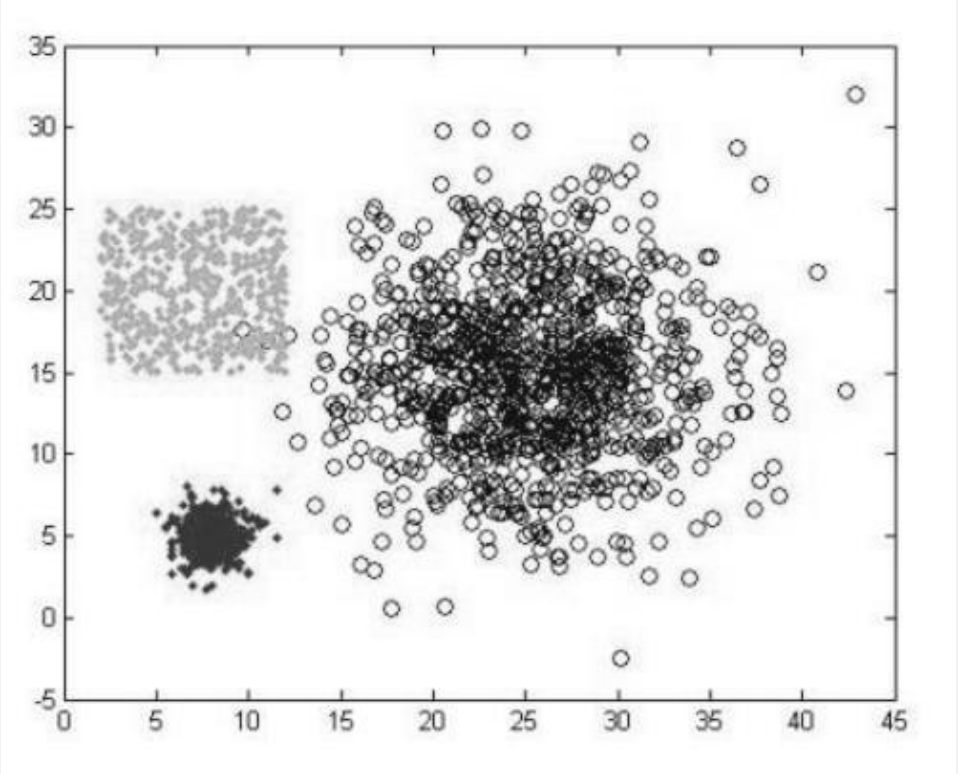
大数据分析与应用

第7章. 聚类方法

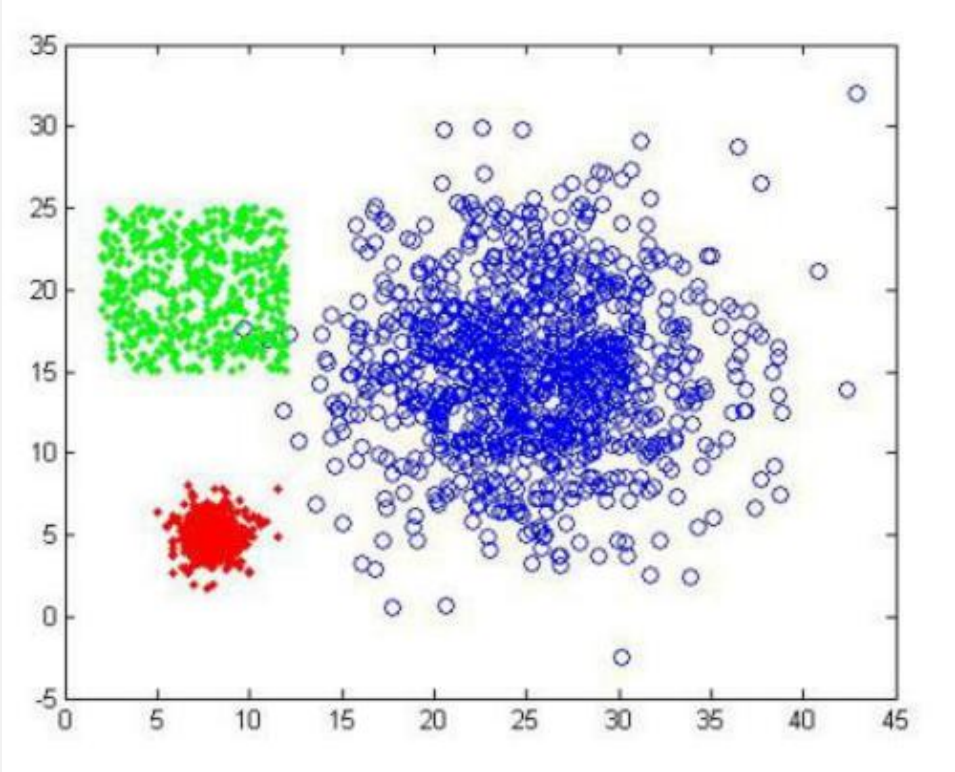
- 很多时候没有学习数据可以提供，但需要寻找数据中的相似性，把相似的数据放在一起：**非监督学习**: 数据点中没有预制的标签

- 生态学：种群聚类
- 互联网：网站划分，文本划分
- 地理：地形划分（基本农田统计）
- 市场营销：发现客户中不同的类别
- 城市规划：划分城市功能区
- 气象学：划分不同的气候地区
- 经济学：市场划分



第7章 聚类方法：基本定义

- 簇 （Cluster）: 数据点的集合
  - 同一个簇中的数据点有相似性
  - 与不同的簇中的数据点有显著不同
- 一个好的聚类方法可以得到质量高的簇
  - 簇内的数据具有高的相似性: 簇内的聚合度高
  - 簇与簇之间相似性低: 簇与簇区分度高

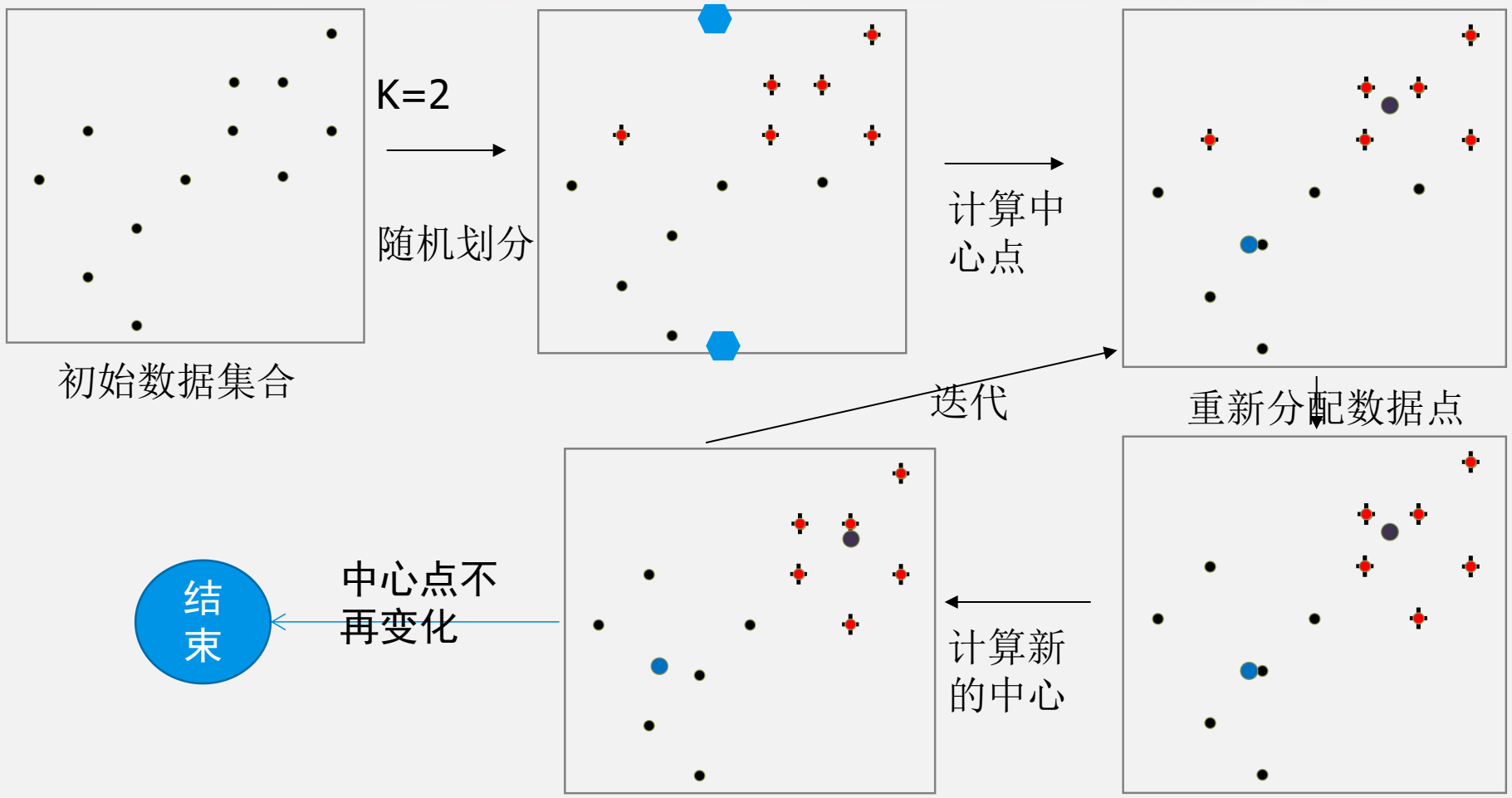


- 聚类分析的作用：
- 预处理: 探索数据的分布情况
- 压缩:数据的压缩，特征抽取
- 缩小数据范围：在更小的范围中搜索结果
- 离群点检测：离群点（异常点）一般远离正常的数据点组成的簇

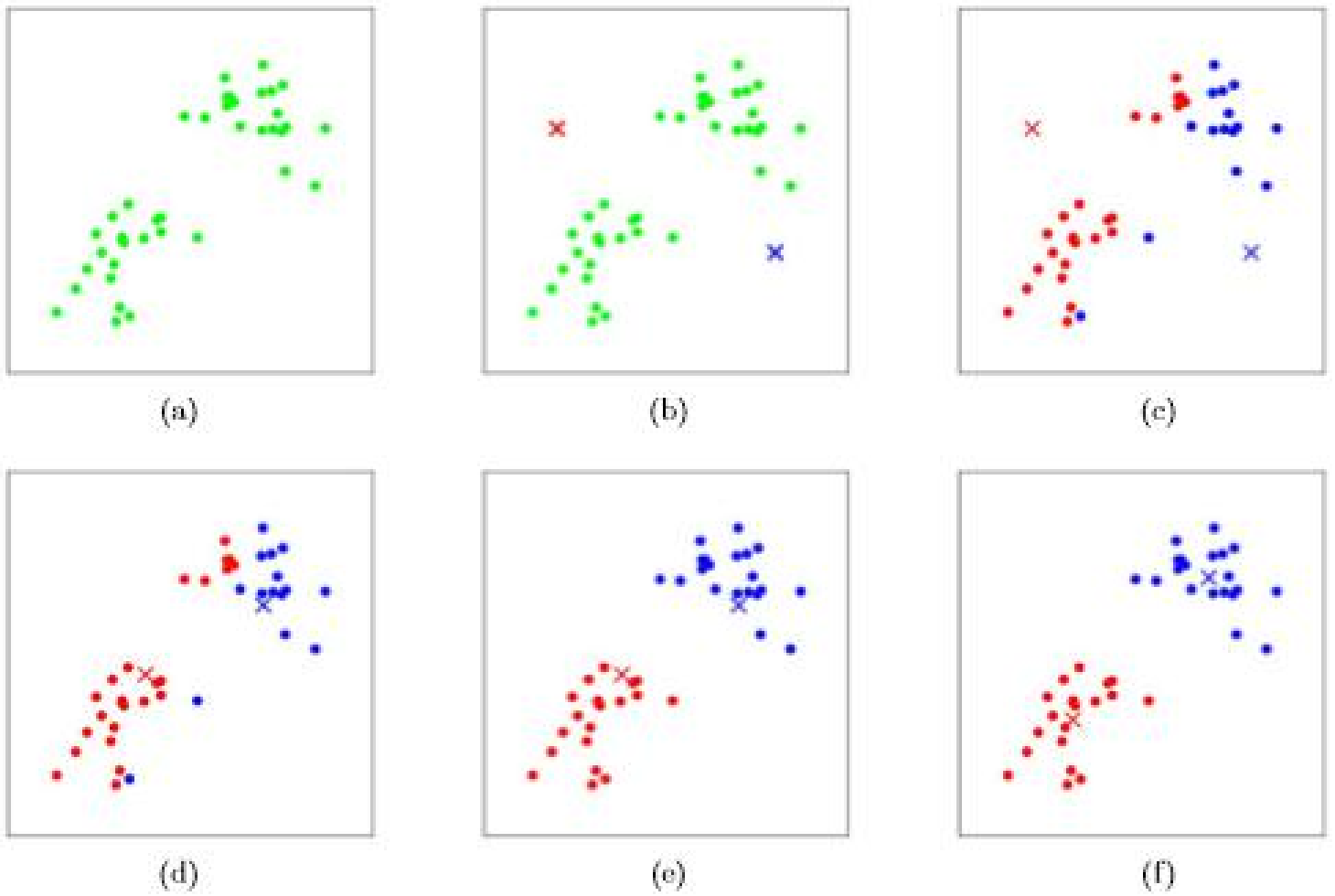
聚类方法目录

1. 划分方法 Partitioning Methods
2. 层次方法 Hierarchical Methods
3. 基于密度的方法 Density-Based Methods
4. 基于网格的方法 Grid-Based Methods
5. 方法的评价 Evaluation of Clustering

7.1.划分方法：K-Means （原型方法）



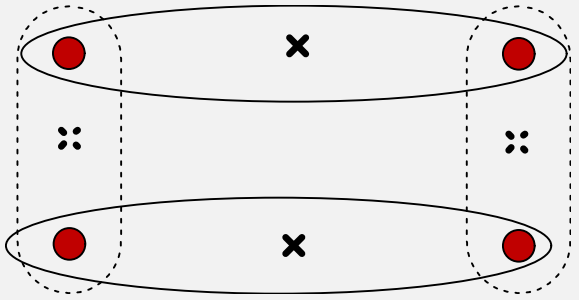
7.1.划分方法：K-Means较差初始值的例子



7.1.划分方法：K-Means 方法的特点

- 计算复杂度: $O(tkn)$, n 数据点数, k簇的数量, t迭代次数. 一般 $k, t << n$.

- 缺点：
  1. K-均值方法需要连续的n维数据空间；
     - 若是离散（标称）数据可以使用k-modes （k-众数）方法
     - k-medoids（k-中位数）可以在更广的数据集上使用
  2. 需要指明k的值, 受到离群点的影响；
  3. 对于非凸形状的簇不适合；
  4. 簇大小差距较大不适用（大簇尺寸与簇间距可比）；



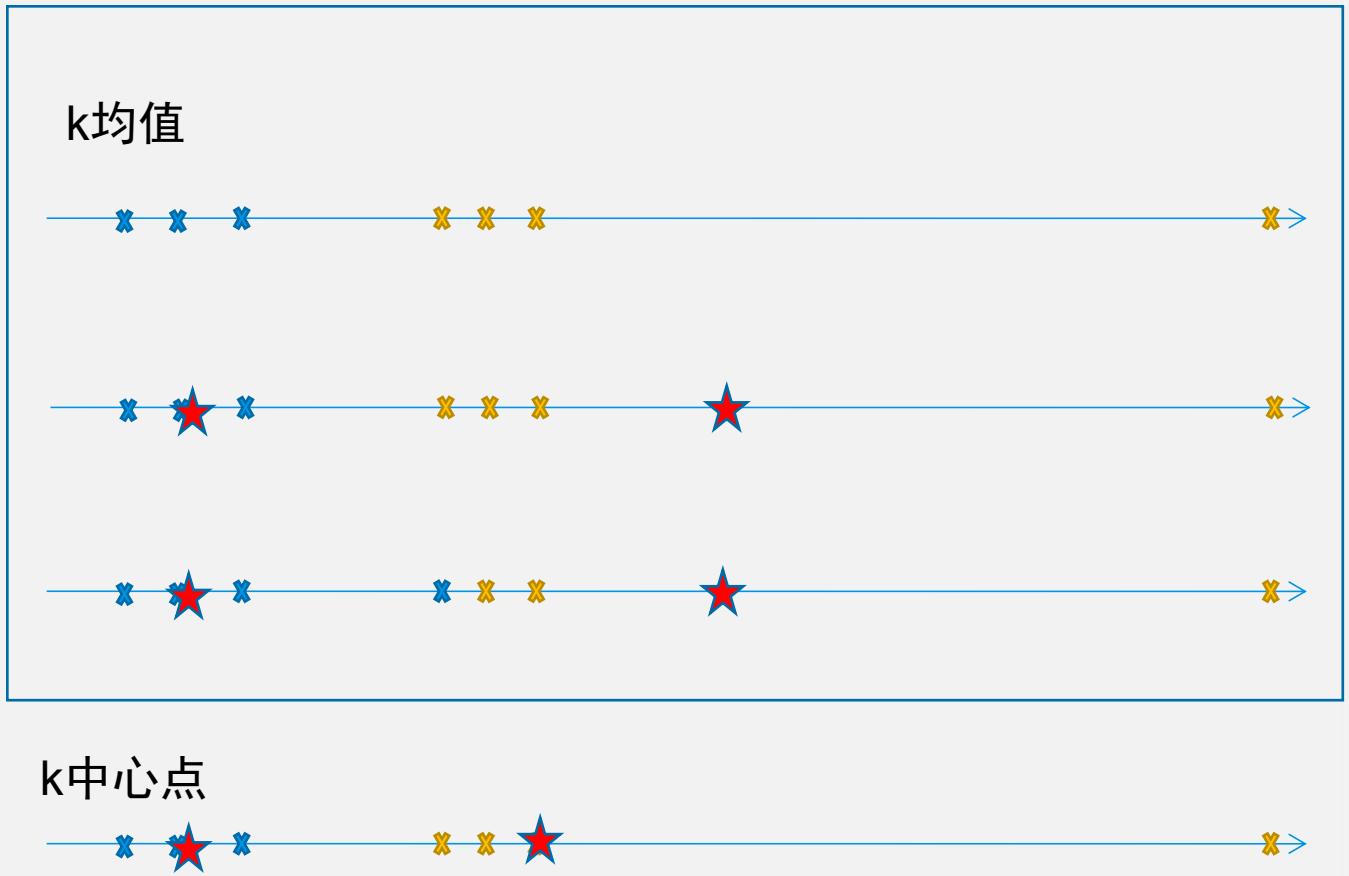
7.1.划分方法： K-Means与k-中心点

- 1，2，3，8，9，10，33
- 人工聚类：{1，2，3}{8，9，10} 离群点：{33}
- 初始：{1，2，3}{8，9，10，33}

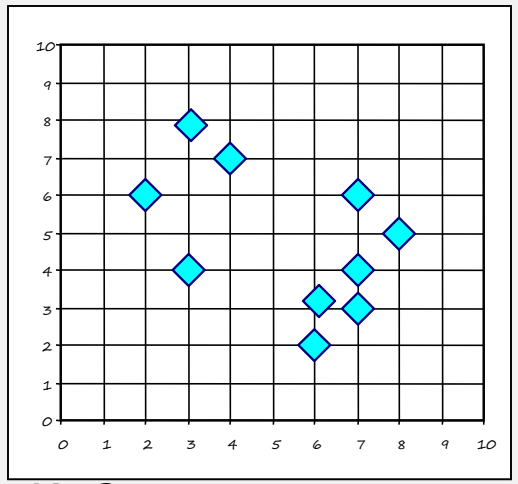
◘ K-中心点 K-Medoids（基于代表对象的技术）

- 簇1均值=2；簇2均值=15
- k-means下一轮：{1，2，3，8}{9，10，33}

- 簇1中心点=2；簇2中心点=10
- k-中心点下一轮：{1，2，3}{8，9，10，33}

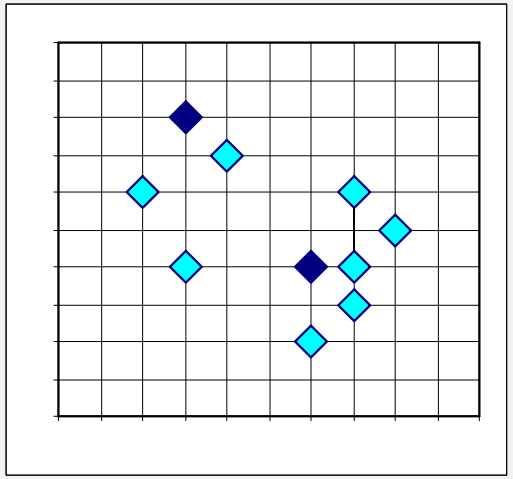


7.1 划分方法：PAM: 围绕中心点划分方法(Partitioning Around Medoids)



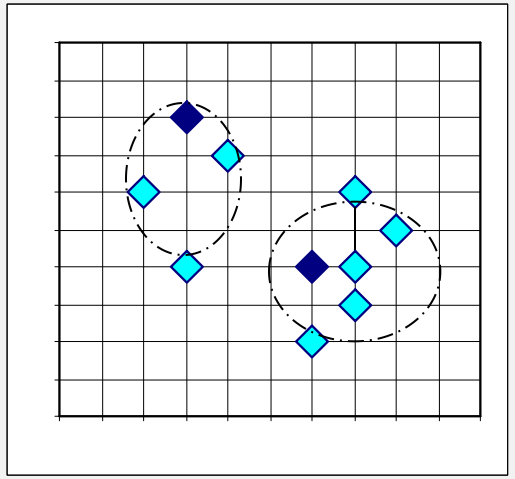
K=2

随机分配中心点

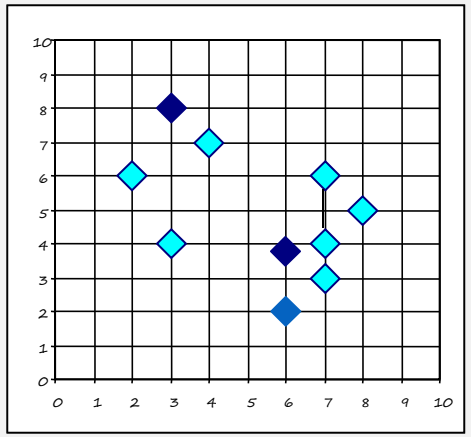


将所有数据点分配给最近的中心点

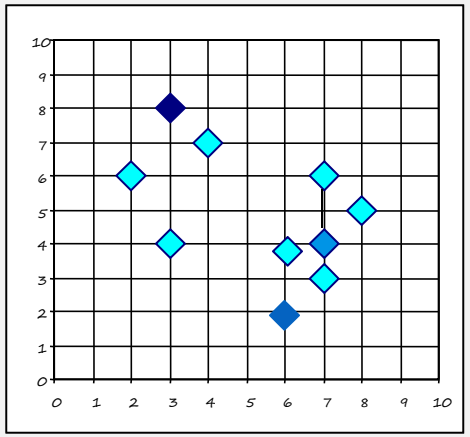
Total Cost = 20



随机选择一个非当前中心点,O_{ramdom}



计算所有点到当前中心点距离的变化



Total Cost = 26

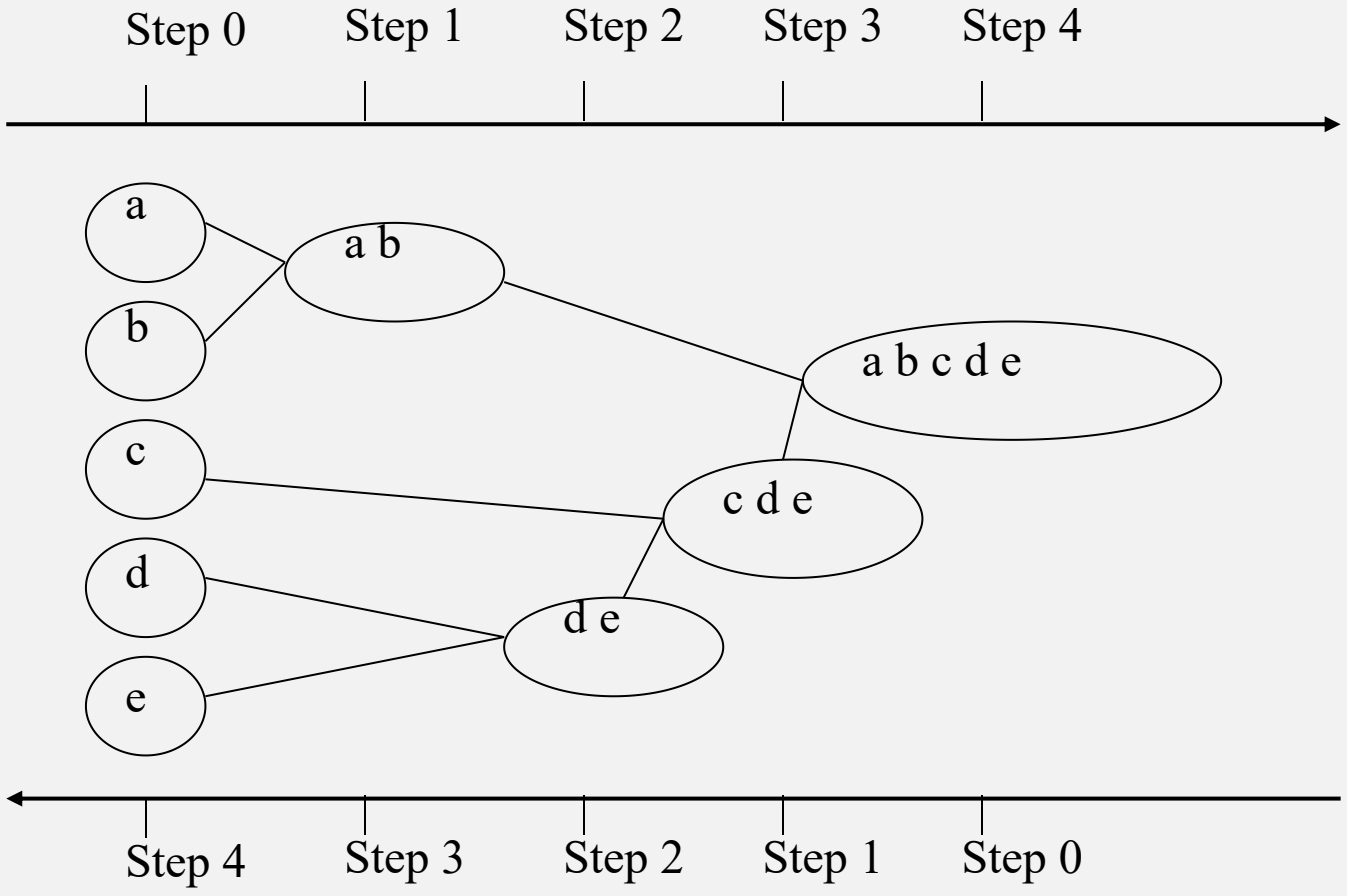
如果距离变小则改变中心点

7.1 划分方法：K-中心点方法

- *K-Medoids* Clustering:以具有代表性的数据点作为中心
  - *PAM* (Partitioning Around Medoids, 围绕中心点划分方法)
    - 缺点：计算复杂度高，对小的数据集效果好
- Efficiency improvement on PAM
  - *CLARA* 在一个抽样子集上解决上述问题
  - *CLARANS* 在几个抽样子集上解决上述问题

7.2. 层次方法

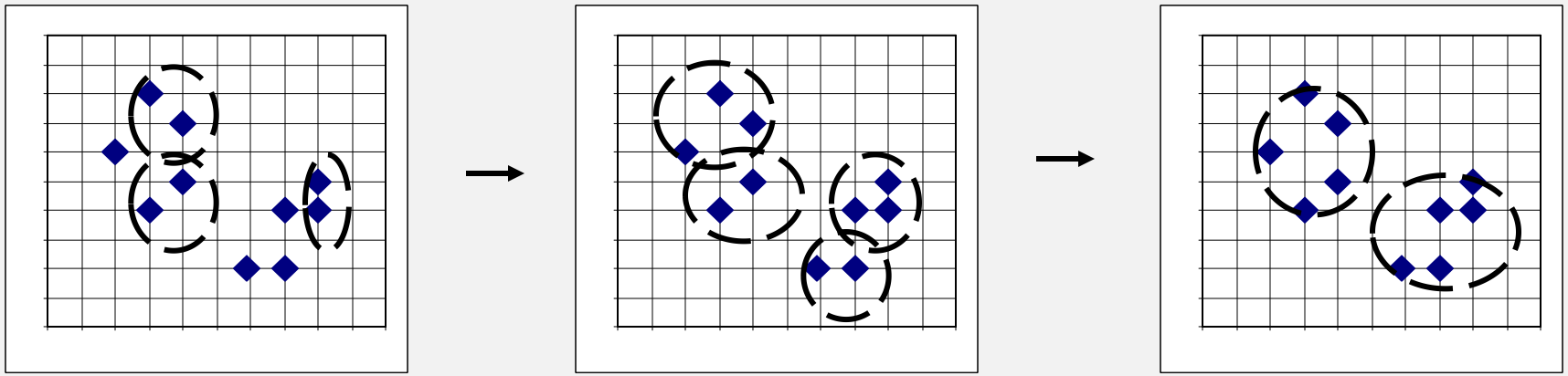
凝聚方法(AGNES)



分裂方法(DIANA)

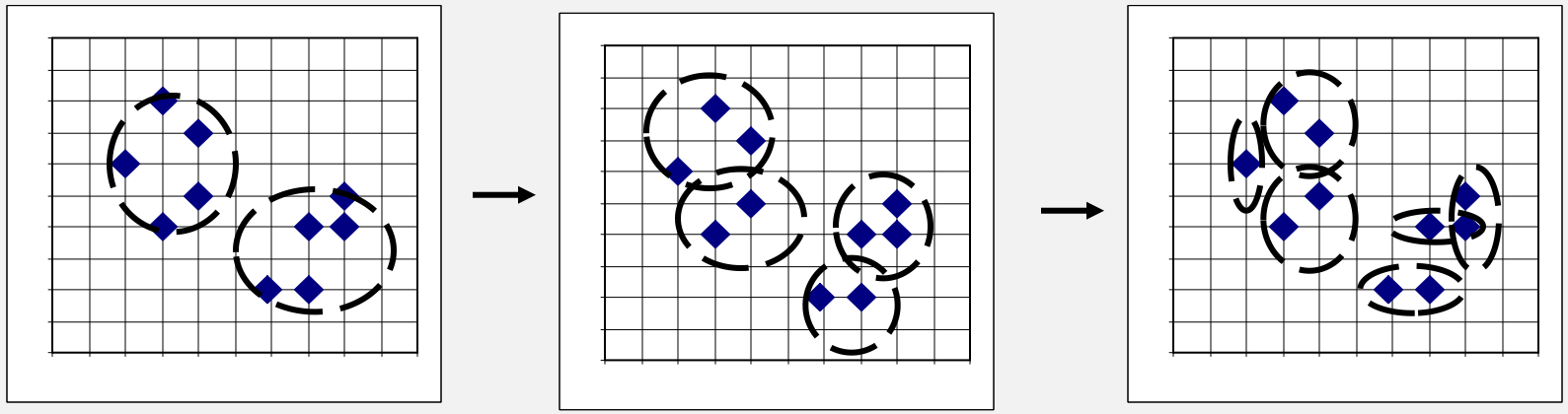
7.2 层次方法： 凝聚方法AGNES (AGglomerate NESting)

- 将最接近的点或者簇合并，最终得到符合要求的簇的集合

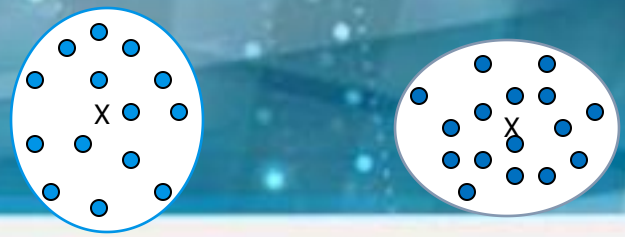


7.2 层次方法：分裂方法DIANA (Divisive ANAlysis)

- 分裂方法是凝聚方法的逆过程

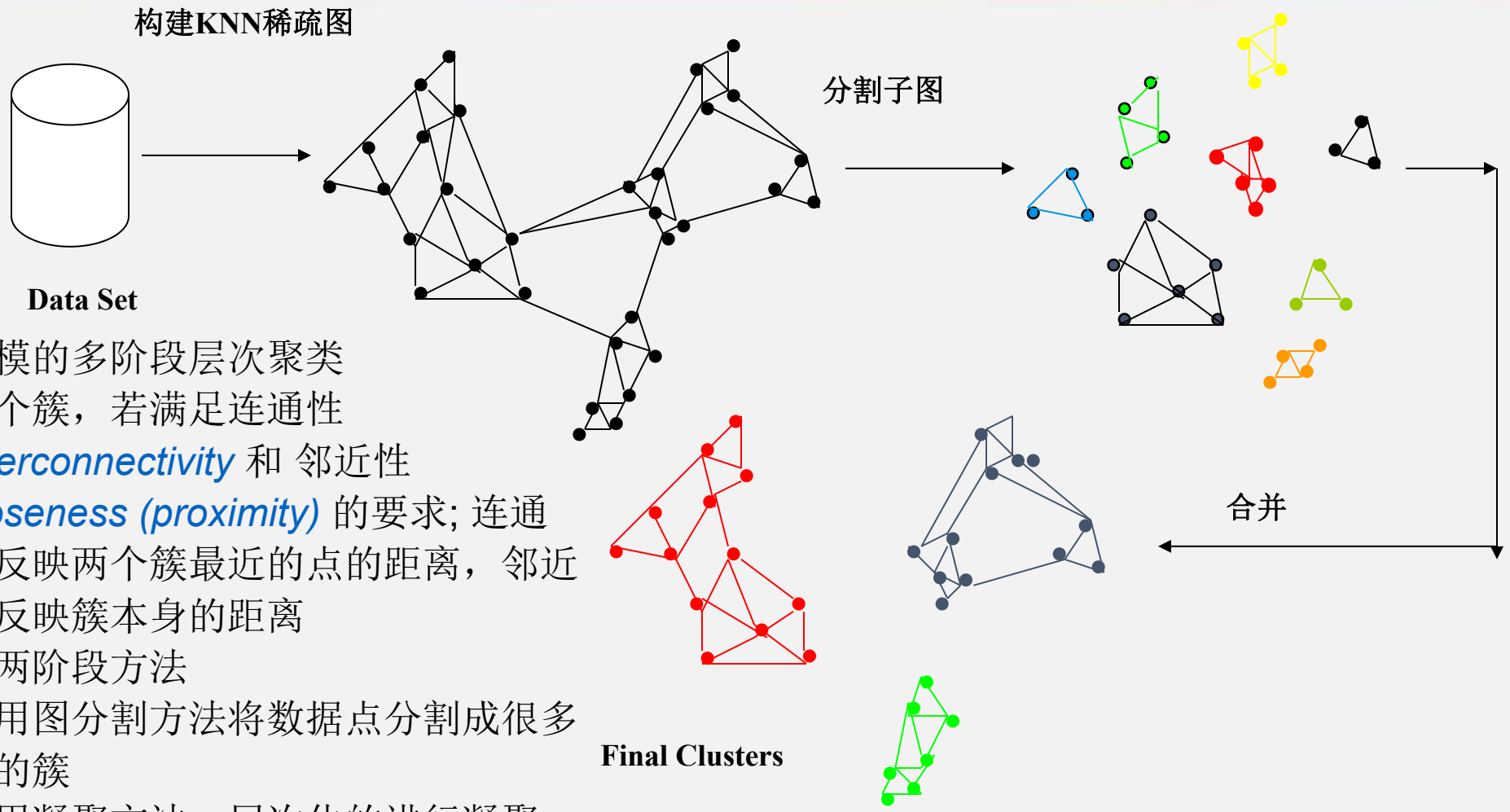


7.2簇间距离的计算



- Single link: 两个簇中相互距离最小的一对点之间的距离 dist(Ki, Kj) = min(tip, tjq)
- Complete link: 两个簇中相互距离最大的一对点之间的距离., dist(Ki, Kj) = max(tip, tjq)
- Average: 所有点对之间距离的平均值，dist(Ki, Kj) = avg(tip, tjq)
- Centroid: 两个簇中心点之间的距离，dist(Ki, Kj) = dist(Ci, Cj)
- Medoid: 两个簇中位点之间的距离，dist(Ki, Kj) = dist(Mi, Mj)

7.2层次方法：CHAMELEON算法



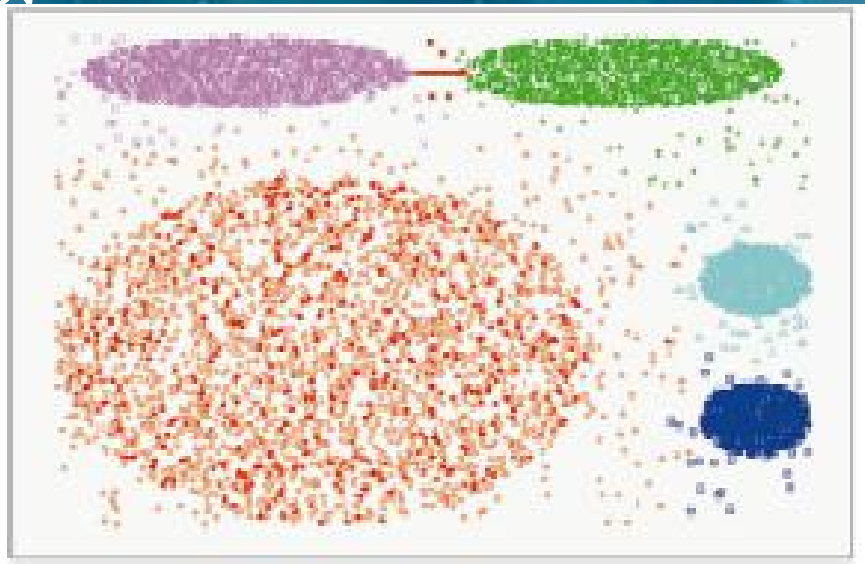
使用动态建模的多阶段层次聚类

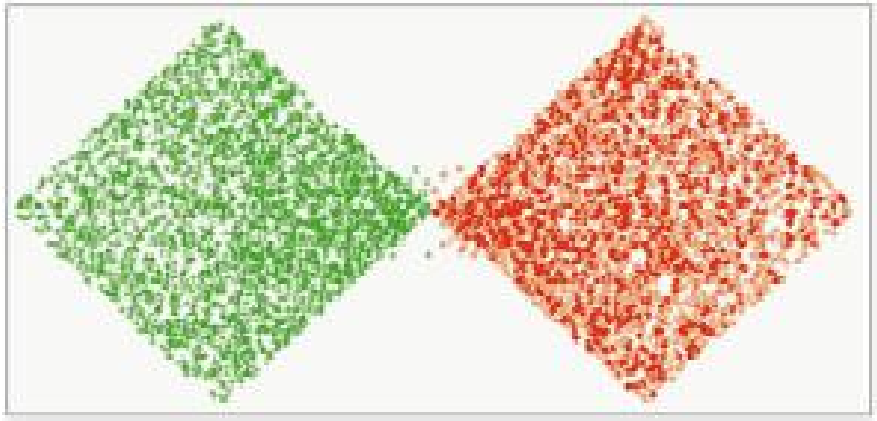
合并两个簇，若满足连通性 *interconnectivity* 和 邻近性 *closeness (proximity)* 的要求; 连通性反映两个簇最近的点的距离，邻近性反映簇本身的距离

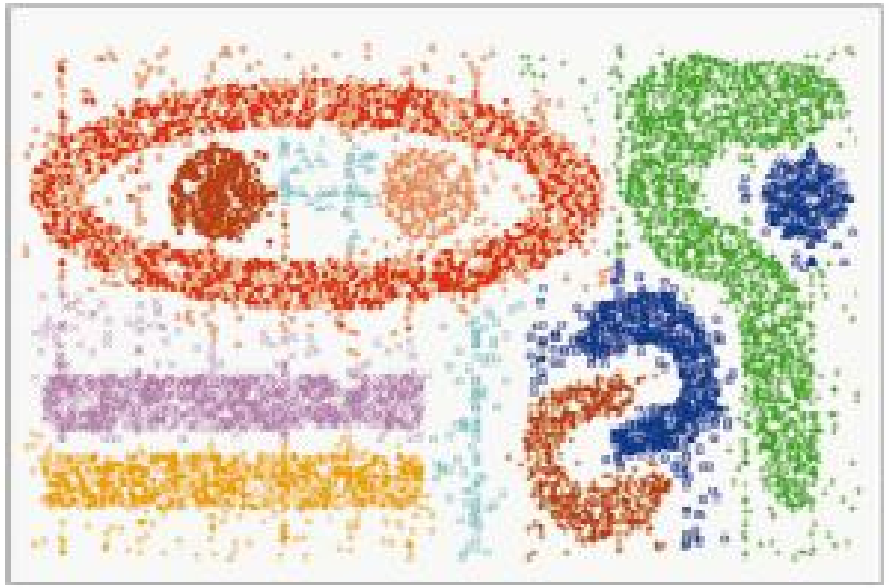
基于图形的两阶段方法

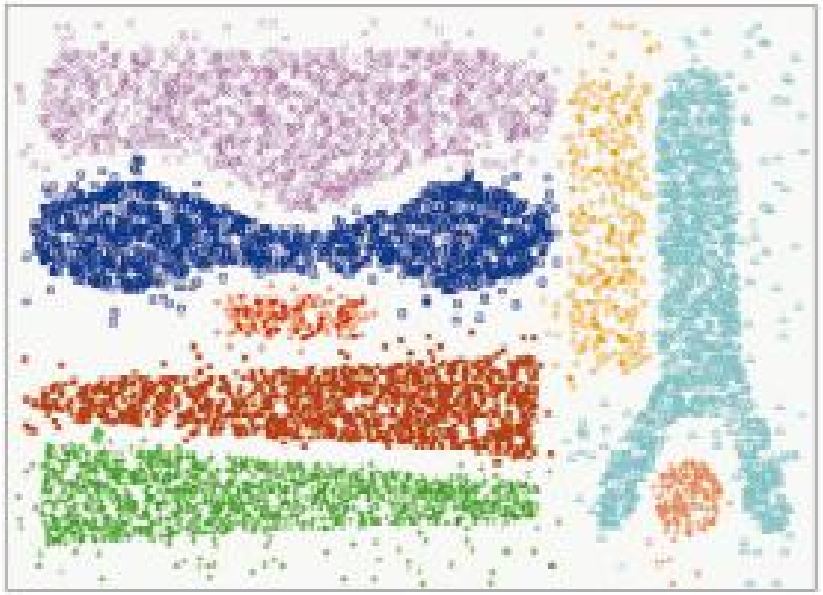
1. 利用图分割方法将数据点分割成很多小的簇
2. 利用凝聚方法，层次化的进行凝聚

7.2层次方法：CHAMELEON用于解决任意形状簇聚类问题



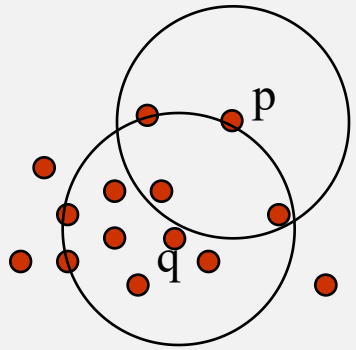






7.3 基于密度的方法

- *定义两个参数:*
  - *Eps*: 邻域半径
  - *MinPts*: 邻域半径范围内的最小点数
  - *点周围的密度可由上述两个值定义*
  - $N_{Eps}(p)$: {q belongs to D | dist(p,q) ≤ Eps}>MinPts
- p对于q “直接密度可达”需要满足下述条件：
  - *p在q的邻域半径内，*
  - *此范围内数据点数量大于最小点数*

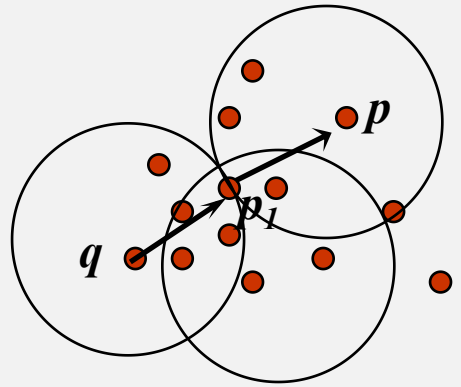


MinPts = 6

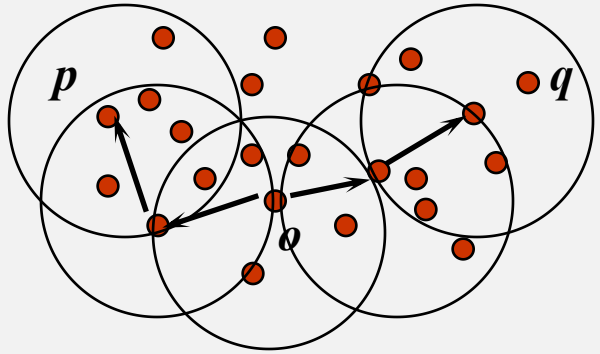
Eps = 1 cm

7.3 基于密度的方法

- 密度可达关系的传递：密度可达
  - 若 p 对于q来说是密度可达的（ density-reachable ）那么存在一系列有序的点的集合 $p_1, \ldots, p_n, p_1 = q, p_n = p$ 满足 p_{i+1} 对于 p_i 来说直接密度可达。

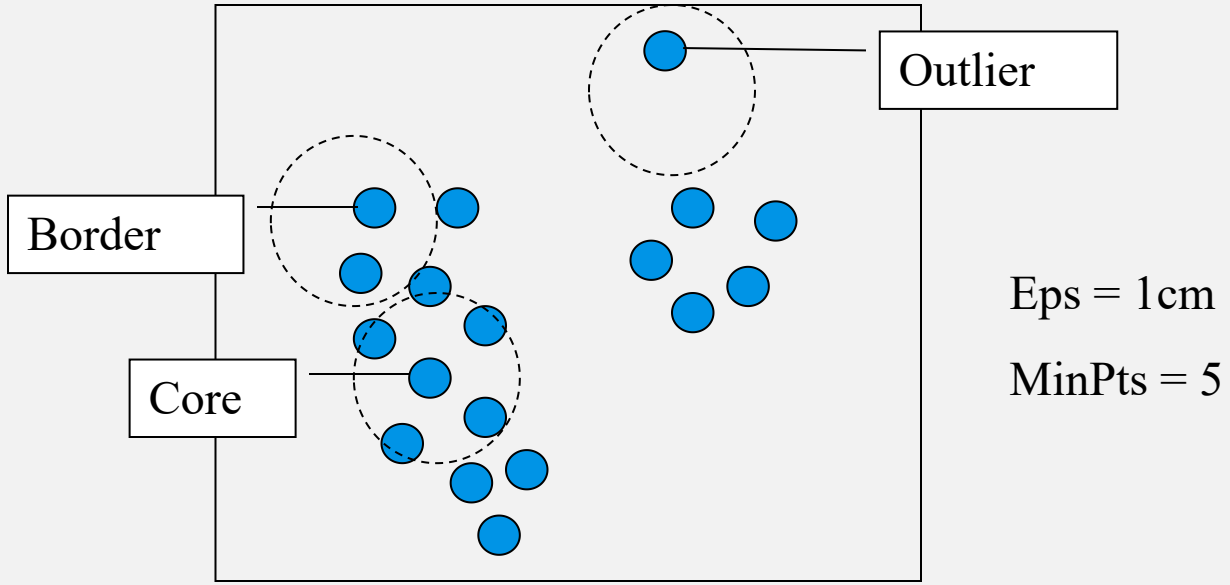


- 密度可达关系的传递：密度相连
  - 若点 p 和 q 是密度相连的，那么存在点o, p 和 q 对于 o 来说密度可达。



7.3 基于密度的方法： DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- DBSCAN算法：
- 目标： 找到密度相连的点的最大集合
- 可以发现任意形状的簇



7.3 基于密度的方法： DBSCAN

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
邻域参数 $(\epsilon, MinPts)$.

过程：

初始化核心对象集合: $\Omega = \varnothing$
for $j = 1, 2, \ldots, m$ **do**
  确定样本 $\boldsymbol{x}_j$ 的 ϵ-邻域 $N_\epsilon(\boldsymbol{x}_j)$;
  if $|N_\epsilon(\boldsymbol{x}_j)| \geqslant MinPts$ **then**
    将样本 $\boldsymbol{x}_j$ 加入核心对象集合: $\Omega = \Omega \bigcup \{\boldsymbol{x}_j\}$
  end if
end for
初始化聚类簇数: $k = 0$
初始化未访问样本集合: $\Gamma = D$
while $\Omega \neq \varnothing$ **do**
  记录当前未访问样本集合: $\Gamma_{\text{old}} = \Gamma$;
  随机选取一个核心对象 $\boldsymbol{o} \in \Omega$, 初始化队列 $Q = < \boldsymbol{o} >$;
  $\Gamma = \Gamma \setminus \{\boldsymbol{o}\}$;
  while $Q \neq \varnothing$ **do**
    取出队列 Q 中的首个样本 $\boldsymbol{q}$;
    if $|N_\epsilon(\boldsymbol{q})| \geqslant MinPts$ **then**
      令 $\Delta = N_\epsilon(\boldsymbol{q}) \bigcap \Gamma$;
      将 Δ 中的样本加入队列 Q;
      $\Gamma = \Gamma \setminus \Delta$;
    end if
  end while
  $k = k + 1$, 生成聚类簇 $C_k = \Gamma_{\text{old}} \setminus \Gamma$;
  $\Omega = \Omega \setminus C_k$
end while

输出： 簇划分 $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

7.3 基于密度的方法： DBSCAN

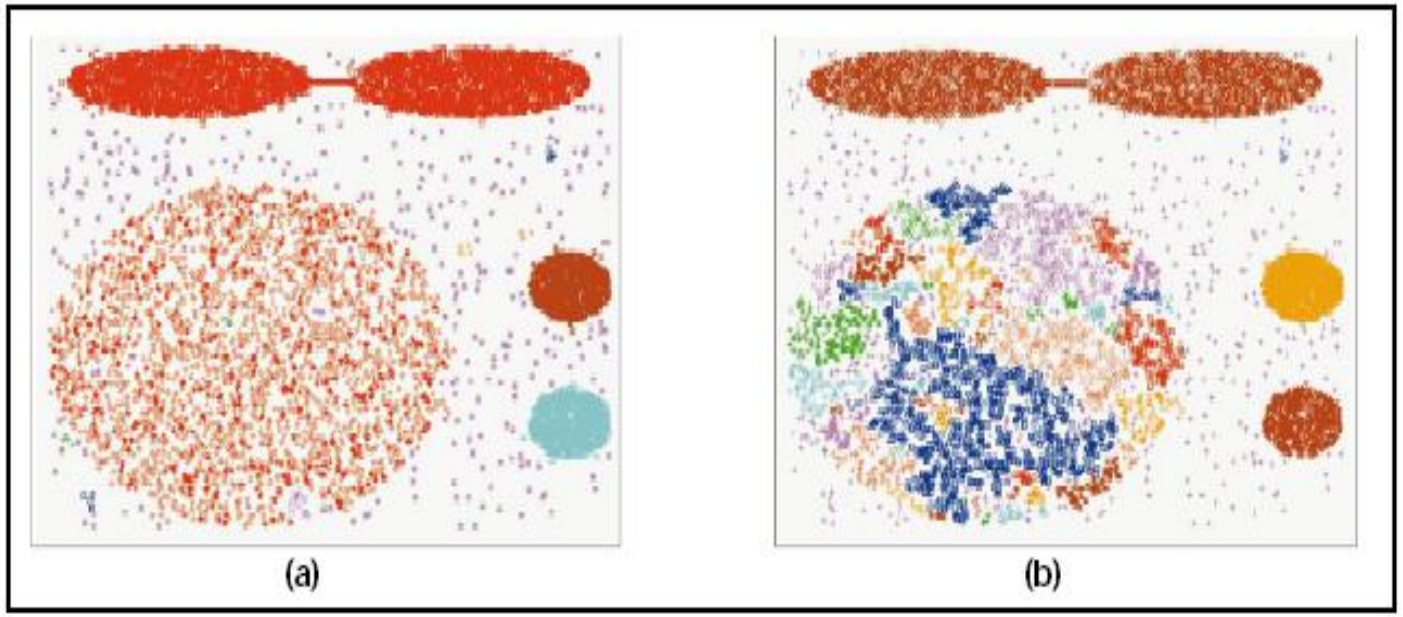
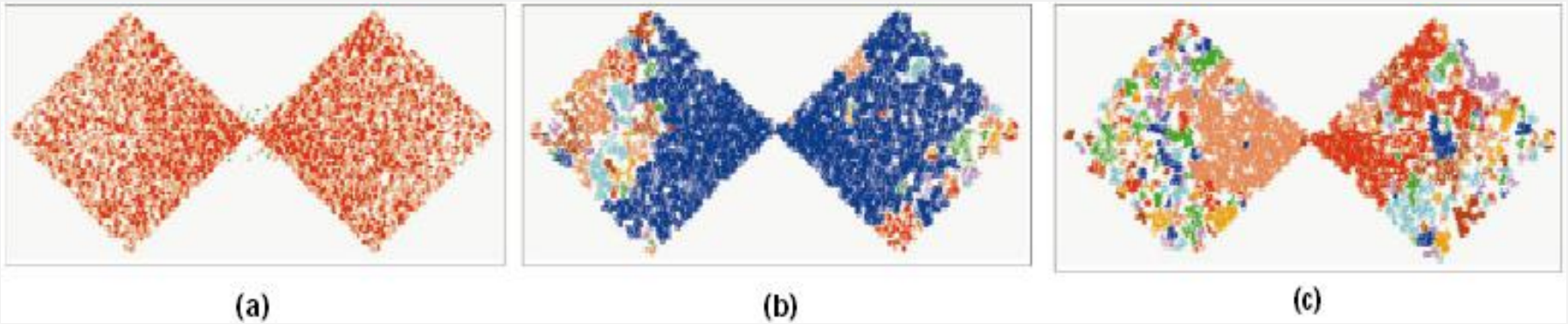


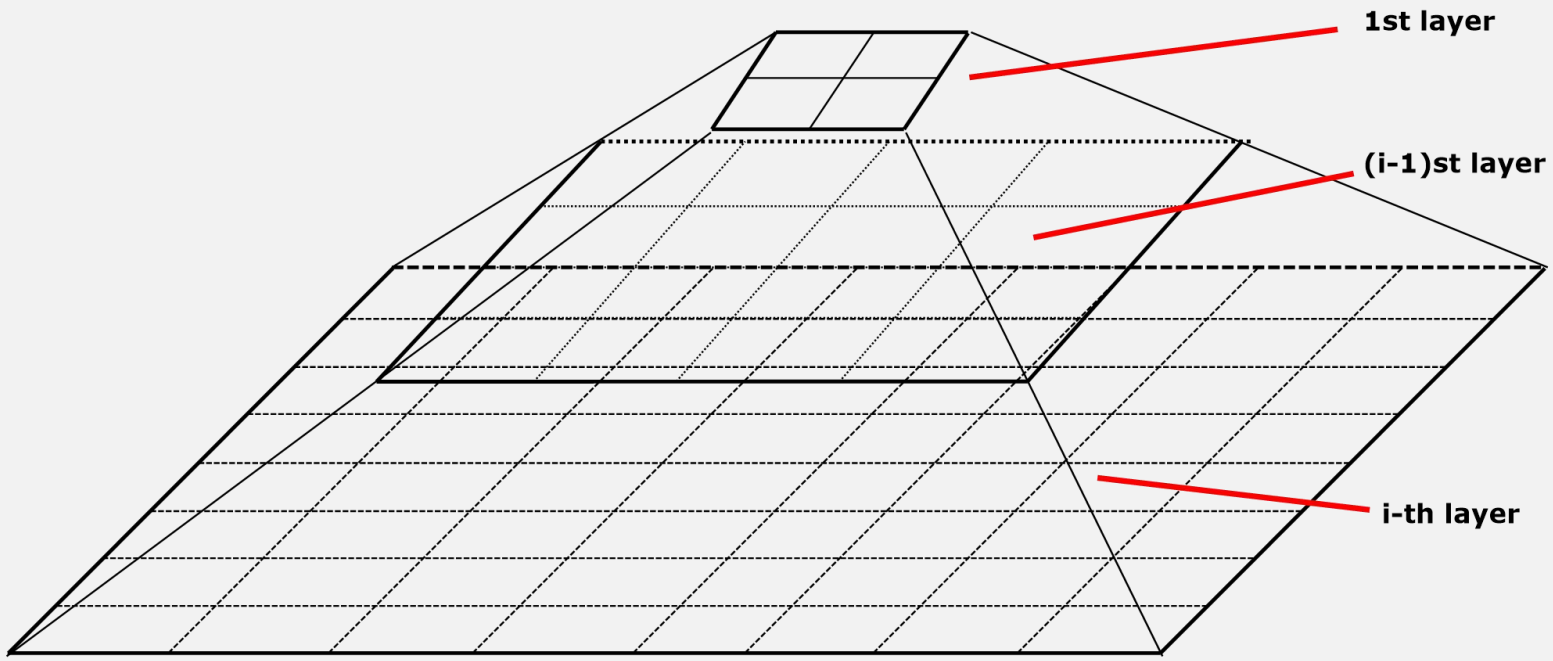
Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

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



7.4.基于网格的方法

- 将特征平面分为网格，计算网格的归属



7.4.基于网格的方法

■（额外的）特点：

- 删除不相关（无数据）的区域
- 完成一个层次的聚类后，可继续细分网格

■优点 Advantages:

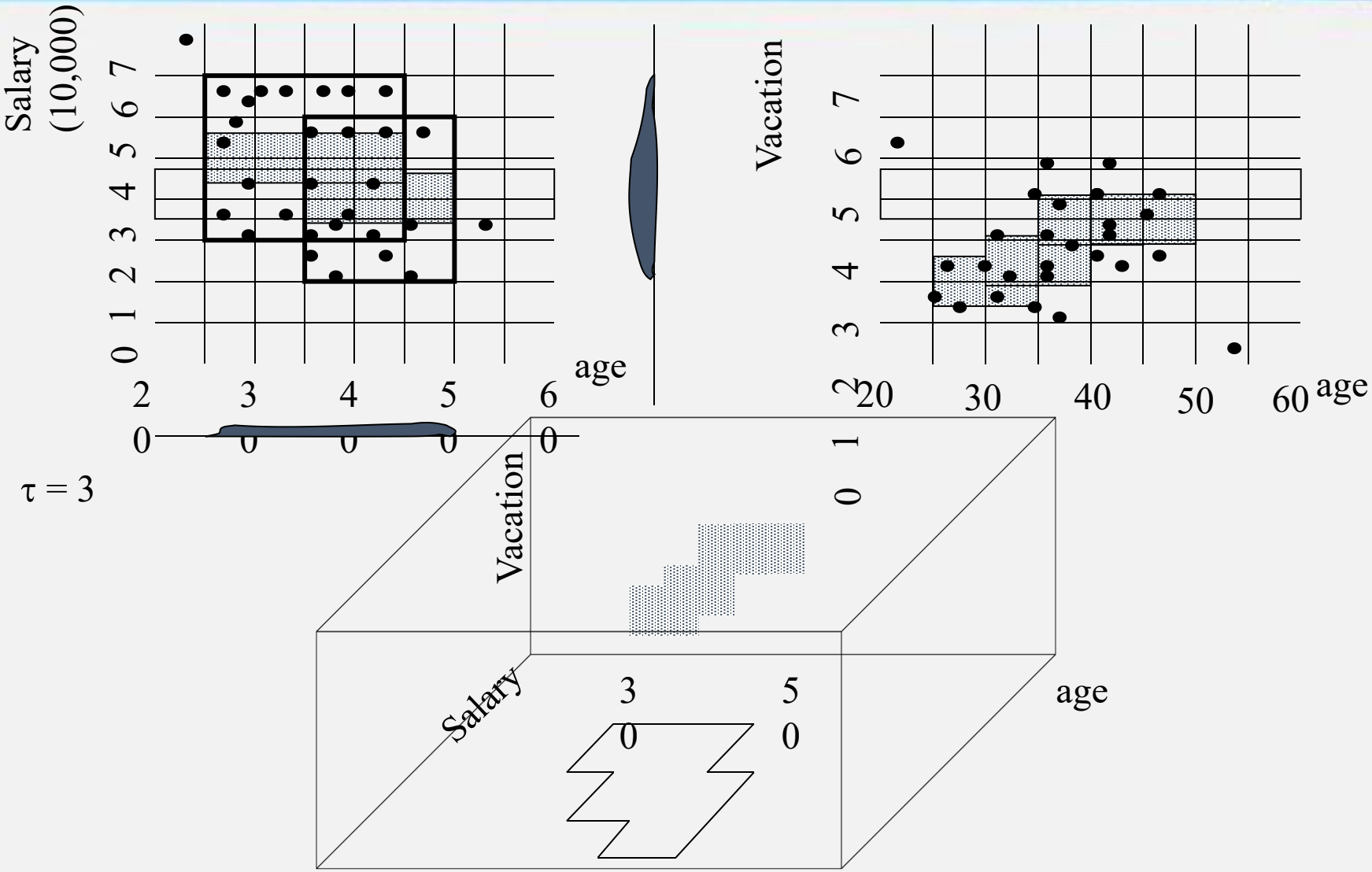
- 可以只计算查询部分，可以并行计算
- 可以控制精度

■缺点 Disadvantages:

- 边界是与坐标平行的

7.4 CLIQUE 方法(Clustering In QUEst)

- kmeans等基于距离计算的方法遇到的问题：
- 数据在子空间中簇的性质可能被更高维空间距离所掩盖



24

7.4 CLIQUE: 主要步骤

CLIQUE算法：

1. 在特征空间做划分，确定所有数据点所属的cell
2. 利用 Apriori 发现哪些维度区间经常一起出现
3. 识别簇：
   - 识别频繁子空间
   - 合并频繁子空间.
4. 给出这些簇的最小描述

优点
- 自动找到最高维度的簇
- 与特征空间的总维度无关（与商品数量无关）
- 可扩展性好

缺点
- 受到划分粒度的影响

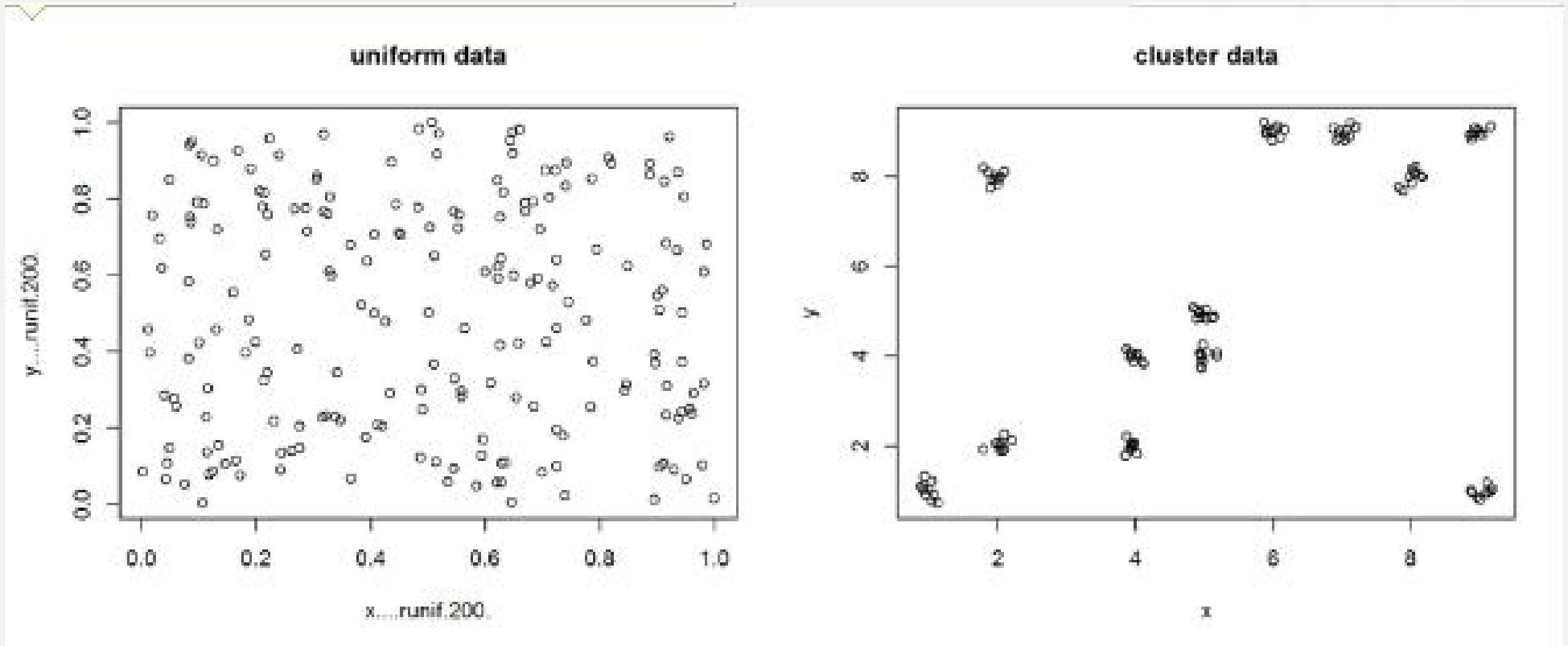
7.5聚类的评价——1.是否可聚类

- 完全不存在簇规律的数据集也可以为算法计算，从而得出结果，但这样的结果没有意义；
- 所以首先评价聚类的意义，再评价聚类的质量
- Hopkins Static 霍普金斯统计量 检验变量的空间随机性（有无内在的规律性）
  - 计算特征空间中数据点的分布与均匀分布的差距
  - 均匀的抽取n个点 $p_1, \ldots, p_n$, 于数据集D所处的特征空间. 对于点 p_i, 找到它与数据集D中最近的邻居： $x_i = min\{dist\ (p_i, v)\}$
  - 均匀的抽取n个点 $q_1, \ldots, q_n$, 于数据集D. 对于点 q_i, 找到它最近的邻居 $D - \{q_i\}$: $y_i = min\{dist\ (q_i, v)\}$
  - 计算 Hopkins Statistic:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

  - 如果数据集D 服从均匀分布，那么 $\sum x_i$ 和 $\sum y_i$ 将很接近， H 趋近于 0.5. 如果 D 有可被聚类的规律, H趋近于0.

7.5聚类的评价——1.是否可聚类



7.5 聚类的评价——2.确定簇的数量

- 经验方法
  - # of clusters ≈√n/2 for a dataset of n points
- 肘方法
  - 簇内方差和关于簇的数量曲线的拐点。（二阶导为0）
- 交叉验证方法
  - 将数据集分成 m 份
  - 由其中的 m – 1 份参与算法计算得到聚类模型
  - 由剩余的1份验证聚类质量
    - E.g., 按照距离的定义找到验证集中的数据点所属于的簇, 计算距离平方和。
    - 重复 m 次, 计算m次计算的均值
  - 比较不同的簇数选择下，验证集距离平方和的数值

7.5 聚类的评价——3.测定簇的质量

- 外在方法 （要求已知事实真相）

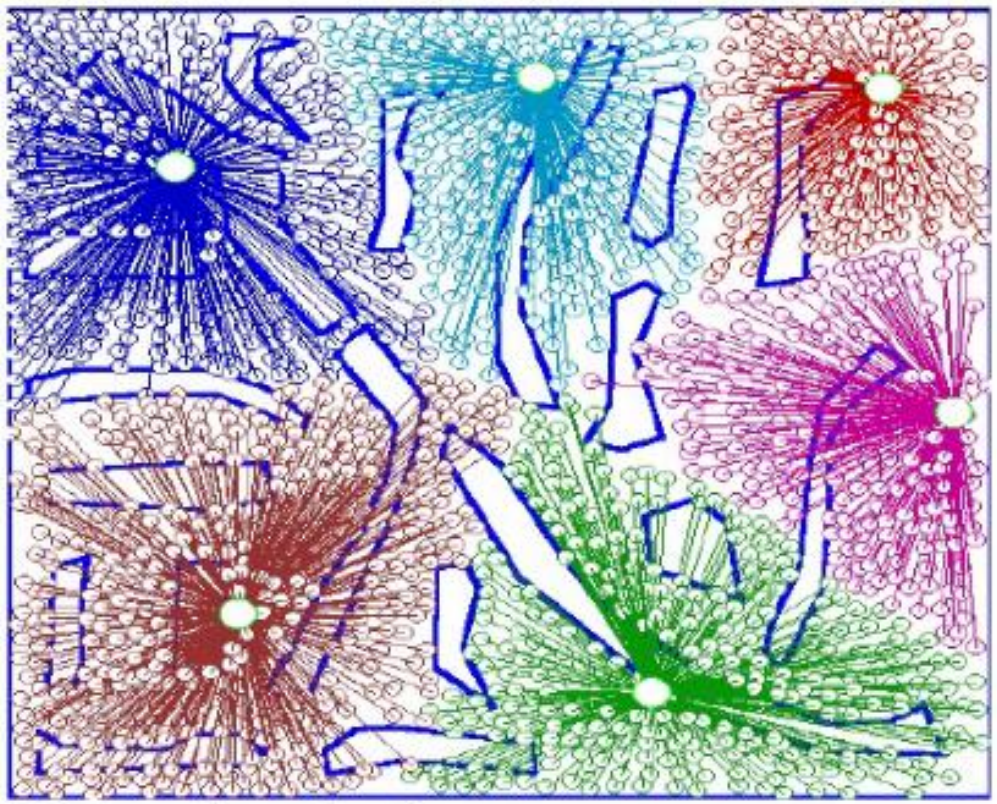
  簇的同质性 （Cluster homogeneity）: 簇的纯洁程度 ——对比分类问题的精度
  簇的完整性 （Cluster completeness）: 簇的完整性——对比分类问题的召回率

  BCubed精度：在算法结果的簇上统计每个数据点与该簇所有节点是否属于一个真实的类
  BCubed召回率：在真实的簇上统计每个数据点与该簇所有节点是否属于一个真实的类

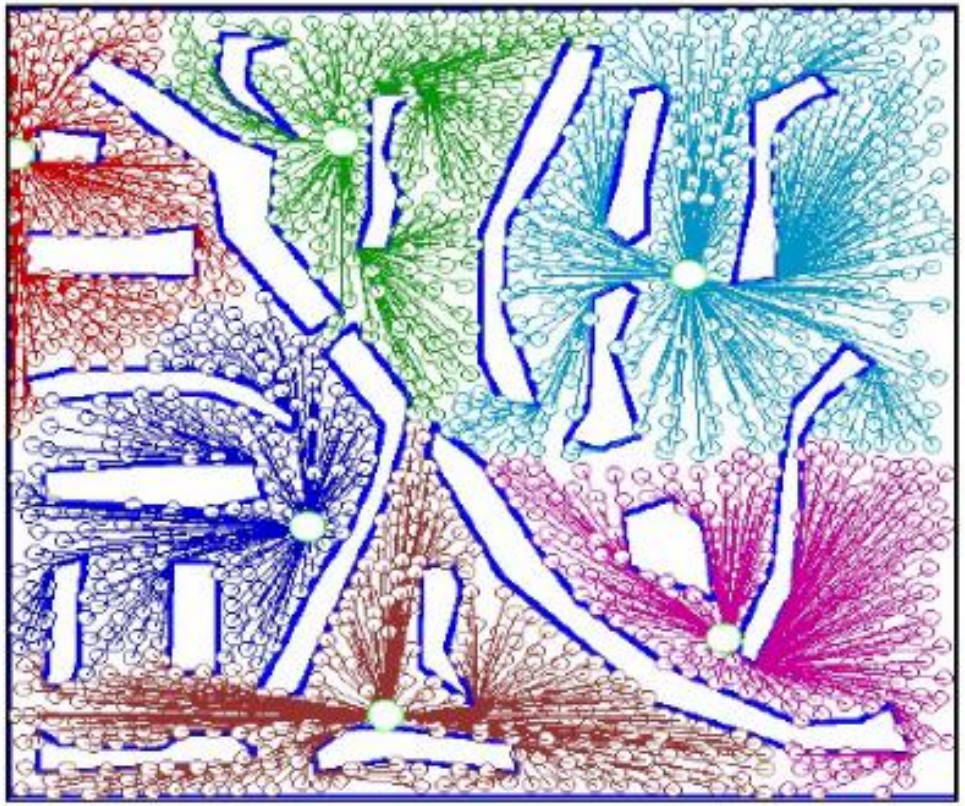
- 内在方法 （未知事实真相）

  轮廓系数 （silhouette coefficient）：对于每个对象o，a表示o与o所属簇的其他对象之间的平均距离，b表示o与不属于o所在簇的其他对象之间的平均距离
  s= (b-a)/max{a,b}

7.*带障碍的聚类问题

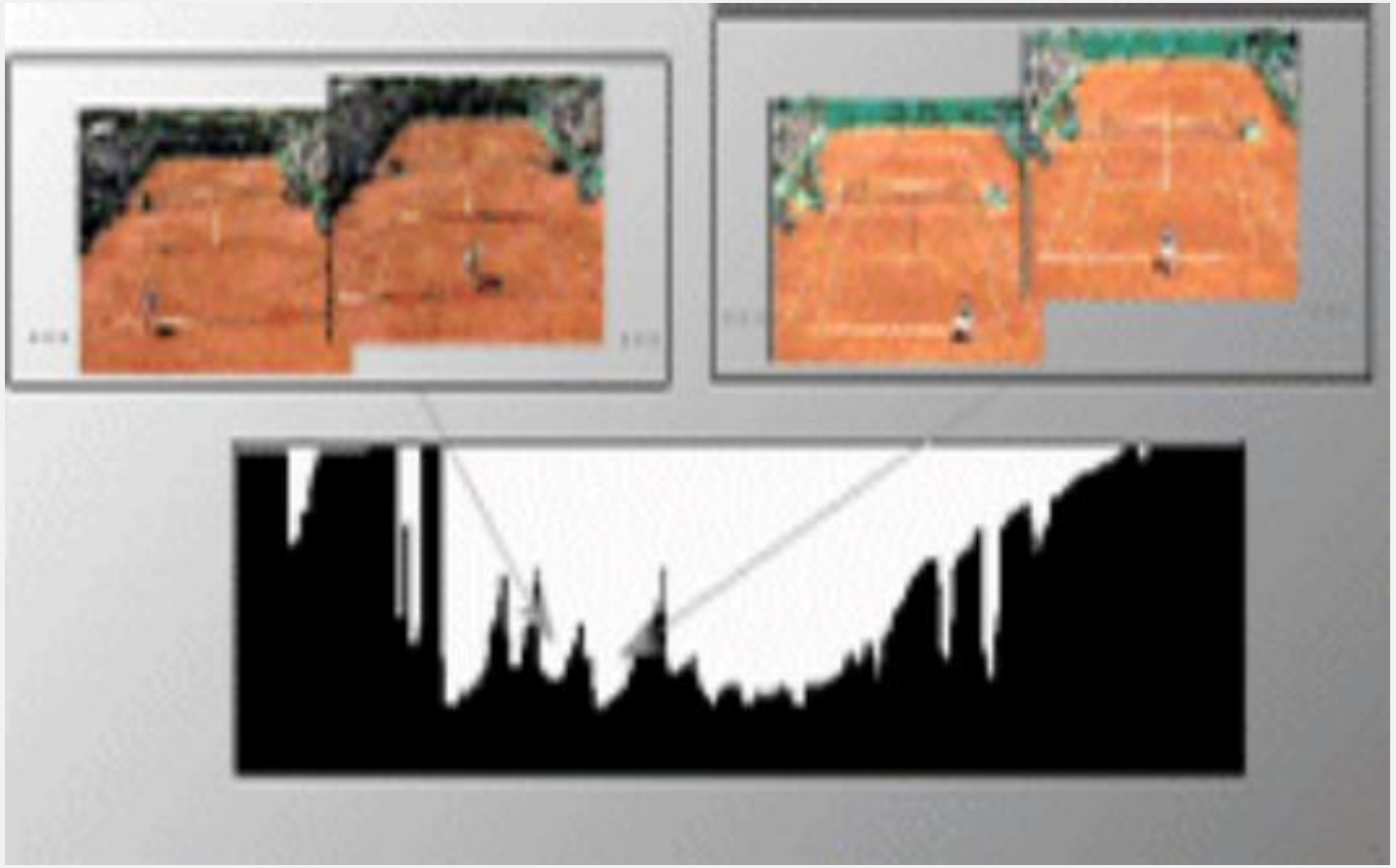


Not Taking obstacles into account



Taking obstacles into account

7.*更多的应用——数据转换到特征空间



聚类实验

1. 问题描述：本实验使用高斯分布生成理论数据集，进行聚类

下载FTP中的DBSCAN的样例代码：plot_dbscan.py

内容1：运行代码了解聚类算法的基本作用。

内容2：查看min_samples参数在不同取值下的结果，理解参数含义。

内容3：将DBSCAN算法更换为KMeans算法，观察区别。

内容4：总结原型聚类方法KMeans和密度聚类方法DBSCAN的区别。