin{array}{c|cc}
 & (1,2,3) & (4,5) \\
\hline
(1,2,3) & 0 & \\
(4,5) & 5 & 0
\end{array}
\qquad
\begin{array}{c|c}
 & (1,2,3,4,5) \\
\hline
(1,2,3,4,5) & 0
\end{array}
$$

Figure 9: Single Link 后续迭代

最后可以将构建好的聚类树画出来如下图。

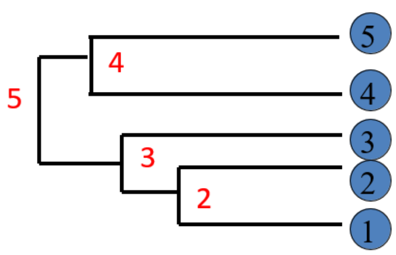

Figure 10: Single Link 聚类树

3.2 复杂度分析

- 计算两个聚类间的距离：需要计算两个向量之差的模，若维数为 p，总开销 $O(p)$
- 计算整个距离矩阵：距离矩阵里有 $\frac{1}{2}n(n-1)$ 个距离需要计算，每个 $O(p)$，则总共 $O(n^2p)$。
- 选取当前最近的两个聚类：遍历距离矩阵，$O(n^2)$

每次迭代只合并一次，聚类减少一个，共 $n-1$ 次迭代，总时间复杂度：$O(n^3p)$

3.3 讨论

- 相比于 k-means 聚类，不需要事先指定聚类个数，层次聚类会生成一个聚类树，之后可以方便的选择不同的聚类方式。
- 层次结构很好地贴合了一些特定的应用场景，例如人员管理、网站结构等，信息本来就是以树状结构组织的，用层次聚类就能较好地还原现实中的结构。
- 时间复杂度较高（为 $O(n^3p)$），在数据点较多时性能不佳。
- 自底向上构建是一种启发式算法，并没有最优性保证，也没有明确的待优化目标函数，得到的聚类树可能并不是很合理。

引用

[1] 'Statistical Learning' Hang Li