层次聚类：原理、公式、应用与实战

2025 年 9 月 17 日

1 引言

层次聚类通过不断合并或拆分簇来构建数据的多层次结构，无需预先指定簇数。常见的自底向上（凝聚式）方法先将每个样本视为独立簇，再依据链接准则（如最短距离、最长距离、平均距离或 Ward 距离）逐步合并最相近的簇。最终得到的树状图（dendrogram）记录了整个合并过程，可在不同高度切割以获得不同数量的簇。

2 原理与公式

2.1 链接函数

考虑两个簇 A 与 B，其样本分别为 $\mathbf{x}_i$ 与 $\mathbf{x}_j$，常见链接定义如下：

$$\text{single}(A,B) = \min_{\mathbf{x}_i \in A,\, \mathbf{x}_j \in B} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \tag{1}$$

$$\text{complete}(A,B) = \max_{\mathbf{x}_i \in A,\, \mathbf{x}_j \in B} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \tag{2}$$

$$\text{average}(A,B) = \frac{1}{|A|\,|B|} \sum_{\mathbf{x}_i \in A} \sum_{\mathbf{x}_j \in B} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \tag{3}$$

$$\text{Ward}(A,B) = \frac{|A|\,|B|}{|A| + |B|} \|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|_2^2, \tag{4}$$

其中 $\boldsymbol{\mu}_A$ 与 $\boldsymbol{\mu}_B$ 为对应簇的质心。Ward 链接在每次合并时最小化簇内方差的增量。

2.2 凝聚式算法

典型的凝聚式层次聚类流程为：

1. 将每个样本初始化为单独簇，并计算所有点对距离；
2. 重复选择链接距离最小的簇对并将其合并；
3. 按所选链接准则更新簇间距离矩阵；

4. 直至所有样本合并为一个簇，或达到设定的停止条件。

聚类过程形成的树状图记录了每次合并的高度；在高度 h 处切割树状图，可得到最大簇内不相似度不超过 h 的划分。

2.3 共表距离

共表距离指的是树状图中两个样本合并时的高度。将共表距离矩阵与原始距离矩阵进行相关性分析，可以评估层次结构对原始距离关系的保真度。

3 应用与技巧

- **分类与谱系分析**：树状图形象展示物种、文档等对象之间的层级关系。
- **商品篮分析**：对商品进行层次聚类，挖掘嵌套的组合模式，指导推荐与陈列。
- **特征构造**：在不同高度切割树状图，可得到多尺度聚类标签或用于其他算法的分组约束。
- **实用建议**：特征需标准化，选择与数据几何特性一致的链接方式，并注意链式效应（single 链接）或对离群点的敏感性（complete 链接）。

4 Python 实战

脚本 `gen_clustering_hierarchical_clustering_figures.py` 构造多个高斯簇并执行 Ward 链接的凝聚式聚类，输出树状图以及按三簇切割后的二维散点图。

Listing 1: 脚本 $\mathrm{gen}_{c}lustering_{h}ierarchical_{c}lustering_{f}igures.py$

```python
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

linkage_matrix = linkage(points, method="ward")
labels = fcluster(linkage_matrix, t=3, criterion="maxclust")

# 绘制树状图及依据切割结果上色的散点图
```

5 实验结果

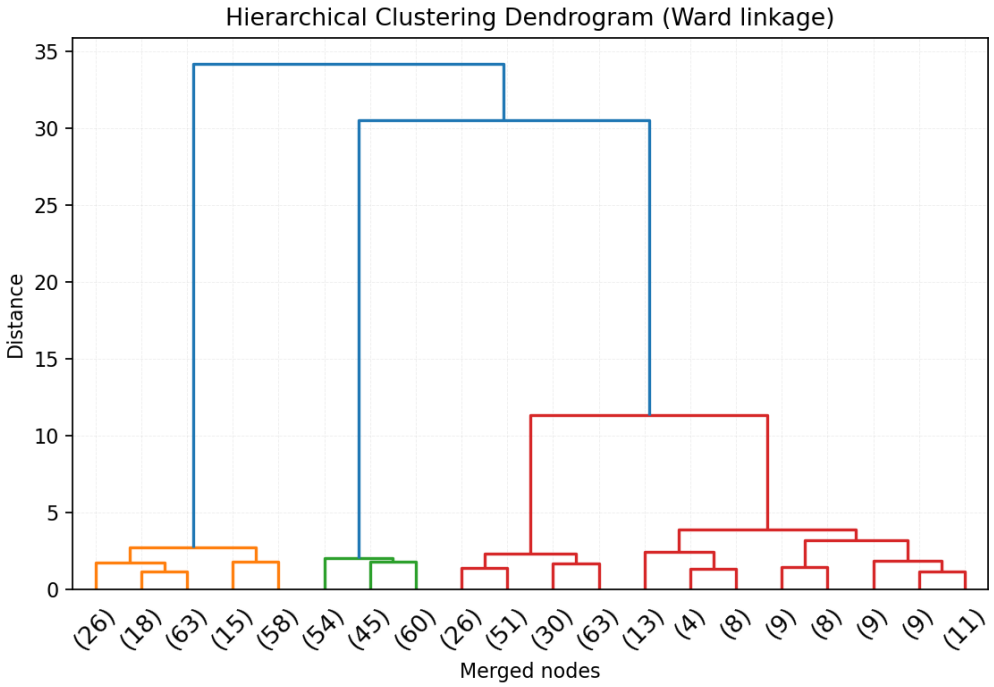

图 1: Ward 链接在合成数据上的树状图

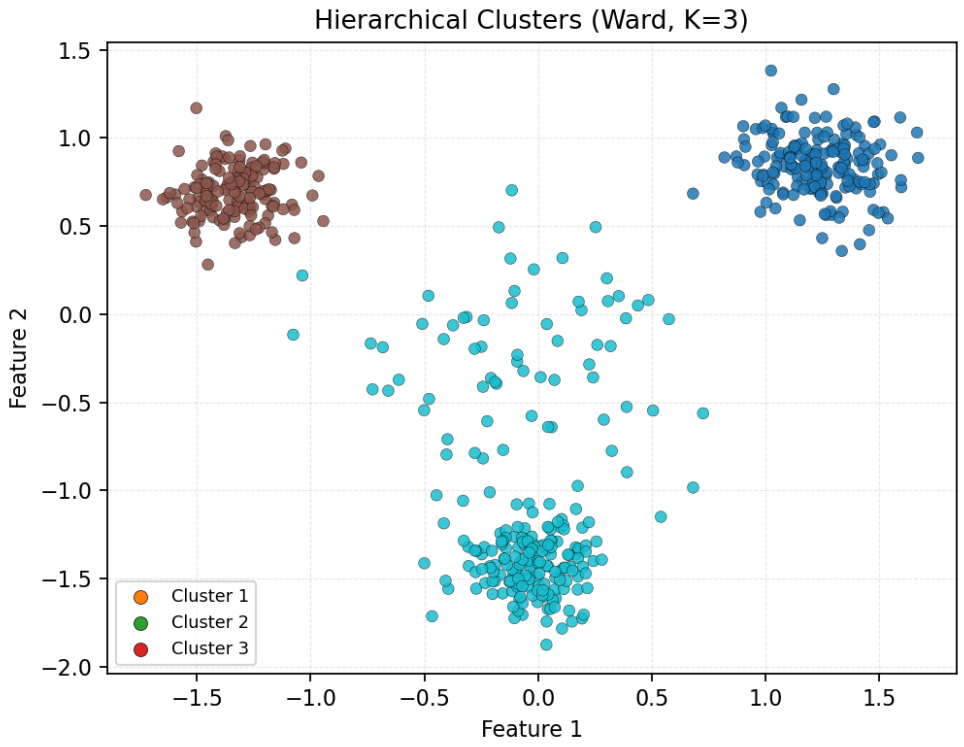

图 2: 将树状图切割为三个簇后的二维聚类可视化

6 总结

层次聚类无需事先指定簇数，适合探索数据的嵌套结构。不同链接方式会影响簇形态：Ward 偏向紧凑簇，average 在链式效应与紧致度之间折中，而 single、complete 强调连通性两端。示例展示了树状图与切割结果如何相辅相成地支持探索分析。