

2022-2023秋季课程:数据科学与大数据导论

Introduction to Data Science and Big data


Chapter 4: Big Data Analytics Algorithms

曹劲舟 助理教授

深圳技术大学 大数据与互联网学院

caojinzhou@sztu.edu.cn

2022年9月



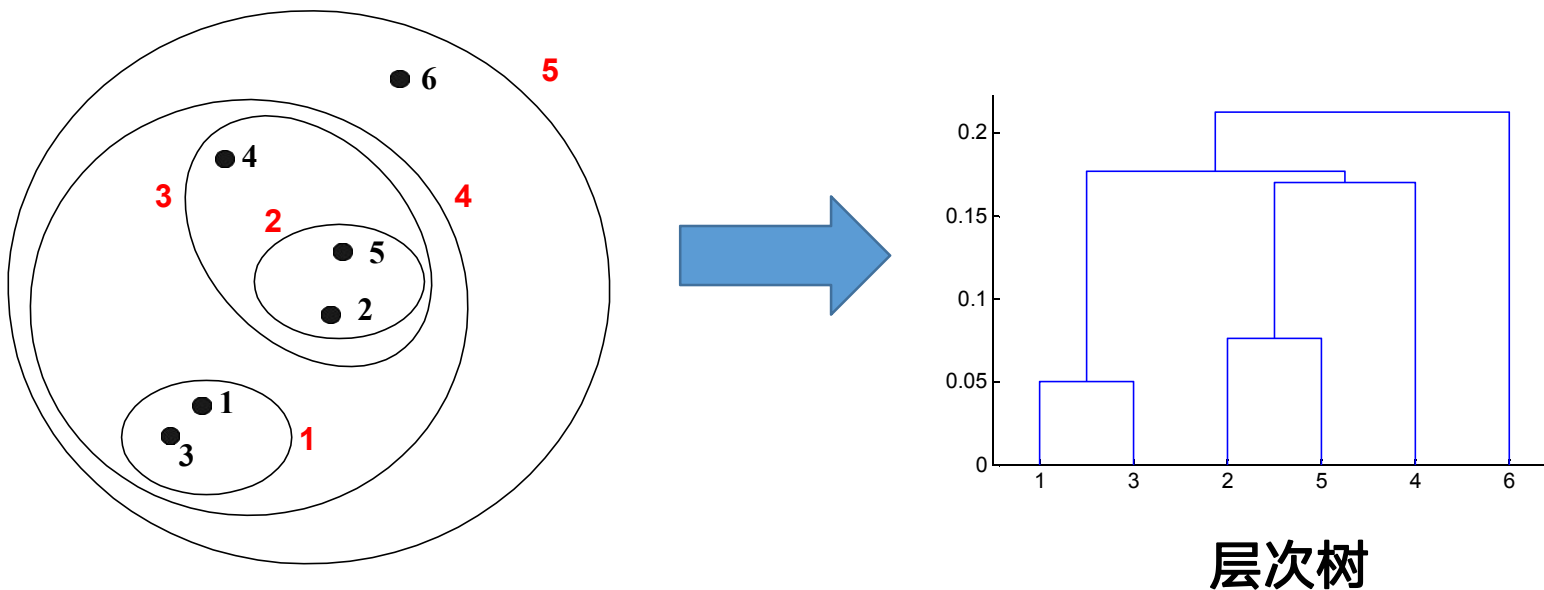
Outline

□ Unsupervised Learning 非监督学习-扩展学习

聚类分析：层次聚类

□ 层次聚类(Hierarchical Clustering)

- **特点：**生成一组嵌套的簇，表示为**层次树**
- 整体呈树状结构，除叶节点外，每个节点是子节点的并集
- 叶节点：一般为单个样本组成的单元簇
- 一种树状的图表，记录合并或拆分的顺序



聚类分析：层次聚类

- 层次聚类的优势

- 无需事先指定簇的个数

- 可以根据需要随时调节，只需要在相应的层数切分树状结构即可

- 层次聚类的结果往往可以对应到具有一定意义的分类学目录上

- 例如，可以对应到WordNet的层次结构

- 例如，犬类动物的层次结构

```
dog, domestic dog, Canis familiaris
=> canine, canid
=> carnivore
=> placental, placental mammal, eutherian, eutherian mammal
=> mammal
=> vertebrate, craniate
=> chordate
=> animal, animate being, beast, brute, creature, fauna
=> ...
```

犬 > 类犬动物 > 食肉动物 > 胎盘类 > 哺乳动物 > 哺乳动物 > 脊椎动物

聚类分析：层次聚类

- 层次聚类有以下两种基本形式

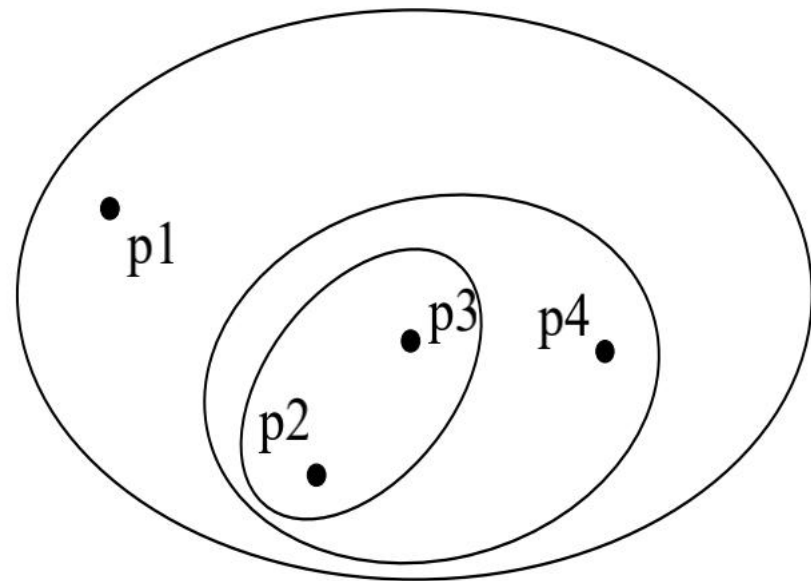
- 凝聚式聚类 (Agglomerative, 自下而上聚类)

- 1. 初始时，每个数据均视为一个簇
 - 2. 每一轮将最近的两个簇合并
 - 3. 直到只剩一个簇

- 分裂式聚类 (Divisive, 自上而下聚类)

- 1. 初始时，所有数据属于一个簇
 - 2. 每一轮将簇切分
 - 3. 直到每个簇仅包含一个样本

- 一般而言，**凝聚式聚类**更为常见



聚类分析：层次聚类

- 凝聚式层次聚类

- 引入 邻近度矩阵 的概念

- 用于存储两两簇之间的邻近度

- 基本流程非常直观，主要迭代以下两步，直到仅剩一个簇

- 1. 合并邻近度最高的两个簇
 - 2. 基于更新的簇重新计算距离，更新距离矩阵

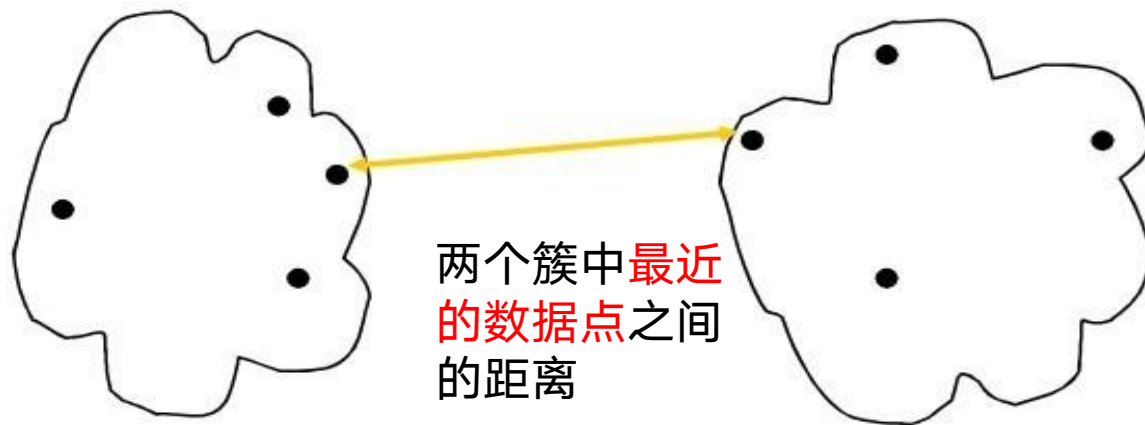
- 关键点在于计算两个簇的距离

- 算法流程

- 计算距离矩阵，将每个样本视为一个簇
 - Repeat
 - 合并两个最近的簇
 - 更新距离矩阵
 - Until 只剩一个簇

聚类分析：层次聚类

- 凝聚式层次聚类的距离定义
 - 核心问题在于距离的计算，不同聚类方法计算方式不同
 - 常见的距离定义与计算方式
 - 1. 单链（Single Link），表示为MIN
 - 指不同簇最近的点之间的距离
 - 优势：擅长处理非椭圆形状的簇
 - 缺点：对噪声比较敏感

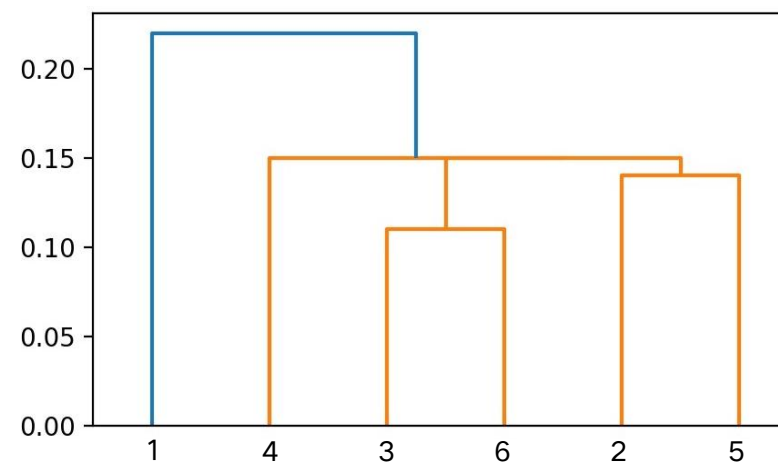
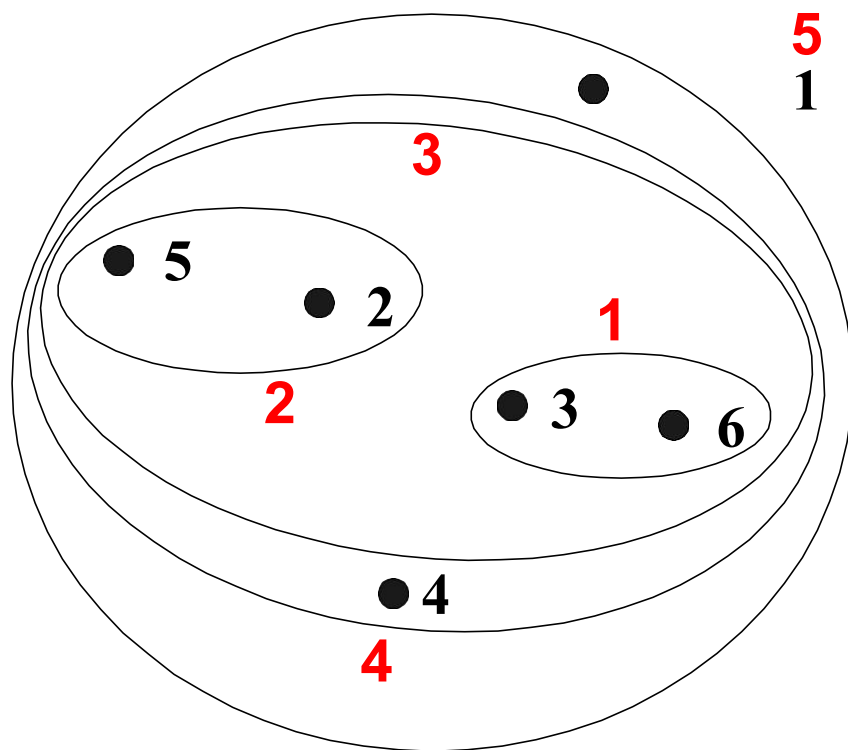


聚类分析：层次聚类

□ 例子：凝聚式层次聚类——单链方式MIN

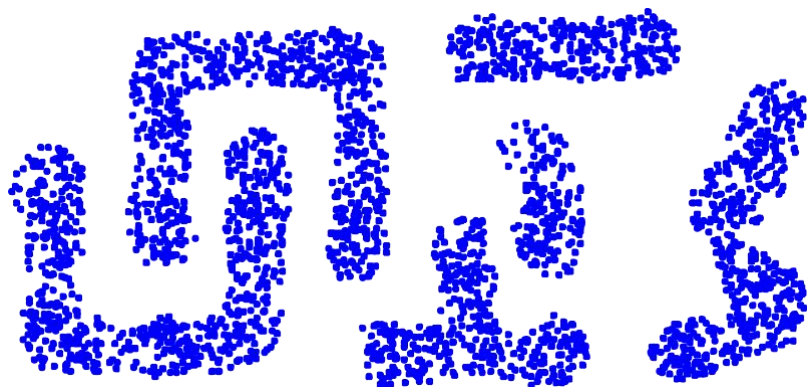
距离邻近度矩阵

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

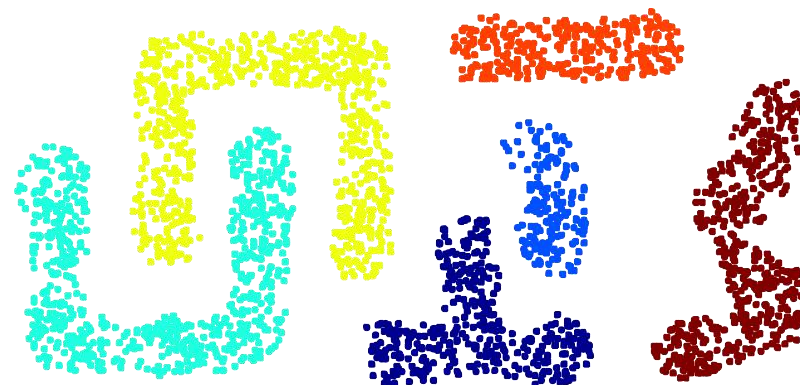


聚类分析： 层次聚类

- ▣ 例子：凝聚式层次聚类——单链方式MIN
 - ▣ 优点：能够处理非椭圆形状



Original Points

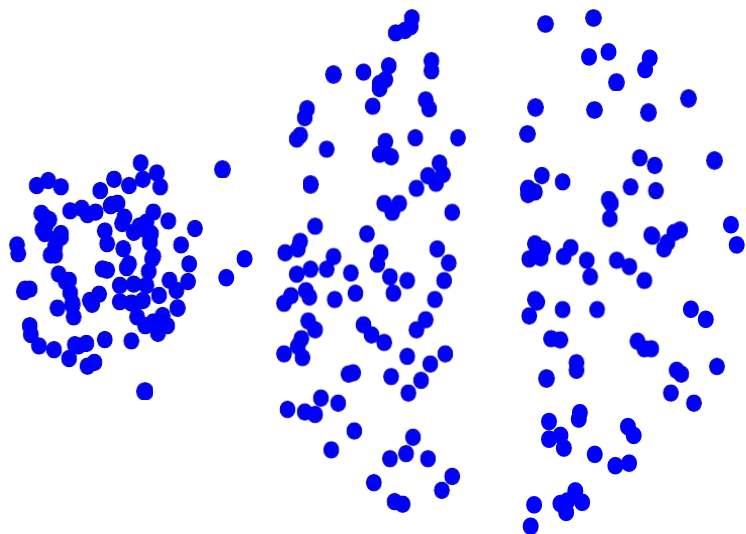


Six Clusters

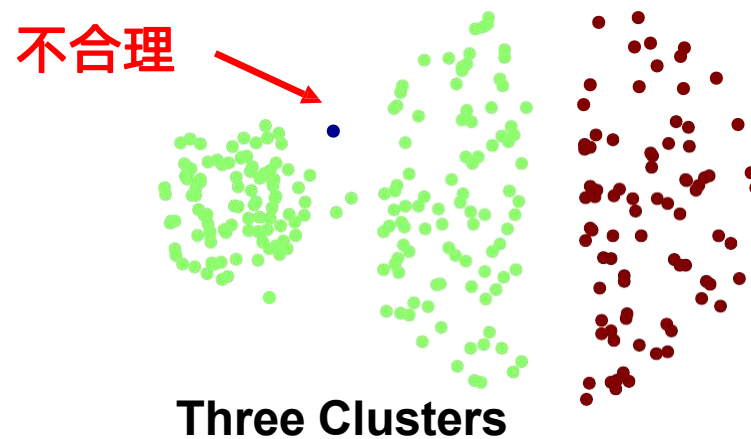
聚类分析：层次聚类

□ 例子：凝聚式层次聚类——MIN(单链)

□ 缺点：对噪声和离群点敏感



数据（含有噪声）



聚类分析：层次聚类

- 凝聚式层次聚类的距离定义

- 核心问题在于距离的计算，不同聚类方法计算方式不同

- 常见的距离定义与计算方式

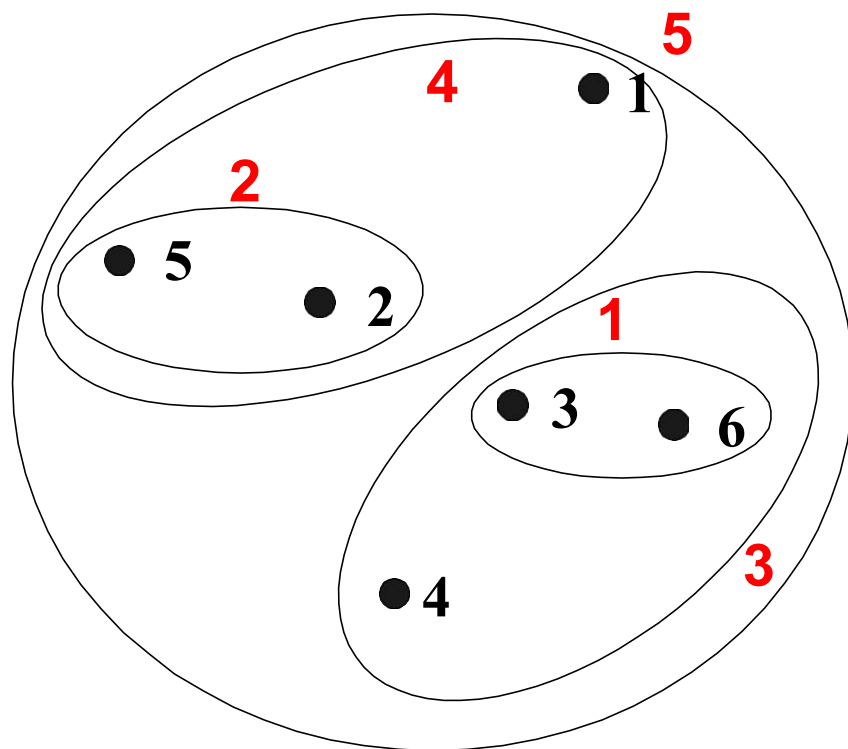
- 2. 全链（Complete Link），表示为MAX

- 指不同簇最远的点之间的距离
 - 优势：对噪声不太敏感
 - 缺点：可能使得较大的簇变得支离破碎



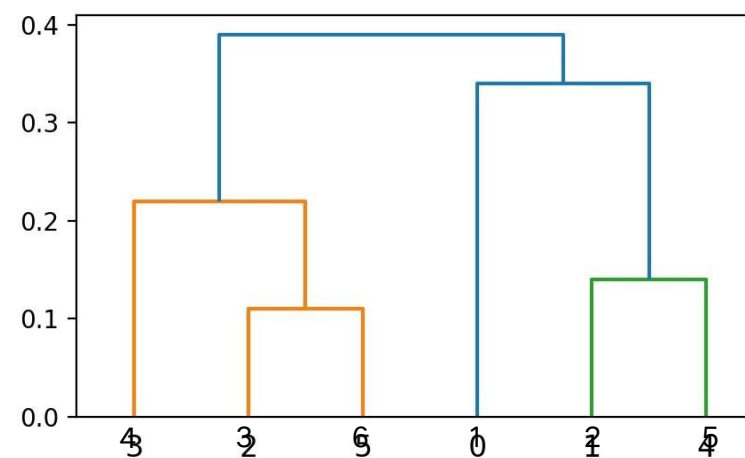
聚类分析：层次聚类

□ 例子：凝聚式层次聚类——MAX(全链)



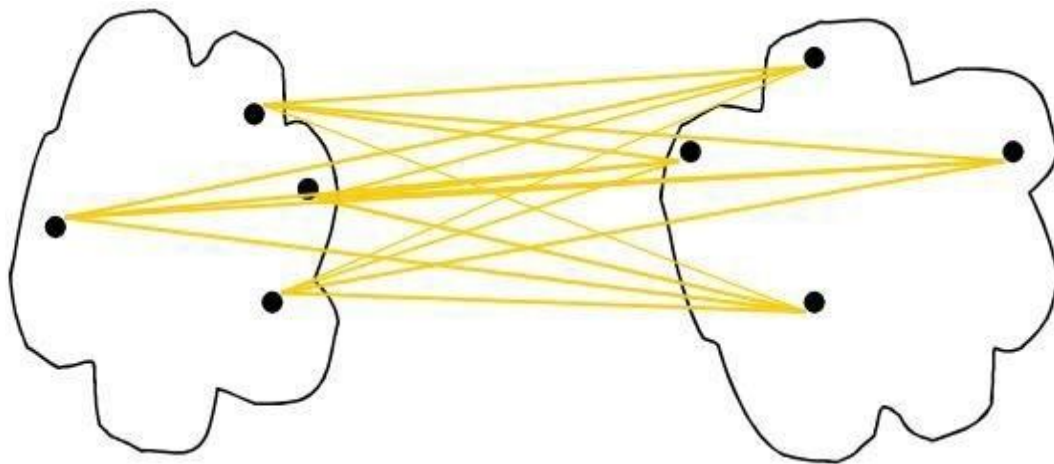
距离邻近度矩阵

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



聚类分析：层次聚类

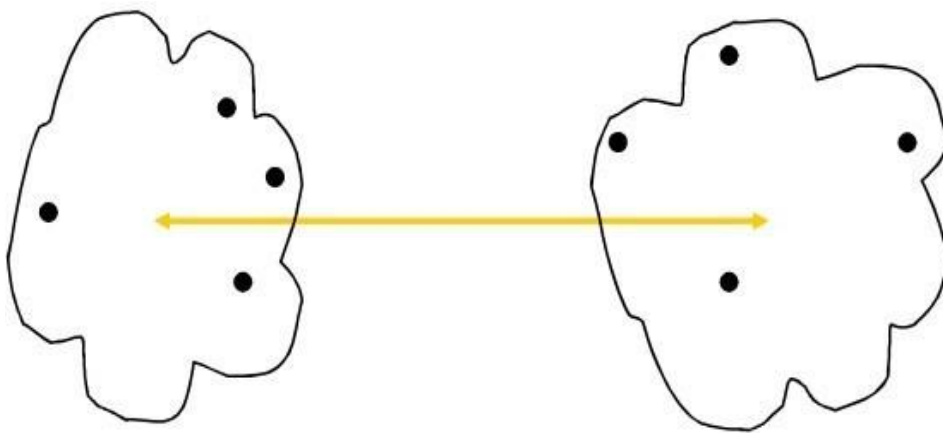
- 凝聚式层次聚类的距离定义
 - 核心问题在于距离的计算，不同聚类方法计算方式不同
 - 常见的距离定义与计算方式
 - 3. 组平均 (Group Average)
 - 所有来自不同簇的两点之间的平均距离
 - 前面两种方法的折中产物



两个簇中的数据点两两之间的平均距离

聚类分析：层次聚类

- 凝聚式层次聚类的距离定义
 - 核心问题在于距离的计算，不同聚类方法计算方式不同
 - 常见的距离定义与计算方式
 - 4. 中心距离（Group Average）
 - 所有来自不同簇的两个簇中心之间的距离
 - 使用合并两个簇导致的SSE增加值等度量方式来衡量



两个簇中的簇中心的距离

聚类分析：层次聚类

- 层次聚类的局限性
 - 每一步的合并决策都是最终的
 - 一旦做出合并两个簇的决策，就无法撤销
 - 没有全局的优化目标函数
 - 每一步都是一个局部最优的过程
 - 不同的聚类方法（邻近度定义），或多或少都具有一些问题
 - 例如，对于噪声的敏感性，或者难以保留较大的簇等

聚类分析

- ▣ 聚类方法：最常见的无监督学习算法
- ▣ 常用方法
 - ▣ K均值聚类(K-means)
 - ▣ 层次聚类(Hierarchical Clustering)
 - ▣ 密度聚类(Density-based Clustering)
 - ▣ 聚类效果验证
 - ▣ 前沿聚类方法

聚类分析：密度聚类

□ 密度聚类

- 基本假设：只有达到一定密度，才足以成为一个簇
- 密度：指定样本一定半径的样本数量
 - 半径，记为Eps
 - 半径内样本数阈值，记为MinPts

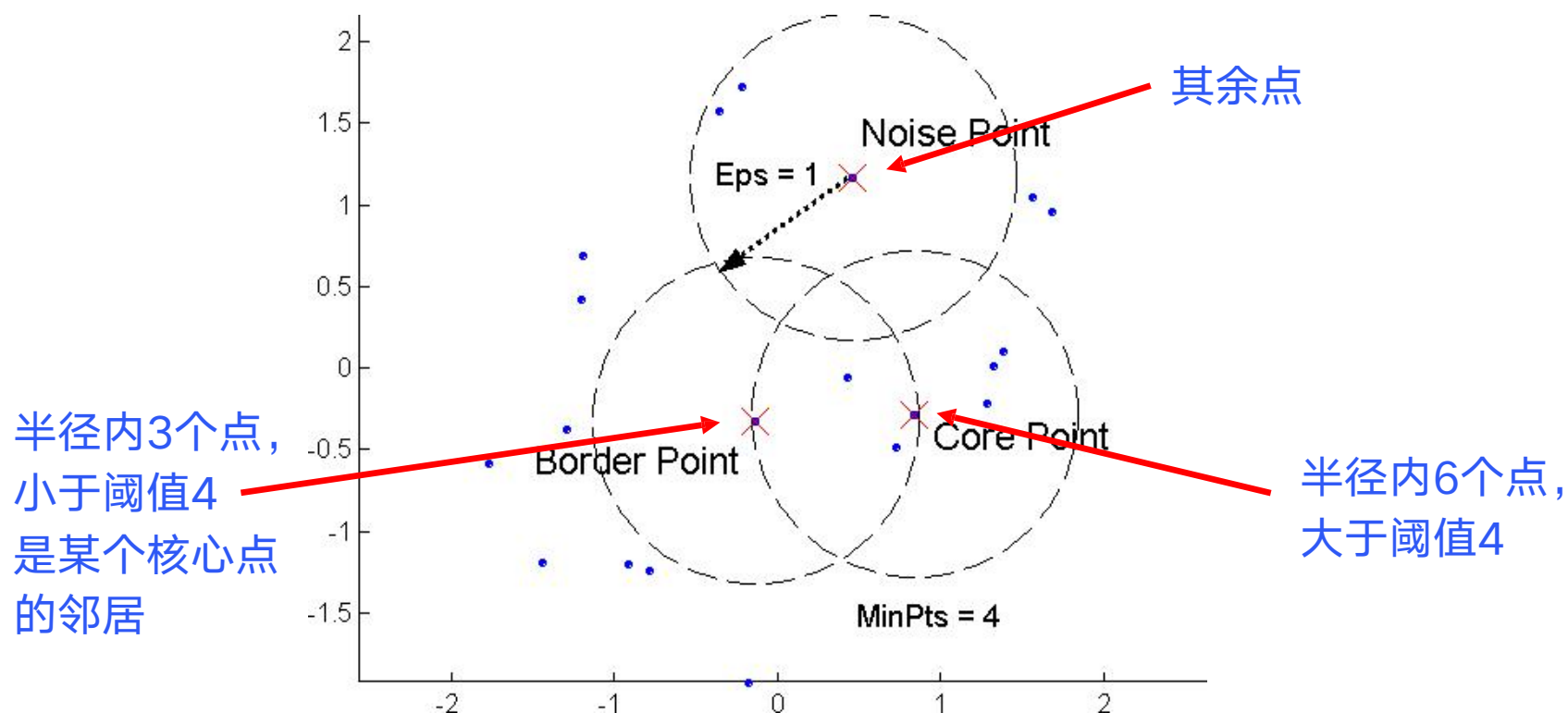
□ 典型算法：DBSCAN

- 核心要素：三类不同的数据点
- 1. 核心点(Core point)：稠密部分内部的点
 - 其Eps的范围内的样本个数不少于MinPts，这些核心点位于簇的中心
- 2. 边界点(Border point)：非核心点，但是处于稠密区域边界内/上的点
 - 其Eps的范围内的样本个数少于MinPts，但它是某个核心点的邻居
- 3. 噪音点(Noise point)：处于稀疏区域的点
 - 除核心点和边界点之外的样本

聚类分析：密度聚类

DBSCAN

三类点：核心点、边界点和噪音点 示意图



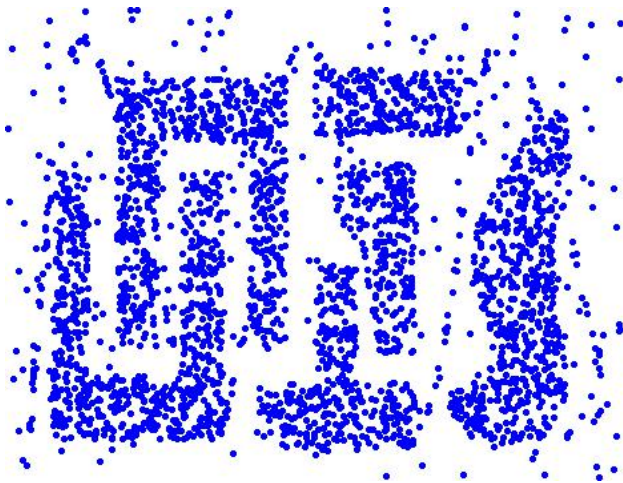
聚类分析：密度聚类

- ▣ DBSCAN的基本流程可归纳如下
 - ▣ 1. 将所有节点区分为核心点、边界点或噪声点
 - ▣ 2. 删除噪声点
 - ▣ 3. 将所有距离在预定半径内的核心点之间连一条边
 - ▣ 4. 连通的核心点形成一个簇
 - ▣ 5. 将所有的边界点指派到一个与之关联的核心点所在的簇中

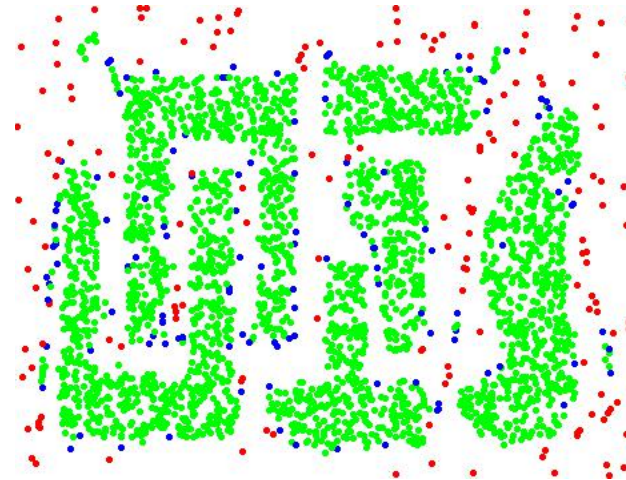
聚类分析：密度聚类

▣ DBSCAN实例

▣ Eps = 10, MinPts = 4



Original Points



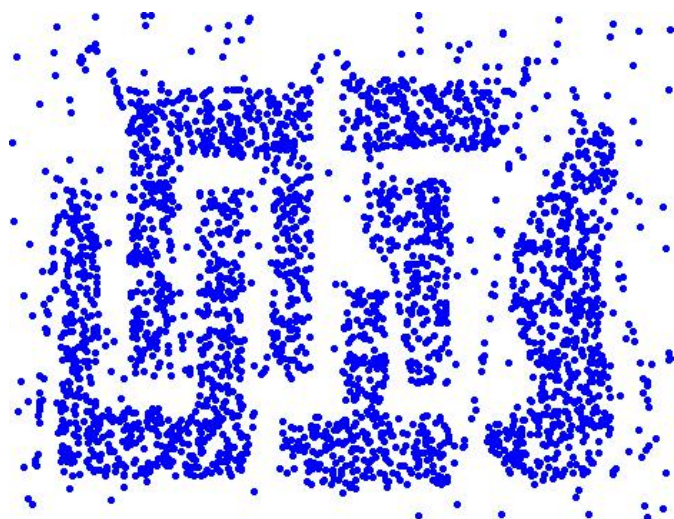
Point types:

绿色core, 蓝色border, 红色noise

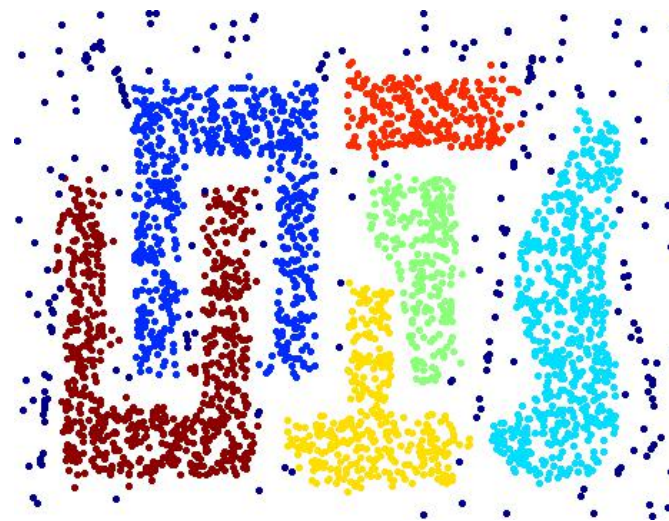
聚类分析：密度聚类

- ▣ DBSCAN的优势
 - ▣ 对噪声鲁棒
 - ▣ 能够处理不同形状和大小的簇

周边的噪声除去，内部的数据很好的聚类



Original Points



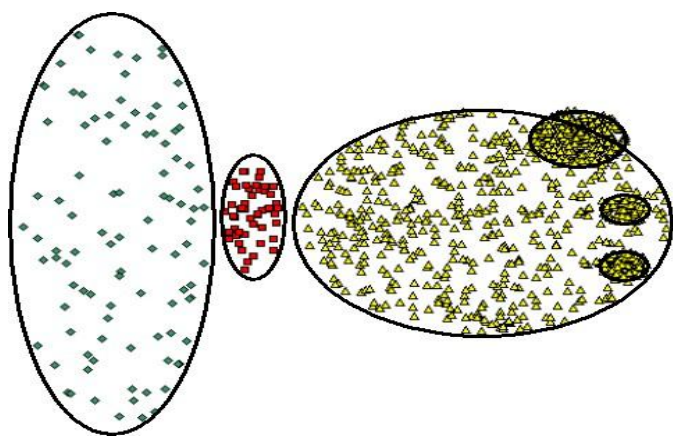
Clusters

聚类分析：密度聚类

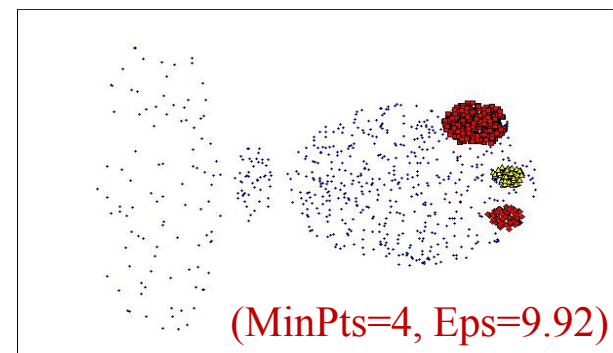
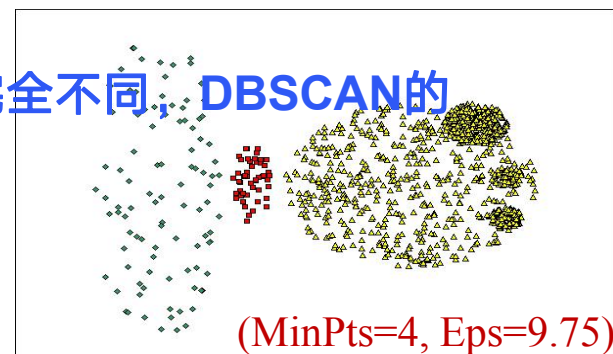
□ DBSCAN的局限性

- 簇的密度变化使得DBSCAN的效果可能会受到影响
- 参数难以设置：Eps、MinPts的选取需与数据维度匹配

例子：两种方式参数相近，但簇的密度 完全不同，DBSCAN的结果差距很大



Original Points



聚类分析：密度聚类

- ▣ DBSCAN算法获得ICDM2013 “Research Contributions Award”

