聚类算法学习笔记

一、聚类的基本概念

1. 什么是聚类?

- **定义**：将一组数据点根据其相似性（距离）划分到不同的组（簇），使得：
  - 同一簇内的数据点尽可能相似；
  - 不同簇的数据点尽可能不相似。
- **关键要素**：
  - **距离度量**：欧氏距离、余弦相似度、Jaccard距离等。
  - **高维挑战**：高维空间中，数据点之间的距离趋于相似，聚类更困难。

2. 应用场景

- **天文学**：通过辐射频段聚类天体（如星系、恒星）。
- **电商推荐**：根据用户购买记录聚类相似商品（如音乐CD）。
- **文本分类**：根据词频向量聚类相似主题的文档。

二、聚类方法

1. 层次聚类（Hierarchical Clustering）

- **核心思想**：通过逐步合并或分裂簇构建树状结构（树状图）。
- **两种策略**：
  - **自底向上（Agglomerative）**：
    - 初始每个点是一个簇；
    - 重复合并距离最近的两个簇，直到所有点合并为一簇。
  - **自顶向下（Divisive）**：
    - 初始所有点为一簇；
    - 递归分裂为更小的簇。
- **关键问题**：
  - **簇的表示**：
    - **质心（Centroid）**：簇内点的平均值（适用于欧氏空间）。
    - **簇核（Clustroid）**：簇内到其他点平均距离最小的点（适用于非欧氏空间）。
  - **簇间距离计算**：
    - 质心间距离、最小点间距离、最大点间距离等。

示例：
假设有4个点：A(1,2)、B(2,1)、C(4,1)、D(5,0)。

1. 初始每个点为一簇；
2. 合并距离最近的簇（如B和D）；

3. 更新质心，继续合并，最终形成树状图。

---

2. k-means算法

核心步骤：

1. **初始化**：随机选择k个初始质心。
2. **分配点**：将每个点分配到最近的质心所属簇。
3. **更新质心**：重新计算每个簇的质心。
4. **迭代**：重复步骤2-3，直到质心稳定。

示例：

- 数据点：[(1,1), (1,2), (2,1), (5,4), (6,5), (5,6)]，k=2。
  - 初始质心：随机选(1,1)和(5,4)；
  - 第一轮分配：前3点归簇1，后3点归簇2；
  - 更新质心：簇1质心=(1.33, 1.33)，簇2质心=(5.33, 5)；
  - 第二轮分配：点不变化，算法终止。

如何选择k值?

- **肘部法则（Elbow Method）**：
  - 绘制不同k值的平均距离（误差平方和SSE）曲线；
  - 选择SSE下降速度骤减的拐点（类似"肘部"）。

---

3. BFR算法（大规模数据处理）

- **适用场景**：数据量极大，无法全部加载到内存。
- **核心思想**：
  - 用三类集合管理数据点：
    - **DS（Discard Set）**：已分配到簇的点，用统计量（N, SUM, SUMSQ）压缩存储。
    - **CS（Compression Set）**：未分配但相互靠近的点组。
    - **RS（Retained Set）**：孤立点，暂未分配。
  - **Mahalanobis距离**：考虑各维度方差，判断点是否属于某簇。

示例：

- 若某簇在x轴的标准差为2，y轴为1，点P(3,2)到质心(1,1)的Mahalanobis距离为：
  $$ \sqrt{\left(\frac{3-1}{2}\right)^2 + \left(\frac{2-1}{1}\right)^2} = \sqrt{1 + 1} = \sqrt{2} $$
- 若阈值设为2，则P属于该簇。

---

4. CURE算法（任意形状聚类）

- **核心思想**：用多个代表点（而非单一质心）表示簇，适应非球形分布。
- **步骤**：

1. 对采样数据层次聚类；
2. 为每个簇选取分散的代表点，并向质心收缩（如20%）；
3. 全量扫描数据，将点分配到最近的代表点所属簇。

示例：

- 数据分布呈月牙形，k-means无法正确聚类。
- CURE选取多个代表点（如月牙两端），收缩后仍保留形状特征，最终正确划分簇。

三、总结对比

算法	适用场景	优点	缺点
层次聚类	小规模数据	可视化树状图，无需预设k	计算复杂度高（$O(N^2)$）
k-means	球形簇，内存数据	简单高效	需预设k，对噪声敏感
BFR	大规模高维数据	内存占用低，支持流式处理	仅适用于高斯分布簇
CURE	任意形状簇，非均匀分布	适应复杂形状	参数调整复杂，计算成本高

练习题

1. 对数据集[(2,3), (2,5), (3,4), (7,8), (8,8), (6,7)]，手动运行k-means（k=2），写出每一步的质心和簇分配。
2. 如何用肘部法则确定右图中的最佳k值？画出SSE随k变化的曲线示意图。

附：课件问题答案

Q：K-means算法中的K值如何确定？

A：

1. 使用肘部法则：绘制不同K值的误差平方和（SSE）曲线，选择拐点对应的K。
2. 实际示例：当K=3时，SSE下降速度明显变缓，因此选择K=3。