第七章　聚类分析

张静
（Jingzhang@ecust.edu.cn）

主要内容

- 基本概念
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

基本概念

- 聚类是无监督学习（unsupervised learning）
  - 没有预定义的类别
  - 观察式学习
- 典型应用
  - 作为独立工具（stand-alone tool），可表征数据分布
  - 作为其他算法的预处理步骤（preprocessing step）

聚类的应用场景

- 空间数据分析
  - 在GIS中，通过对特征空间聚类来创建主题地图。
- 图像处理
- 城市规划
- 气候研究
- WWW
  - 文档分类
  - 对Web日志进行聚类，从而发现相似的访问模式。
- 离群点检测
  - 信用卡欺诈检测；监控电子商务中的犯罪活动等。

什么是好的聚类方法?

- 一个好的聚类方法将会产生高质量的簇
  - 高簇内相似性：类内凝聚性
  - 低簇间相似性：类间区分性
- 判定一个聚类方法质量好坏依赖于
  - 用于该聚类方法的相似度度量
  - 具体实现方法
  - 能否发现部分或者所有隐藏的模式

数据挖掘对聚类的要求

- 可伸缩性
- 处理不同类型属性的能力
  - 数值型、二元类型、分类/标称类型、序数型。
- 发现任意形状的聚类
  - 基于欧几里德距离或曼哈顿距离，偏向于发现具有相近尺寸和密度的球状簇
  - 开发其他类型的其他度量
- 对于决定输入参数的领域知识需求最小
  - 参数选择

数据挖掘对聚类的要求

- 处理噪声数据的能力
  - 离群点、空缺值、未知数据、错误数据
- 增量聚类和对于输入纪录的顺序不敏感
- 聚类高维数据的能力
- 基于约束的聚类
- 可解释性和可用性

用于比较聚类方法的各个方面

- 划分准则
  - 单层 vs. 多层
- 簇的分离性
  - 互斥的 vs. 非互斥的
- 相似性度量
  - 距离 vs. 基于密度或区域的连通性
- 聚类空间
  - 整个空间 vs. 子空间

主要内容

- 基本概念
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

不同数据类型的距离度量

- 区间标度变量:
  - 粗略线性标度的连续度量，如：重量，高度等
  - **Minkowski Distance:**

$$d(X,Y)=\left(\sum_{i=1}^{n}|x_i-y_i|^{r}\right)^{\frac{1}{r}}$$

  - **Special cases: Euclidean (L_2-norm), Manhattan (L_1-norm)**

$$d(X,Y)=\sqrt{\sum_{i=1}^{n}|x_i-y_i|^{2}} \qquad d(X,Y)=\sum_{i=1}^{n}|x_i-y_i|$$

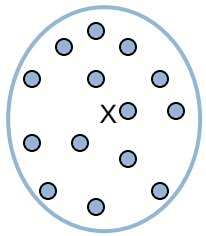
不同数据类型的距离度量

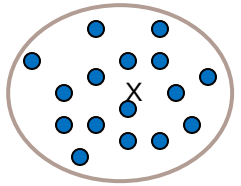
- 二元变量:
  - 只有两种状态：0, 1
  - 相异度计算：相异矩阵
  - 对称 vs. 非对称
    - 对称二元变量：两个状态具有同等价值和相同的权重。
    - 非对称二元变量：输出的状态不是同等重要的。
- 标称变量（分类变量）:
  - 二元变量的推广，可以取多个状态值
  - 相异度计算：不匹配变量的数目（或不匹配率）

不同数据类型的距离度量

- 序数变量：
  - 相异度计算：处理方法同区间标度变量
- 矢量：
  - 相异度计算： **cosine measure**
- 混合类型变量：
  - 相异度计算：
    - 按类型分组，对每种类型的变量进行单独的聚类分析
    - 将所有类型的变量一起处理，只进行一次聚类分析

簇之间的距离





- **Single link:** 一个簇中的对象和另一个簇中对象的最小距离, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- **Complete link:**一个簇中的对象和另一个簇中对象的最大距离, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- **Average:**一个簇中的对象和另一个簇中对象的平均距离, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$
- **Centroid:** 两个簇质心之间的距离, i.e., $dist(K_i, K_j) = dist(C_i, C_j)$
- **Medoid:**两个簇中心点之间的距离, i.e., $dist(K_i, K_j) = dist(M_i, M_j)$
  - Medoid: a chosen, centrally located object in the cluster

一个簇的质心（Centroid），半径（Radius）和直径（Diameter）(对于数值数据集合)

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

主要内容

- 什么是聚类分析?
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

主要聚类方法

- 划分算法
  - 构造各种各样的划分,并用一些标准来评估它们
  - 给定初始划分数目k，产生一个初始划分，然后采用迭代的重定位技术，直到找到一个好的划分。
- 层次算法
  - 使用一些策略来进行数据(或对象)集的层次分解
  - 凝聚的和分裂的
  - 缺点：不能被撤销
  - 改进
    - 在每层划分中，仔细分析对象间的“连接”
    - 集成层次凝聚和其他聚类方法。

主要聚类方法

- 基于密度的方法
  - 基于连续和密度函数
  - 只要临近区域的密度（对象或数据点的数目）超过某个阈值，就继续聚类
- 基于网格的方法
  - 基于多层粒度结构
  - 把对象量化为有限数目的单元，形成一个网格结构。聚类操作在网格结构（即量化的空间）上进行。
  - 处理时间独立于数据对象数目，只与量化空间中每一维的单元数目有关。
- 基于模型的方法
  - 为每个簇假设一个模型,寻找数据对给定模型的最佳拟合

主要内容

- 什么是聚类分析?
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

划分方法：基本概念

- 划分方法：基于一个*n*个对象或元组的数据库，构建数据的*k*个划分，每个划分表示一个簇，*k*<=*n*
- 给定一个*k*，找到一个划分方法，含*k*个簇，并且这个划分是最优的。
  - 全局最优：需要穷举所有可能的划分
  - 启发式方法：*k-means*和*k-medoids*算法
  - ***k-means*** ：每个簇用该簇中对象的平均值来表示
  - ***k-medoids*** **or PAM (Partition around medoids):** 每个簇用接近聚类中心的一个对象来表示

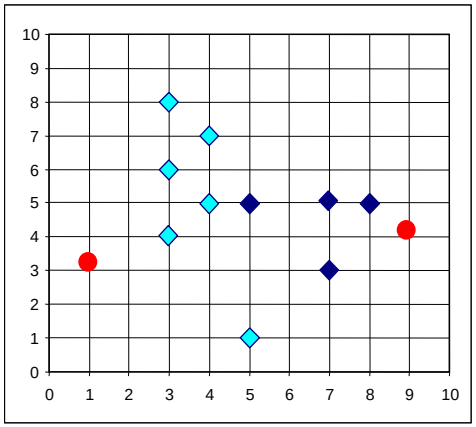
K-Means聚类方法

- K均值的处理流程如下：
  - 随机选择k个对象，每个对象初始地代表了一个簇的平均值或中心
  - 对剩余每个对象，根据其与各个簇中心的距离，将它赋给最近的簇。
  - 重新计算每个簇的平均值。
  - 这个过程不断重复，直到簇不再发生变化或准则函数收敛。
    - 平方误差准则
      - P是空间中的点，m_i是簇C_i的平均值

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} | p - m_i |^2$$

K-Means聚类方法

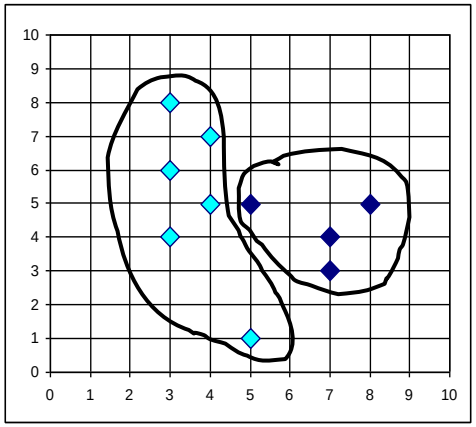
- 举例



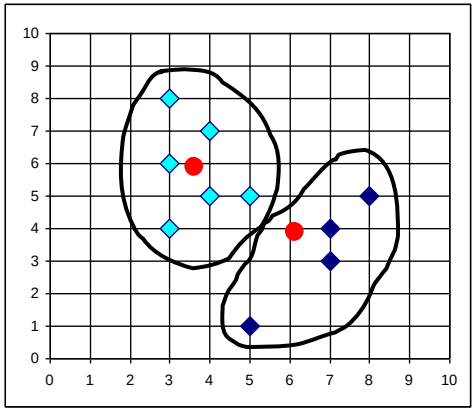
K=2

任意选择K 个对象作为初始化类中心

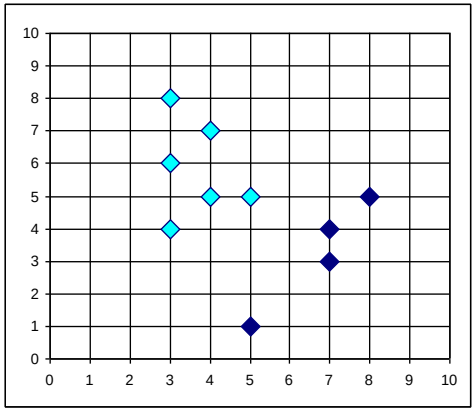
把每个对象归为最相似的中心



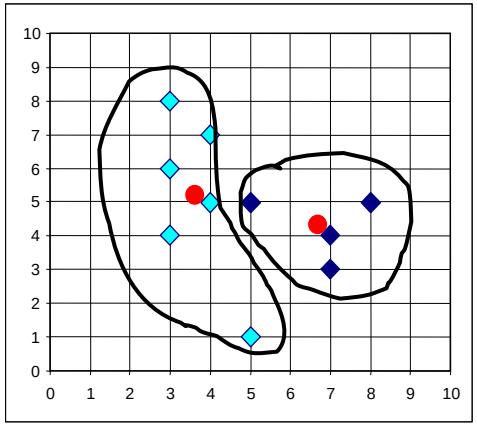
重新指派



更新簇的均值



重新指派



更新簇的均值

K-Means聚类方法

- 优点
  - 复杂度: O(nkt), 其中n 是对象的数目, k 是簇的数目, t 是迭代的次数. 通常k, t << n.
  - 相对可伸缩和高效。
  - 不能保证得到全局最优解，通常以局部最优解结束。
- 缺点
  - 只有在簇的平均值被定义的情况下才能使用，当涉及有分类属性的数据时无法处理
  - 需要事先给出k，簇的数目
  - 对噪声和离群点数据敏感
  - 不适合发现非凸形状的簇，或者大小差别很大的簇

K-Means方法的变种

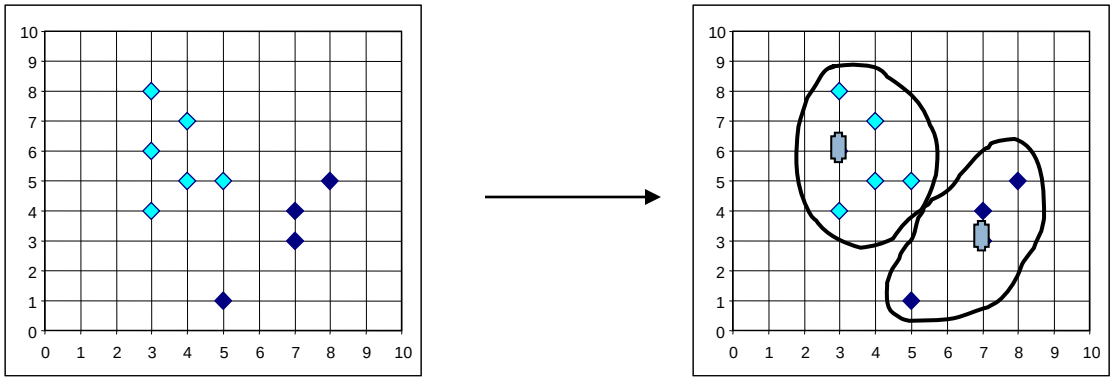
- 有许多*k-means*算法的变种，区别在于
  - 初始k个平均值的选择
  - 相异度的计算
  - 计算聚类平均值的策略
- 处理分类数据: k-众数方法（*k-modes* (Huang'98)）
  - 用众数代替簇的平均值
  - 采用新的相异度度量
  - 采用基于频率的方法更新簇众数
- 混合处理分类和数值数据: k-原型方法（*k-prototype*）
  - 将k-均值和k-众数方法综合起来

如何提高k-均值算法的可伸缩性

- 选择合适规模的样本
- 使用过滤的方法，使用空间层次数据索引节省均值的开销
- 利用微聚类的思想
  - 先划分成“微簇”再对微簇k-均值聚类

K-Means 方法存在的问题?

- k-means 算法对离群点非常敏感!
  - 因为拥有极端值的对象将在很大程度上影响数据的分布。
- K-Medoids:
  - 用中心点（位于簇最中心位置的对象）而不是簇中对象的平均值作为参考点。



K-Medoids 聚类算法

- 在各个簇中找到最有代表性的对象，即中心点（medoids）
- 基本策略
  - 为每个簇随意选择一个代表对象
  - 剩余的对象按照它跟代表对象的距离分配给最近的一个簇
  - 然后反复地用非代表对象替代代表对象，以改进聚类质量。
- 方法
  - ***PAM* (Partitioning Around Medoids, 1987)**
    - 从一个初始的集合开始，循环利用**non-medoids**替换**medoids**，看看是否能够提高各个簇的性能
    - ***PAM*** 处理小数据集合时非常有效，但是处理大数据集合时却并不很有效
  - ***CLARA* (Kaufmann & Rousseeuw, 1990)：基于抽样的PAM**
  - ***CLARANS* (Ng & Han, 1994): 随机的样本**

PAM (Partitioning Around Medoids)

- **PAM (Partitioning Around Medoids, Kaufman and Rousseeuw, 1987)**
- 算法：
  - 随机选择k个对象作为初始的中心点
  - **Repeat**
    - 指派每个剩余的对象给离它最近的中心点所代表的簇；
    - 随机地选择一个非中心点对象O_h；
    - 计算用O_h代替O_i的总代价(total swapping cost)TC_{ih}；
    - If $TC_{ih}<0$, then O_h替换O_i，形成新的k个中心点的集合；
  - **Until** 不发生变化

k-Means与k-Medoids

- 当存在噪声或离群点数据时，k-Medoids方法比k-Means方法更健壮，因为中心点不象平均值那么容易被极端数据影响
- K-Medoids方法执行代价比k-Means高
- K-Medoids方法不具有良好的可伸缩性
- 二者均要求指定结果簇的数目k

CLARA(Clustering Large Applications)

- 基于抽样的方法
- 抽取数据集合的多个样本，对每个样本应用**PAM**算法，返回最好的聚类结果作为输出
- 优点
  - 能处理规模较大的数据集
- 缺点
  - 有效性取决于样本的大小
  - 如果样本发生偏斜，基于样本的好的聚类不一定代表了整个数据集合的一个好的聚类

CLARANS (Clustering Large Application based upon RANdomized Search)

- 将采样技术同**PAM**相结合，随机化的“**CLARA**”
- **CLARANS** 动态的从近邻中抽取样本
- 聚类的过程可以被描述为对一个图的搜索，图中的每一个结点是一个潜在的解，也就是说，**k**个中心点的集合
- 如果发现局部最优，**CLARANS**从新的任意选择的结点开始寻找新的局部最优
- **CLARANS**能够探测离群点，聚类质量取决于抽样算法
- 计算复杂度 $O(n^2)$
- 利用空间数据结构的聚焦技术可以进一步改善**CLARANS**的性能

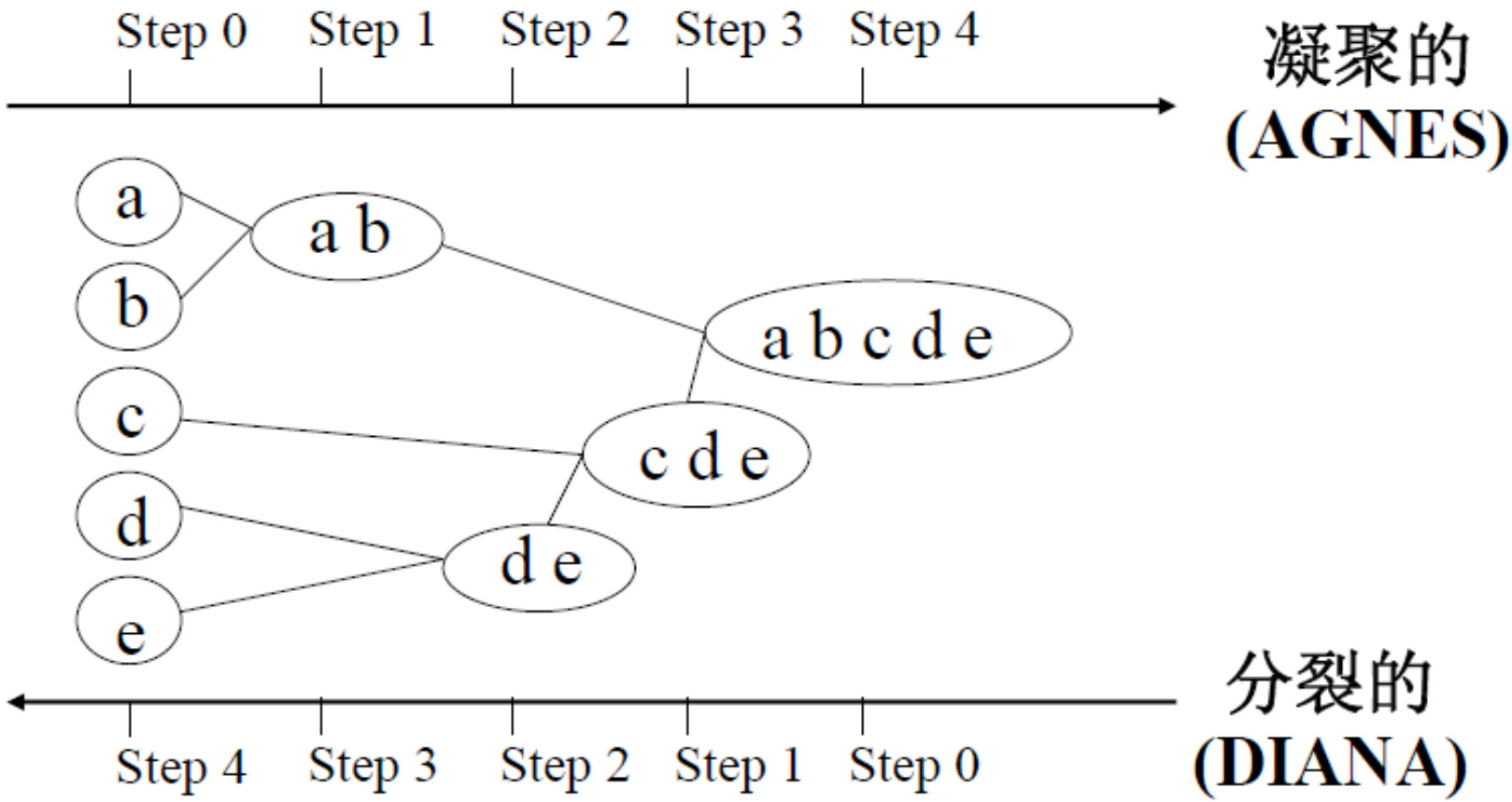
主要内容

- 什么是聚类分析?
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

层次聚类

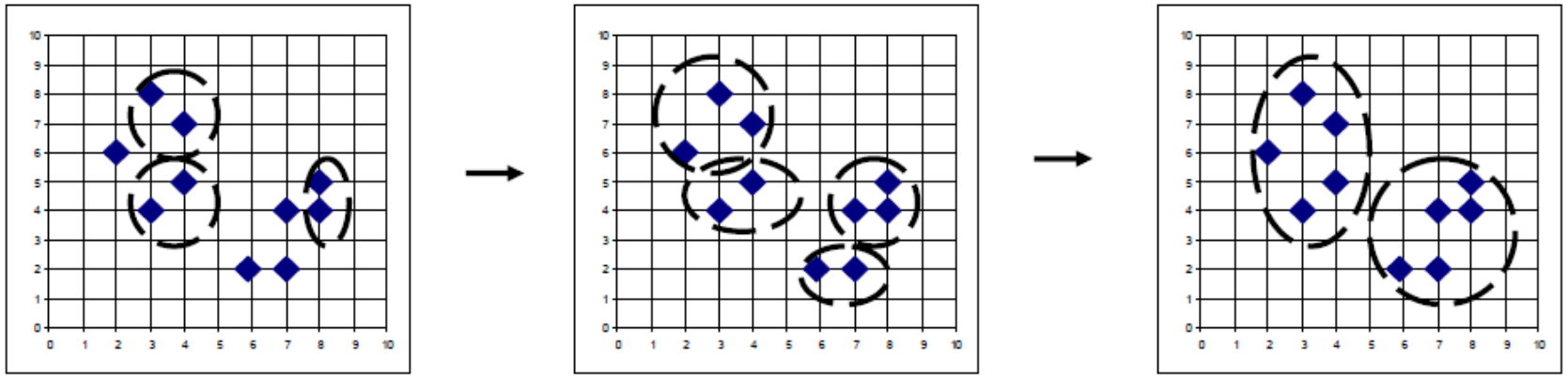
- 使用距离矩阵作为聚类的标准。这种方法不需要簇的数目k作为输入,但需要一个终止条件
- 两种类型的层次聚类方法
  - 凝聚的层次聚类
    - 自底向上
    - 首先将每个对象作为一个簇，然后合并这些原子簇为越来越大的簇，直到所有对象都在一个簇中，或满足终止条件。
  - 分裂的层次聚类
    - 自顶向下
    - 首先将所有对象置于一个簇中，然后逐渐细分，直到每个对象自成一簇，或达到某个终止条件。

层次聚类


Step 0
Step 1
Step 2
Step 3
Step 4
凝聚的
(AGNES)
a
b
c
d
e
a b
d e
c d e
a b c d e
Step 4
Step 3
Step 2
Step 1
Step 0
分裂的
(DIANA)

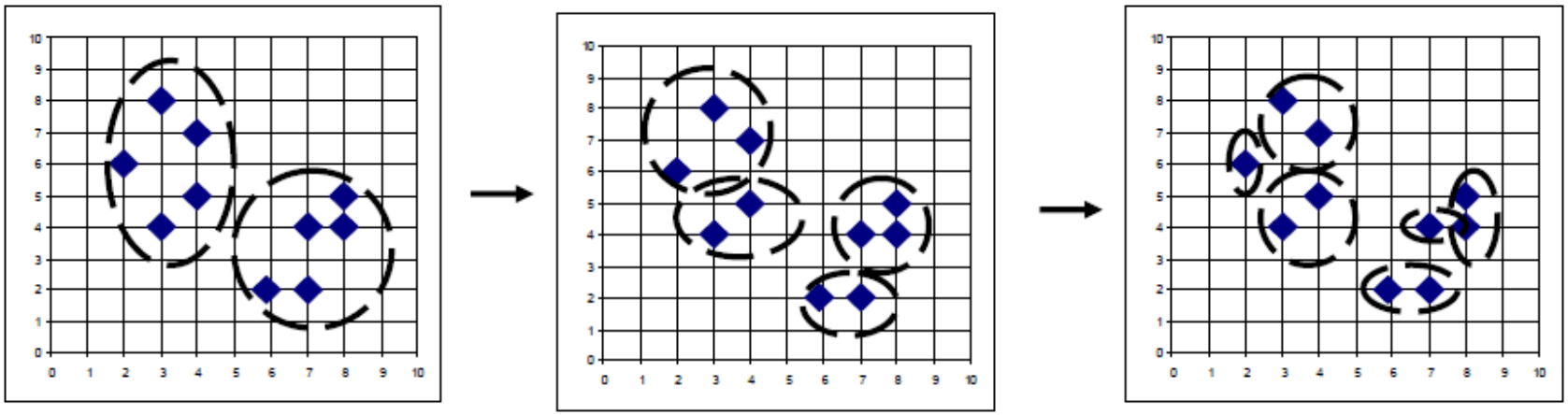
AGNES (Agglomerative Nesting)

- 凝聚层次聚类
- 使用单链接（single-link）方法和相异矩阵
- 合并最小相异的结点
- 以一个非下降的模式进行
- 最终所有的结点属于同一个簇



DIANA (Divisive Analysis)

- 分裂层次聚类
- 与**AGNES**算法相反
- 最终每一个结点形成只包含它本身的簇



层次聚类

- 四个广泛采用的簇间距离度量
  - 最小距离 $d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'|$
  - 最大距离 $d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$
  - 均值距离 $d_{mean}(C_i, C_j) = |m_i - m_j|$
  - 平均距离 $d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|$

更多关于层次聚类方法

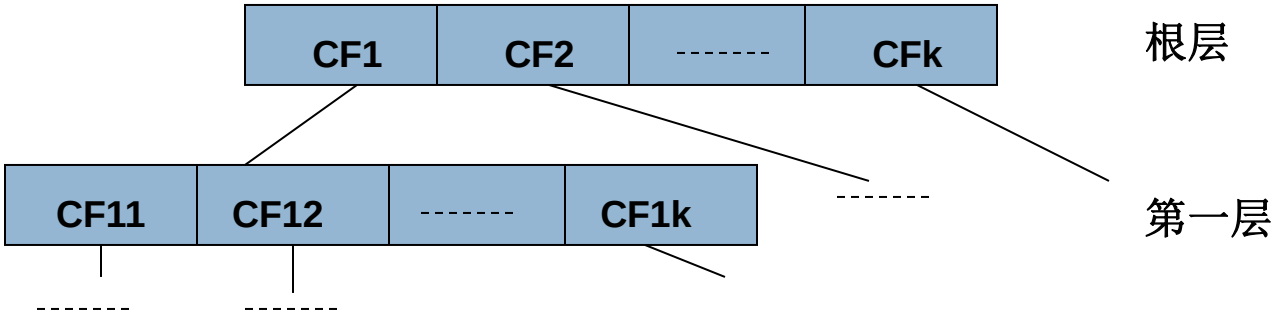
- 凝聚的层次聚类方法的主要缺点
  - 伸缩性不够好: 时间复杂度最小为 $O(n^2)$，这里n是全体对象的数目
  - 先前已做的处理不能被撤销
- 层次聚类和基于距离聚类的集成
  - **BIRCH (1996):** 使用CF－树，增量的调整子聚类的质量
  - **ROCK (1998):** 基于簇间的互连性进行合并
  - **CHAMELEON (1999):** 采用动态模型的聚类方法

BIRCH

- **BIRCH: 利用层次方法的平衡迭代归约和聚类 Zhang, Ramakrishnan, Livny(SIGMOD'96)**
- 综合的层次聚类方法
  - 聚类特征（CF）：三元组，给出对象子聚类的信息的汇总描述。
    - 假设某个子聚类中有N个d维点或对象 $\{O_i\}$，则该子聚类的CF定义如下：

$$\mathrm{CF} = (N, \overrightarrow{LS}, SS)$$

    - N是子类中点的数目，$\overrightarrow{\mathrm{LS}}$是N个点的线性和（即 $\sum_{i=1}^{N} \overrightarrow{O_i}$ ），ss是数据点的平方和（即 $\sum_{i=1}^{N} \overrightarrow{O_i}^2$ ）
  - 聚类特征树（CF树）
    - 高度平衡的树，存储了层次聚类的聚类特征
    - 一个CF树有两个参数：分支因子B，和阈值T。分支因子定义了每个非叶节点孩子的最大数目，而阈值参数给出了存储在树的叶子节点中的子聚类的最大直径。



聚类特征

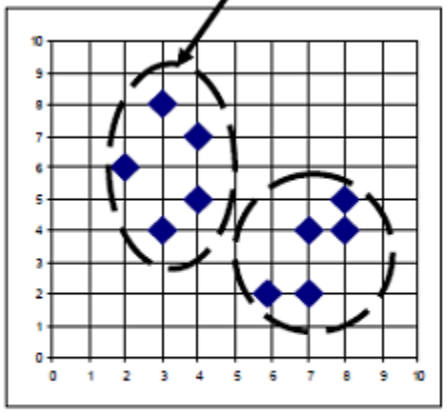
聚类特征：$CF = (N, \overrightarrow{LS}, SS)$

N: 数据点的数目

LS: $\sum_{i=1}^{N} \overrightarrow{X_i}$

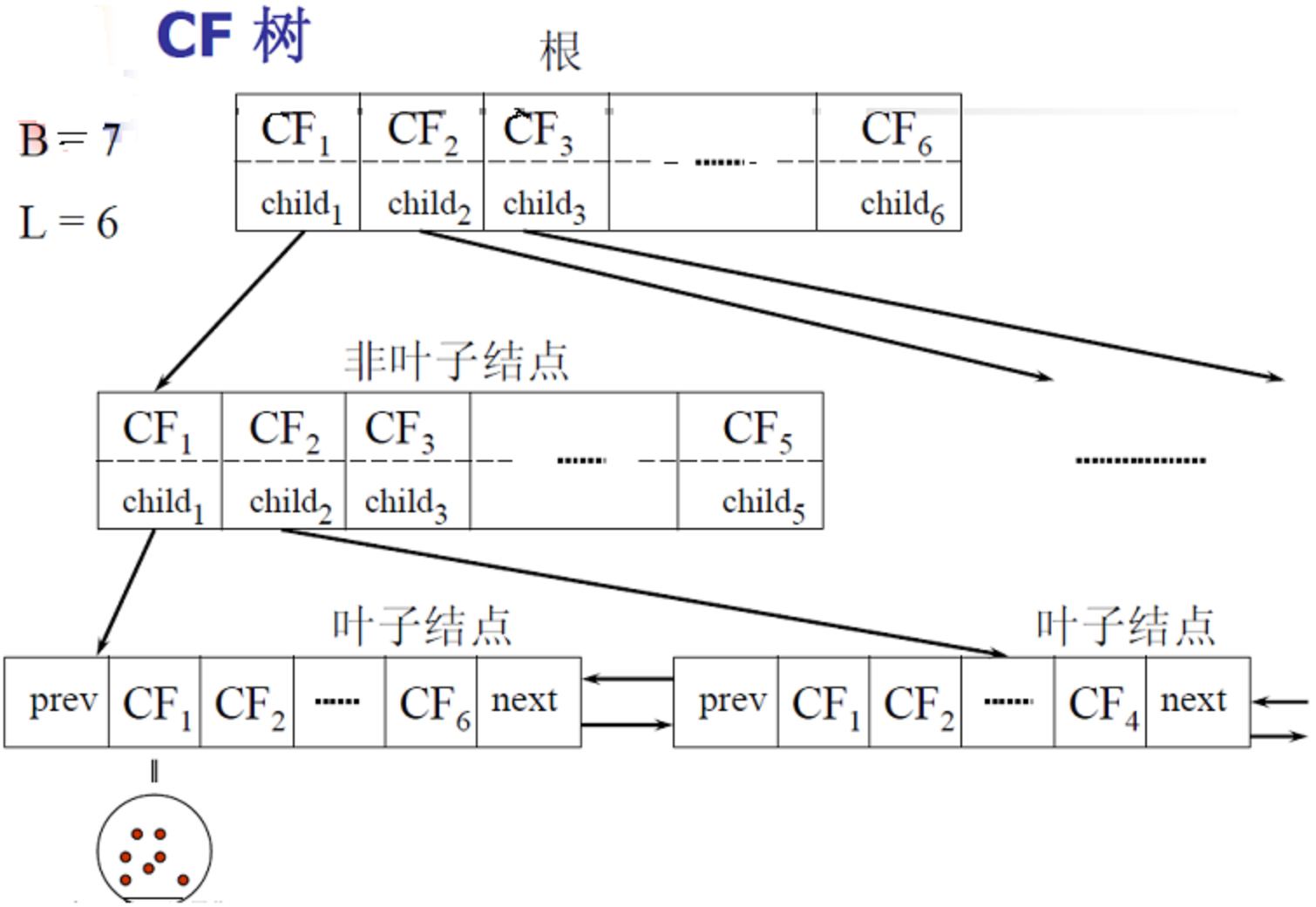
SS: $\sum_{i=1}^{N} \overrightarrow{X_i^2}$

CF = (5, (16,30),(54,190))



(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF 树


CF 树
根
B = 7
L = 6
CF1 CF2 CF3 ..... CF6
child1 child2 child3 child6
非叶子结点
CF1 CF2 CF3 ..... CF5
child1 child2 child3 child5
叶子结点
叶子结点
prev CF1 CF2 ..... CF6 next
prev CF1 CF2 ..... CF4 next

BIRCH算法

- 簇的直径 $\sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2}$
- 对于输入的每一个点
  - 找到距离其最近的叶子
  - 增加该点到该叶子结点并更新CF
  - 如果该叶子结点直径大于最大直径，则分裂该叶子结点。
- 算法的复杂度为O(n)
- 值得注意的几个问题
  - 该算法对数据的插入顺序敏感
  - 因为固定了叶节点的大小，因此聚类的结果可能会不太自然
  - 因为给定了半径和直径的度量，形成的簇趋向于球形

BIRCH

- 增量的构造CF（聚类特征）树,CF树是一个用于多阶段聚类的层次数据结构
  - 阶段1: 扫描数据库，建立一个初始存放于内存的CF树，它可以被看作数据的多层压缩，试图保留数据内在的聚类结构
  - 阶段2: 使用某个聚类算法对CF树的叶结点进行聚类
- 优点
  - 线性伸缩性
  - 支持增量聚类
- 缺点
  - 只能处理数值数据，对数据记录的顺序很敏感

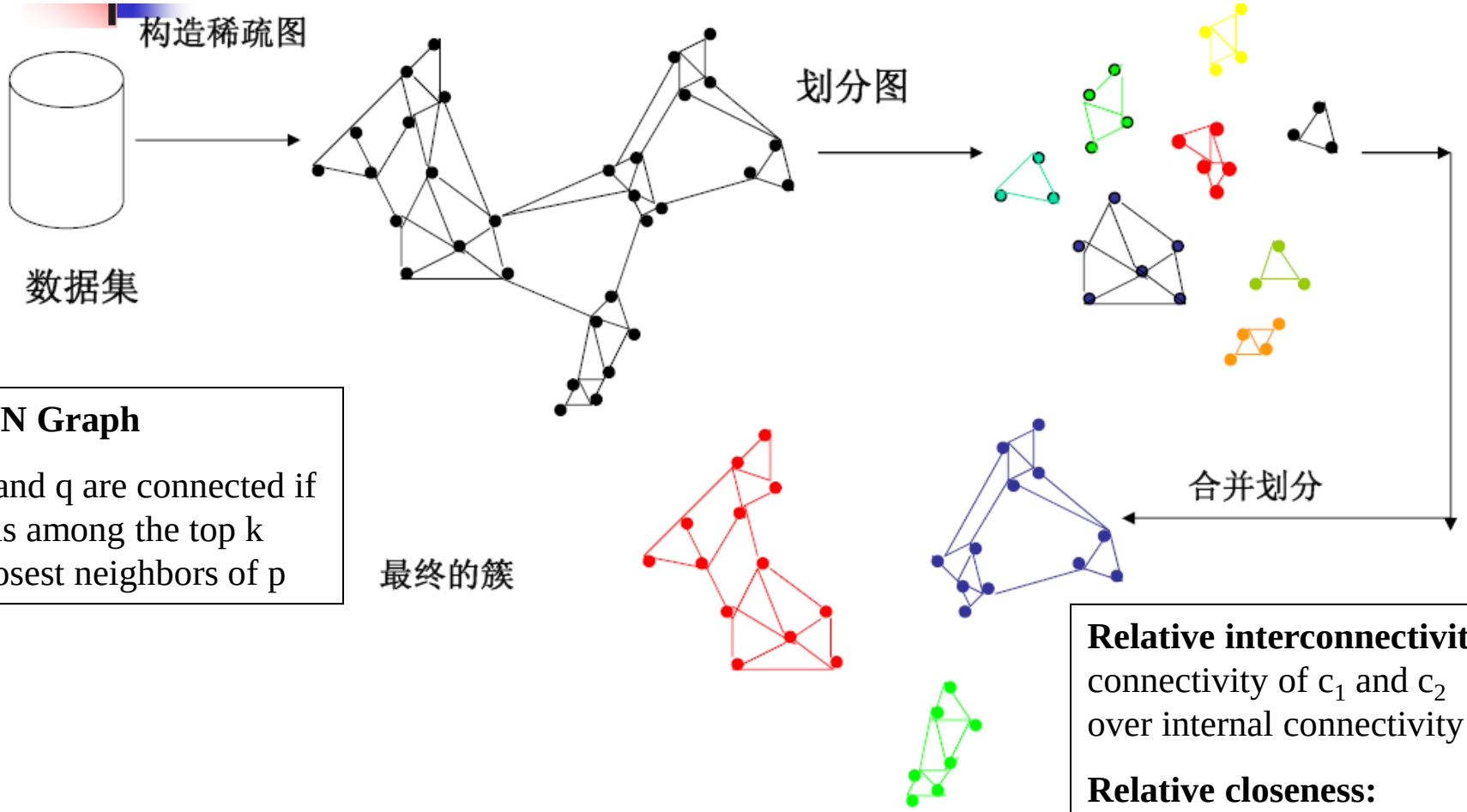
聚类分类数据:ROCK

- **ROCK(Robust Clustering using linKs)**
  - 由S. Guha, R. Rastogi, K. Shim (ICDE'99)提出，是一个凝聚的层次聚类算法，适用于分类属性。
  - 通过将聚集的互连性与用户定义的静态互连性模型相比较来度量两个簇的相似度
  - 链接---两个对象间共同的近邻数目。
  - 簇间相似度用“链接”来描述，即来自于不同簇而有相同近邻的点的数目。
- 思想
  - ROCK首先根据相似度阈值和共享近邻的概念从给定的数据相似度矩阵构建一个稀疏的图，然后再这个稀疏图上执行一个层次聚类算法

CHAMELEON（使用动态建模的多阶段层次聚类）

- **CHAMELEON**
  - 由**G. Karypis, E.H. Han**和**V. Kumar'99** 提出的，使用动态模型的层次聚类
- 在动态模型的基础上度量相似性
  - 两个簇被合并当且仅当簇间的互连性和近似度与簇内部对象间的互连性和近似度高度相关
  - 既考虑了互连性，又考虑了簇间的近似度。
- 一个两步的算法
  - 使用一个图划分算法: 把对象聚类到相对较小的子聚类
  - 使用一个凝聚的聚类层次算法: 通过反复合并这些子聚类来找到真正的聚类簇

CHAMELEON的总体框架



K-NN Graph

P and q are connected if q is among the top k closest neighbors of p

Relative interconnectivity: connectivity of c_1 and c_2 over internal connectivity

Relative closeness: closeness of c_1 and c_2 over internal closeness

主要内容

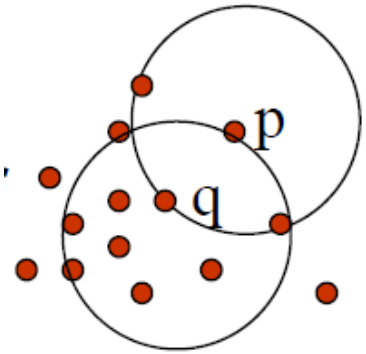
- 什么是聚类分析?
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

基于密度的聚类方法

- 该类方法把簇看作是数据空间中被低密度区域分割开的高密度对象区域。基于密度的簇是密度相连的点的集合
- 主要特点
  - 能够发现任意形状的簇
  - 能处理噪声
  - 只需一次扫描
  - 需要密度参数作为终结条件
- 研究成果
  - **DBSCAN:Ester, et al. (KDD'96)**
  - **OPTICS: Ankerst, et al (SIGMOD'99).**
  - **DENCLUE: Hinneburg& D. Keim (KDD'98)**
  - **CLIQUE: Agrawal, et al. (SIGMOD'98)**

基于密度的聚类: 背景（I）

- 两个参数:
  - ε: 邻域的最大半径
  - MinPts: 一个对象的ε-邻域至少包含的对象数
- $N_\varepsilon(p)$:{q ∈D | dist(p,q) <= ε}
- 直接密度可达的: 一个对象p是从对象q关于ε和MinPts直接密度可达的，如果
  - P是在q的ε邻域内
  - q是一个核心对象
    - 核心对象的条件:
    - $|N_\varepsilon(q)|$ >= MinPts

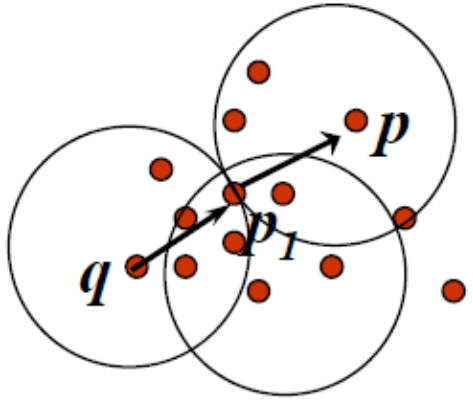


MinPts = 5

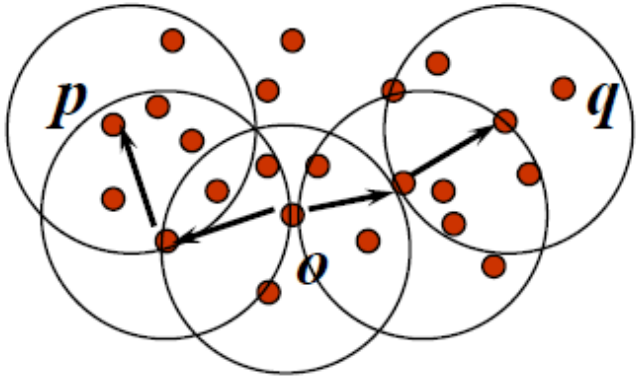
Eps = 1 cm

基于密度的聚类: 背景(II)

- 密度可达的
  - 一个对象p从对象q关于ε和MinPts是密度可达的，如果存在一个对象链$p_1, \ldots, p_n$，$p_1 = q$, $p_n = p$，p_{i+1}是从p_i关于ε和MinPts直接密度可达的

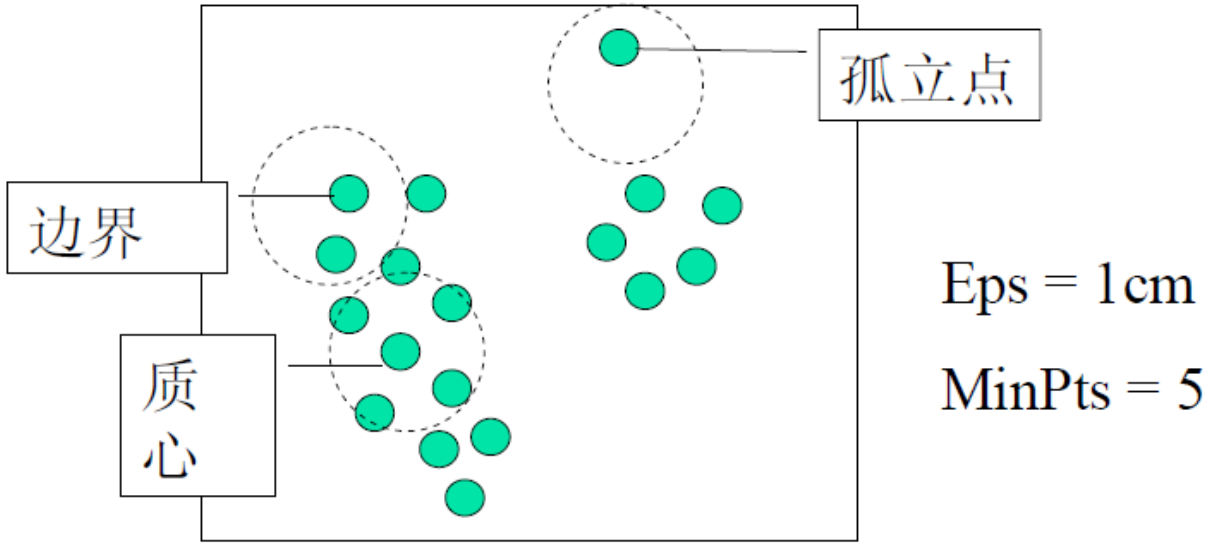


- 密度相连的
  - 如果存在一个对象o，使得对象p和q是从o关于ε和MinPts密度可达的，那么对象关于p和q是关于ε和MinPts密度相连的



DBSCAN: 一个基于高密度连接区域的密度聚类方法

- （DBSCAN）Density-Based Spatial Clustering of Applications with Noise
- 该算法将具有足够高密度的区域划分为簇，能够在带有“噪声”的空间数据库中发现任意形状的聚类
- 它定义簇为密度相连的点的最大集合



DBSCAN算法

- 算法
  - DBSCAN通过检查数据库中每个点的ε-邻域来寻找聚类。
  - 如果一个点p的ε-邻域包含多于MinPts个点，则创建一个以p作为核心对象的新簇。
  - DBSCAN反复寻找从这些核心对象直接密度可达的对象，并合并一些密度可达的簇。
  - 当没有新的点可以被添加到任何簇时，结束。

DBSCAN: 对参数十分敏感

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

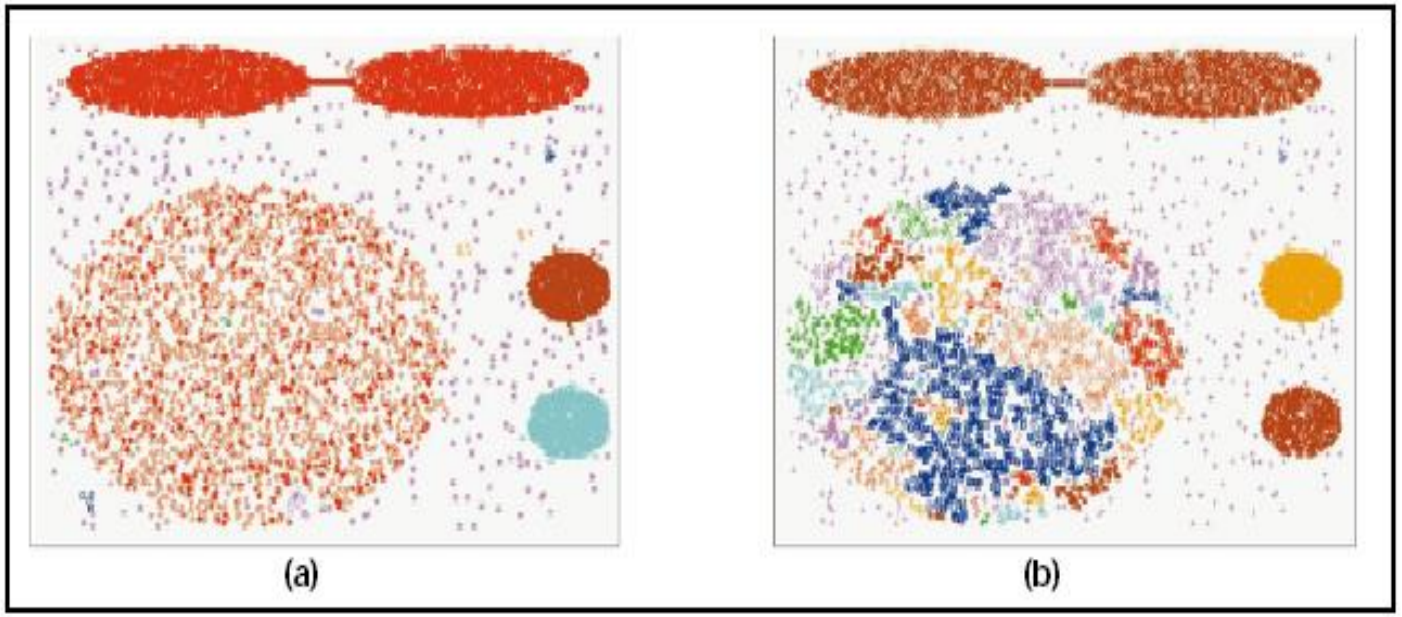
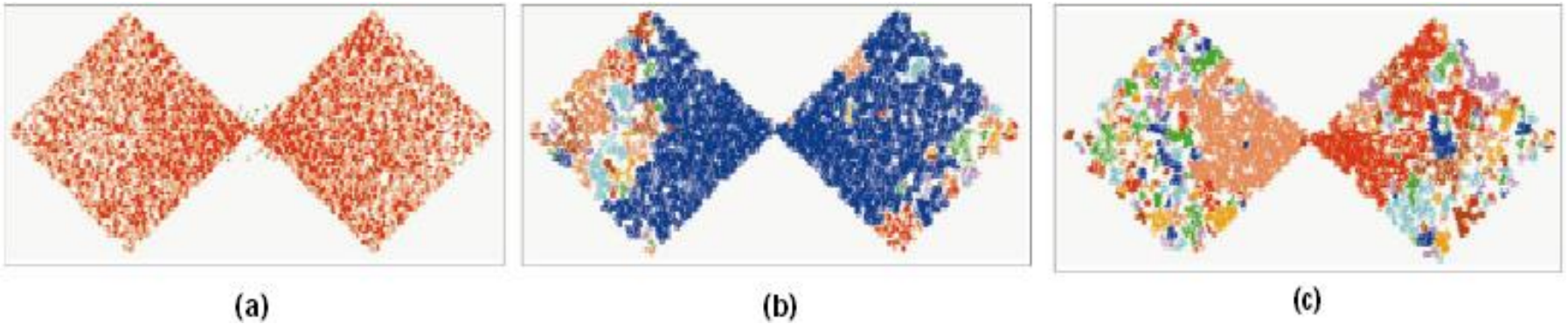


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



OPTICS: 簇次序方法(1999)

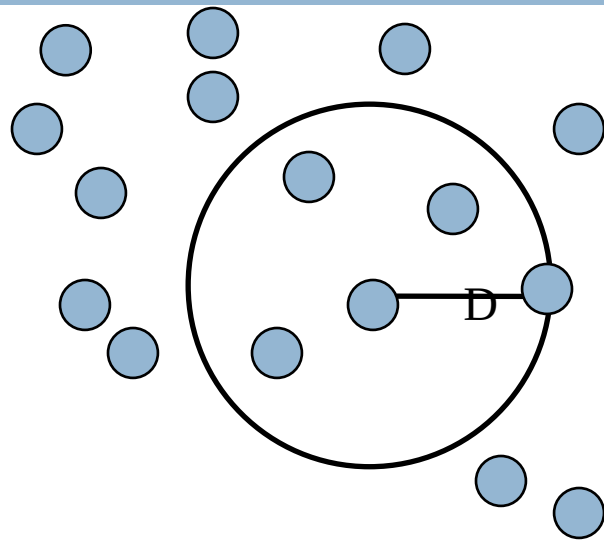
- DBSCAN的缺点
  - 两个全局参数ε和MinPts，且聚类结果与这两个参数关系密切
  - 基于密度的簇关于邻域阈值是单调的。
- 为了克服在聚类分析中使用一组全局参数的缺点，提出了OPTICS聚类分析方法

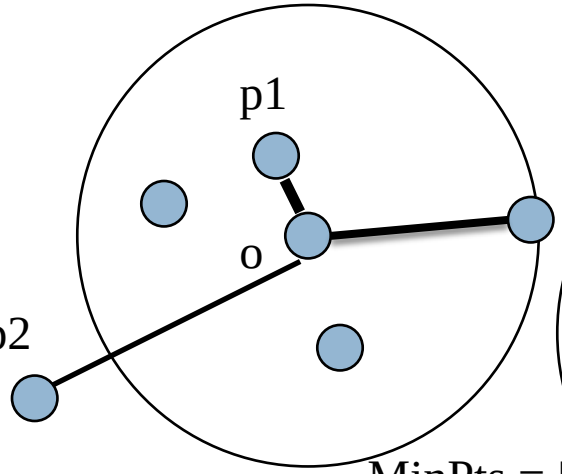
OPTICS: 簇次序方法(1999)

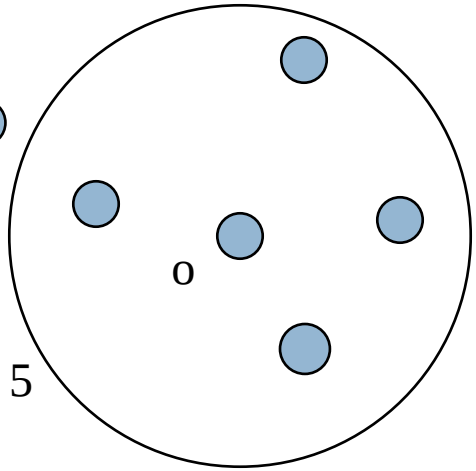
- **OPTICS(Order Points to Identify the Clustering Structure): 通过点排序识别聚类结构**
  - 不显式地产生数据集聚类，而是为自动和交互的聚类分析计算一个增广的簇排序（**cluster ordering**）。
  - 簇排序包含的信息，等同于从一个广泛的参数设置所获得的基于密度的聚类。
  - 簇排序可以用来提取基本的聚类信息（如簇中心，任意形状的簇），也可以提供内在的聚类结构。
  - 该算法对自动的和交互式聚类分析（包括找出内在的聚类结构）很有用。
  - 可以用图形或使用可视化技术表示。

OPTICS: 对DBSCAN的扩展

- 在OPTICS中，每个对象需要存储两个值—核心距离和可达距离。
- 核心距离：
  - 使p成为核心对象的最小eps距离
- 可达距离：
  - 对象p关于对象o的可达距离是o的核心距离和p与o之间的欧几里得距离的较大值







Max (core-distance (o), d (o, p))

r(p1, o) = 3cm. r(p2,o) = 4cm

MinPts = 5

ε = 3 cm

OPTICS: 对DBSCAN的扩展

- OPTICS算法创建了数据库中对象的排序，额外存储了每个对象的核心距离和相应的可达距离。
- OPTICS维护一个称作OrderSeeds的表来产生输出排序。OrderSeeds中的对象按到各自的最近核心对象的可达距离排序，即按每个对象的最小可达距离排序。
- OPTICS与DBSCAN具有相同的时间复杂度，如果使用空间索引，则复杂度为O(nlogn)，否则为 $O(n^2)$ 。

主要内容

- 什么是聚类分析?
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

基于网格的聚类方法

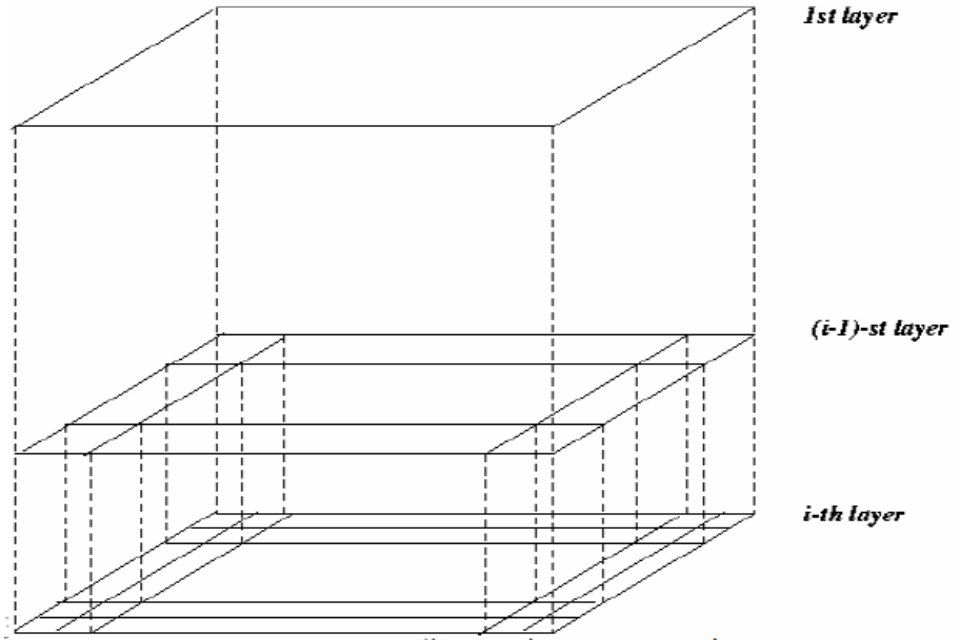
- 数据驱动的方法
  - 基于划分的、层次的、密度的等
- 空间驱动的方法
  - 基于网络的
- 基于网络的聚类方法
  - 使用一个多分辨率的网格数据结构
  - 将空间量化为有限数目的单元，这些单元形成了网格结构，所有的聚类操作都在网格上进行。

基于网格的聚类方法

- 优点
  - 处理速度快，处理速度独立于对象数目，仅依赖于量化空间中每一维上的单元数目。
- 几种代表性的方法
  - **STING (a STatisticalINformationGrid approach)由Wang, Yang 和Muntz在1997年提出**
    - 基于网格的多分辨率聚类技术
  - **WaveCluster： Sheikholeslami,Chatterjee(VLDB'98)**
    - 一个采用小波变换方法的多分辨率聚类算法
  - **CLIQUE: Agrawal, et al. (SIGMOD'98)**
    - 综合了基于密度和基于网格的聚类方法

STING: 统计信息网格方法

- (STING)STatistical INformation Grid, 由Wang, Yang 和Muntz 提出(VLDB'97)
- 基于网格的多分辨率聚类技术
- 将空间区域划分成矩形单元
- 针对不同级别的分辨率，通常存在不同级别的矩形单元



STING:统计信息网格方法

- 高层的每个单元被划分成多个低一层的单元
- 每个单元属性的统计信息被预先计算和存储，这对查询处理是有用的
- 高层单元的统计参数很容易从低层单元的计算得到，这些参数包括：
  - **Count**（计数），**mean**（平均值），**s**（标准差），**min**（最小值），**max**（最大值）
  - 分布类型：正态分布，均匀分布，指数分布等等

STING:统计信息网格方法

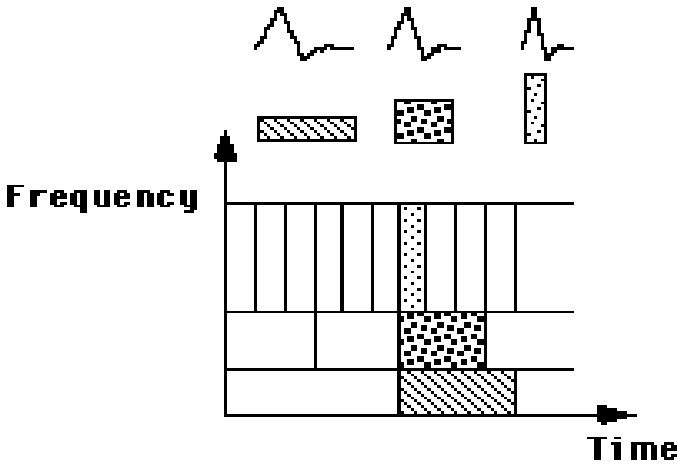
- 使用自顶向下的方法回答空间数据的查询
  - 从一个预先选择的层次开始----通常包含少量的单元
  - 为当前层的每个单元计算置信区间
  - 不相关的单元不再考虑
  - 当检查完当前层，接着检查下一个低层次
  - 重复这个过程直到达到底层
- 如果粒度趋于0（即朝向非常低层的数据），则它趋向于DBSCAN的聚类结果。

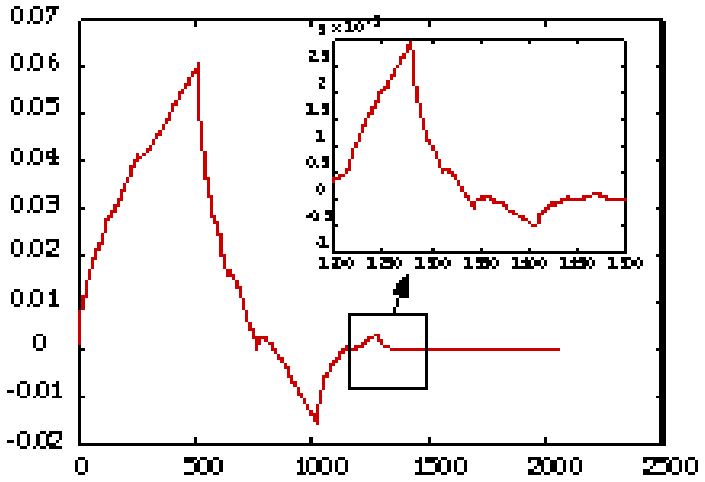
STING:统计信息网格方法

- 优点
  - 独立于查询，因为存储在每个单元中的统计信息提供了单元中数据汇总的信息，不依赖于查询。
  - 网格结构有利于并行处理和增量更新
  - 效率高，查询处理时间O(k),其中k是最底层网格单元的数目
- 缺点
  - 所有的聚类边界或者是水平的，或者是垂直的，没有对角的边界

WaveCluster: 利用小波变换聚类

- **Sheikholeslami, Chatterjee, and Zhang (VLDB'98)**
- 将小波变换应用到特征空间，基于网格和基于密度的聚类算法。
- 小波变换：一种信号处理技术，它将一个信号分解为不同频率的子波段。
  - 数据被转换后，在不同的分辨率水平依然保持对象间的相对距离
  - 使得数据的自然簇变得更加容易区分



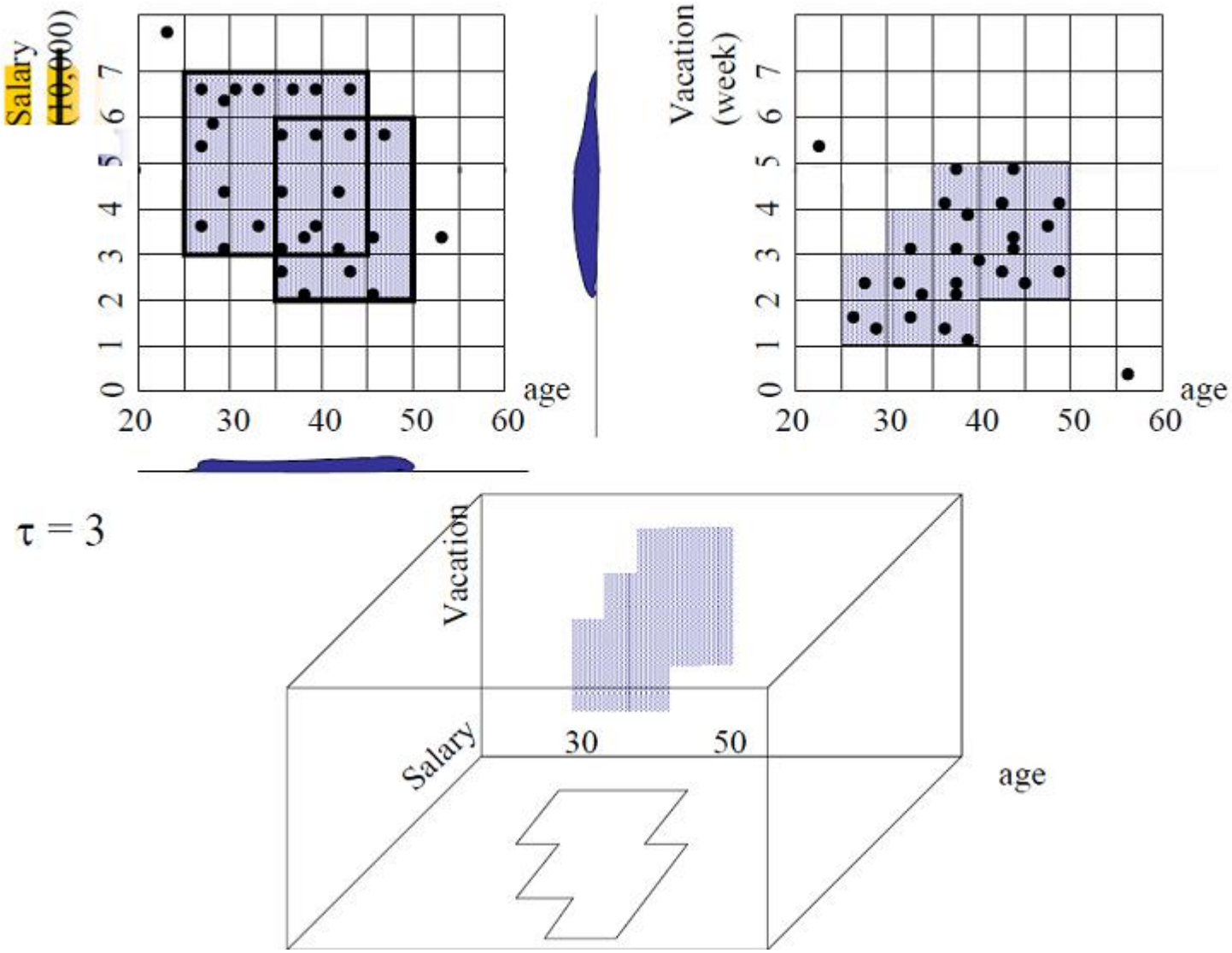


The WaveCluster Algorithm

- 如何利用小波变换找到聚类
  - 在数据空间用多维网格结构概括数据
  - 多维的空间数据对象被表示成n维特征空间
  - 在特征空间上运用小波变换找到密度区域
  - 多次运用小波变换在不同尺度上进行数据聚类
- 主要特征
  - 计算复杂度 O(N)
  - 能够在不同尺度上找到任意形状的聚类
  - 对噪音不敏感，对输入顺序也不敏感
  - 仅应用于低维数据

CLIQUE： 聚类高维空间

- **CLIQUE（CLustering In QUEst）**
  - 一种类似于**Apriori**的子空间聚类算法
  - 综合了基于密度和基于网格的聚类方法
  - 通过自动识别高维数据空间的子空间，来达到比在原空间更好聚类的目的
- **CLIQUE的中心思想**
  - 区分空间中稀疏的和“拥挤的”区域（或单元），以发现数据集合的全局分布模式
  - 簇定义为互连的稠密单元的最大集合。
  - 利用稠密单元关于维度的单调性


Salary
(10,000)
0 1 2 3 4 5 6 7
20 30 40 50 60
age
Vacation
(week)
0 1 2 3 4 5 6 7
20 30 40 50 60
age
τ = 3
Vacation
Salary
30
50
age

CLIQUE： 聚类高维空间

- CLIQUE 分两步进行多维聚类
  - 把一个n维的数据空间划分为互不相交的矩形单元，识别其中的稠密单元。
    - 一个单元是稠密的，如果包含在这个单元的数据点的总数超过了某个输入的模型参数值
    - 如果一个k维单元是稠密的，则该单元在k-1维子空间的投影也是稠密的。
  - 为每个簇生成一个最小描述。
    - 对每个簇，确定覆盖连通稠密单元簇的最大区域
    - 然后为每个簇确定一个最小覆盖

CLIQUE的优缺点

- 优点
  - 自动的发现最高维的子空间，高密度聚类存在于这些子空间中
  - 对元组的输入顺序不敏感，无需假设任何规范的数据分布
  - 它随输入数据的大小线性地扩展，当数据的维增加时具有很好的可伸缩性
- 缺点
  - 由于方法大大简化，聚类结果的精确性可能会降低

主要内容

- 什么是聚类分析?
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

基于模型的方法

- 基于模型的聚类方法通常假定数据具备一定的概率分布，并通过优化给定的数据，使其适应某些数学模型，并基于此产生聚类。
- 统计学方法
  - 期望最大化方法**EM (Expectation maximization)**

混合密度模型（Mixture Models）

- 每个簇都可以用概率分布模型来描述
  - 实际上，假定任何一个分布都可以用多参数正态分布来很好的近似
- 多个簇是不同概率分布的混合。
- 整个数据集就是这些模型的混合。

对象概率

- 假定有k个聚类和一个有m个对象的集合X
  - 令第j个聚类包含参数$\theta_j = (\mu_j, \sigma_j)$
  - 一个点在第j个聚类中的概率是w_j, 其中 $w_1 + \ldots + w_k = 1$
- 对象x的概率为

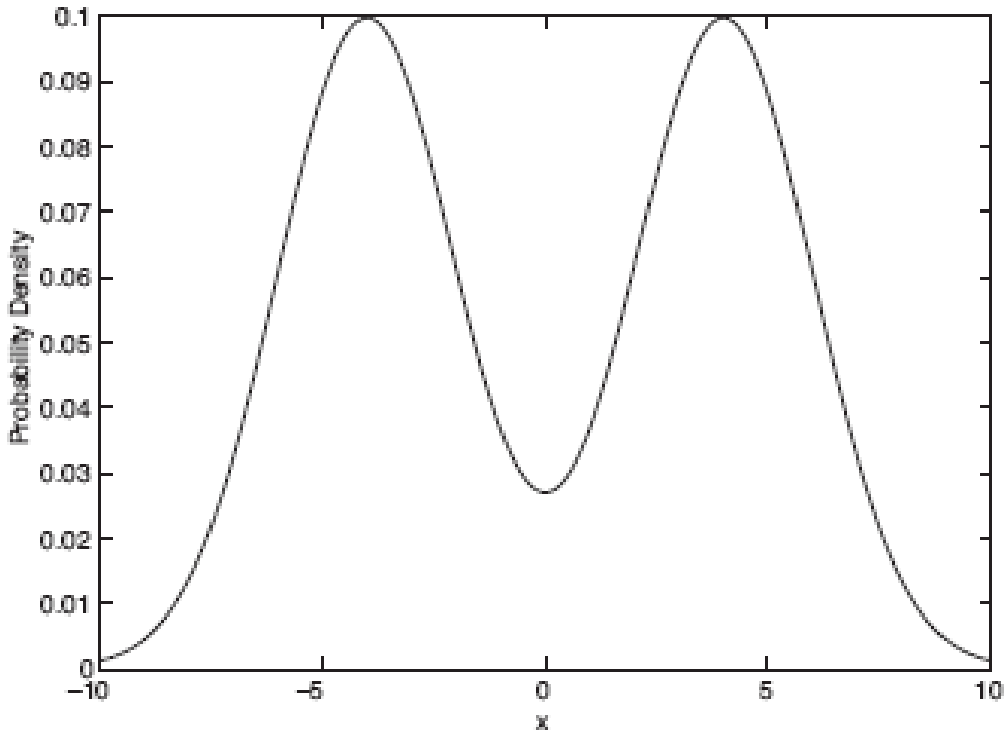
$$prob(x \mid \Theta) = \sum_{j=1}^{k} w_j p_j(x \mid \theta_j)$$

$$prob(X \mid \Theta) = \prod_{i=1}^{m} prob(x_i \mid \Theta) = \prod_{i=1}^{m} \sum_{j=1}^{k} w_j p_j(x_i \mid \theta_j)$$

例子

$$prob(x_i \mid \Theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\theta_1 = (-4,2) \quad \theta_2 = (4,2)$$



$$prob(x \mid \Theta) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x+4)^2}{8}} + \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x-4)^2}{8}}$$

最大似然估计（Maximal Likelihood Estimation）

- 最大似然定律:
  - 如果知道一组对象来自一个分布，但是不知道其中的参数，可以选择参数最大化这个概率。
- 最大化

$$prob(x_i \mid \Theta) = \prod_{j=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  - 等同于最大化

$$\log prob(X \mid \Theta) = -\sum_{i=1}^{m} \frac{(x_i - \mu)^2}{2\sigma^2} - 0.5m \log 2\pi - m \log \sigma$$

The EM (Expectation Maximization) Algorithm

- 期望最大化算法
  - 选择模型参数的一个初始化集合
  - **Repeat**
    - 期望步: 对每一个对象，计算对象x_i隶属于每一个分布概率θ_i的概率，即, $prob(x_i|\theta_i)$
    - 最大化步: 利用前面得到的概率估计重新估计模型参数，对给定数据的分布似然“最大化”
  - **Until** 参数趋于稳定

混合模型的优缺点

- 优点
  - 混合模型比k-means更通用
  - 簇能够被极少的几个参数刻画
- 缺点
  - 计算代价高
  - 需要大的数据集合
  - 很难估计簇的数目

主要内容

- 什么是聚类分析?
- 聚类分析中的数据类型
- 主要的聚类方法
  - 划分方法
  - 层次方法
  - 基于密度的方法
  - 基于网格的方法
  - 基于模型的方法
- 小结

小结

- 主要的聚类方法
  - 划分方法:k-means,k-mediods,CLARA，CLARANS
  - 层次方法: BIRCH,ROCK
  - 基于密度的方法: DBSCAN, OPTICS
  - 基于网格的方法: STING, CLIQUE
  - 基于模型的方法： EM

小结

- 聚类分析组合而成的对象基于它们的相似性有很广泛的应用
- 相异度可以用多种类型的数据来计算
- 聚类算法可以分为划分方法,层次方法,基于密度的方法,基于方格的方法和基于模型的方法
- 在聚类分析上仍旧存在许多研究问题,如基于约束的聚类等

参考文献（1）

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scietific, 1996
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. PrinticeHall, 1988.

参考文献（2）

- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorrand R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.
- G. J. McLachlanand K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering techniques. Future Generation Computer systems, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition, 101-105.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases.VLDB'98.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.

思考题

- 概述k均值和k中心点算法的优缺点。并概述这两种方法与层次聚类方法相比有何优缺点。
- 指出在何种情况下，基于密度的聚类方法比基于划分的聚类方法和层次聚类方法更适合。

END