大数据挖掘与统计学习

软件工程系
文化遗产数字化国家地方工程联合中心
可视化技术研究所
张海波
讲师/博士(后)

什么是聚类分析？

- 聚类分析是根据“物以类聚”的道理，对样本或指标进行分类的一种多元统计分析方法，它们讨论的对象是大量的样本，要求能合理地按各自的特性进行合理的分类，没有任何模式可供参考或依循，即在没有先验知识的情况下进行的。

聚类分析的基本思想

- 基本思想是认为研究的样本或变量之间存在着程度不同的相似性（亲疏关系）。
- 根据一批样本的多个观测指标，找出一些能够度量样本或变量之间相似程度的统计量，以这些统计量作为分类的依据，把一些相似程度较大的样本（或指标）聚合为一类，把另外一些相似程度较大的样本（或指标）聚合为一类，直到把所有的样本（或指标）都聚合完毕，形成一个由小到大的分类系统。

聚类分析无处不在

- 谁经常光顾商店，谁买什么东西，买多少？
- 按会员卡记录的光临次数、光临时间、性别、年龄、职业、购物种类、金额等变量分类
- 这样商店可以……
- 识别顾客购买模式（如喜欢一大早来买酸奶和鲜肉，习惯周末时一次性大采购）
- 刻画不同的客户群的特征

聚类分析无处不在

- 挖掘有价值的客户，并制定相应的促销策略：
  - 如，对经常购买酸奶的客户
  - 对累计消费达到12个月的老客户
- 针对潜在客户派发广告，比在大街上乱发传单命中率更高，成本更低！

聚类分析无处不在

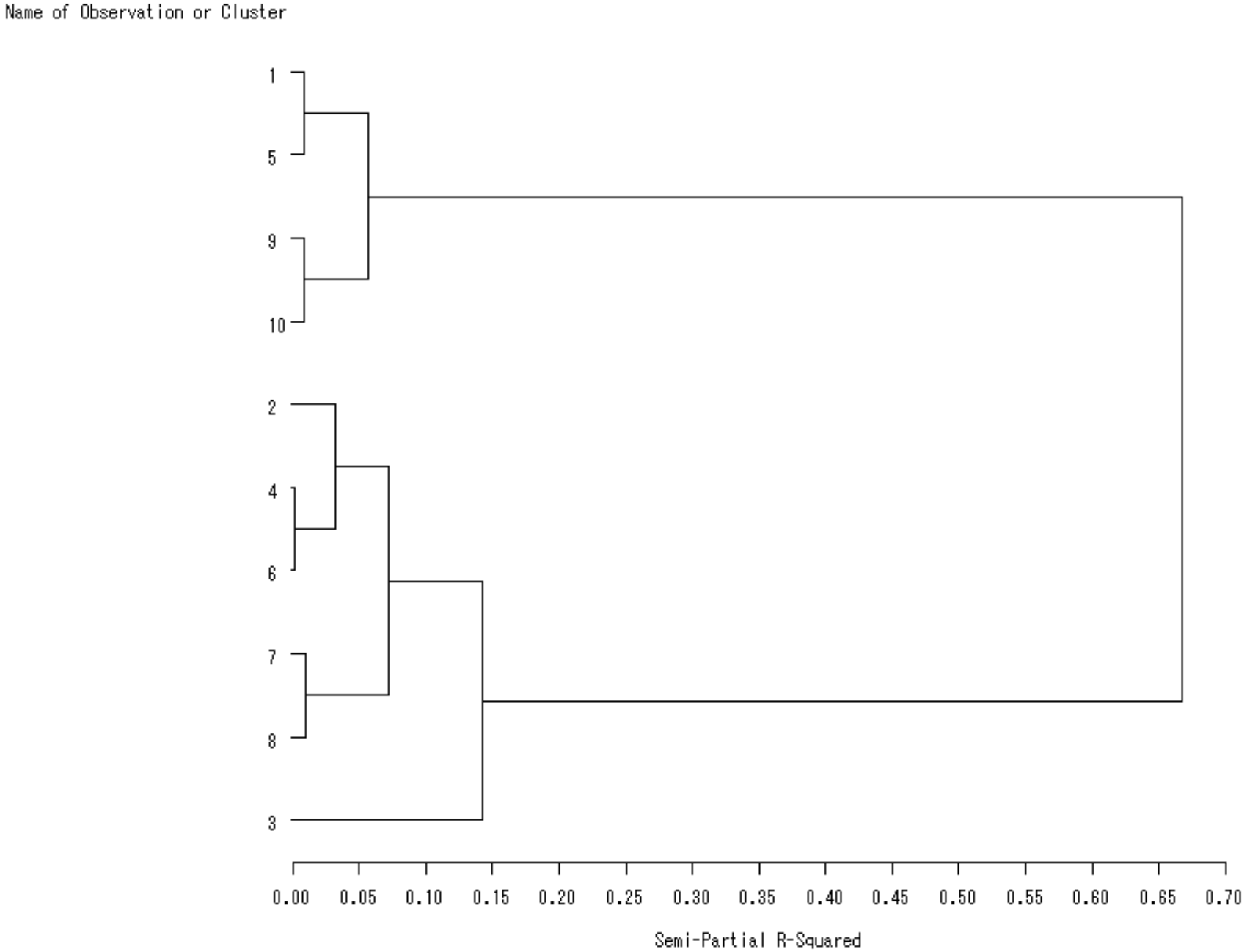
- 谁是银行信用卡的黄金客户？
  - 利用储蓄额、刷卡消费金额、诚信度等变量对客户分类，找出“黄金客户”！
  - 这样银行可以.....
  - 制定更具吸引力的服务，留住客户！比如：
    - 一定额度和期限的免息透支服务！
    - 赠送百盛的贵宾打折卡！
    - 在他或她生日的时候送上一个小蛋糕！

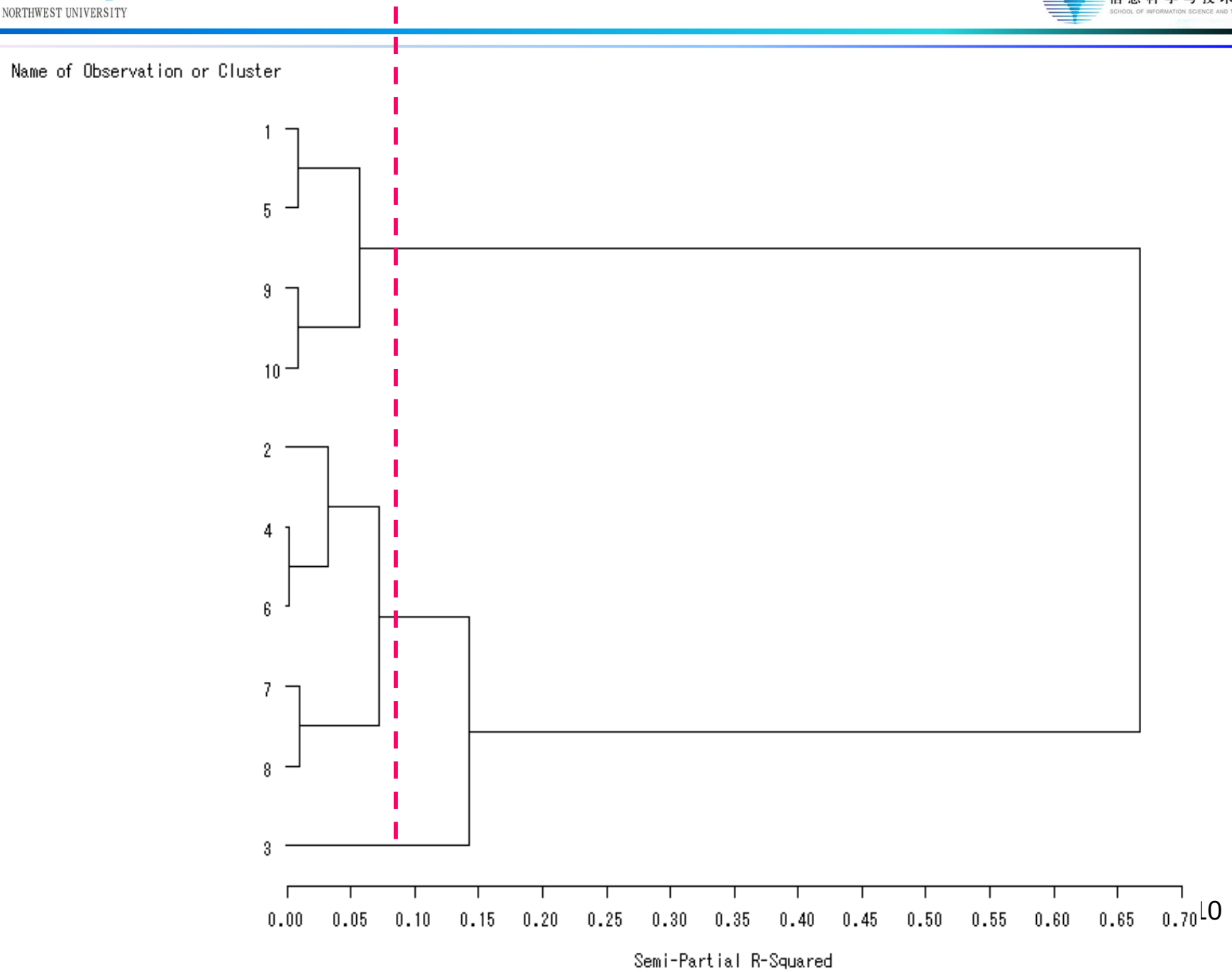
聚类的应用领域

- 经济领域：
  - 帮助市场分析人员从客户数据库中发现不同的客户群，并且用购买模式来刻画不同的客户群的特征。
  - 谁喜欢打国际长途，在什么时间，打到那里？
  - 对住宅区进行聚类，确定自动提款机ATM的安放位置
  - 股票市场板块分析，找出最具活力的板块龙头股
  - 企业信用等级分类
  - ……
- 生物学领域
  - 推导植物和动物的分类；
  - 对基因分类，获得对种群的认识
- 数据挖掘领域
  - 作为其他数学算法的预处理步骤，获得数据分布状况，集中对特定的类做进一步的研究

例　对10位应聘者做智能检验。3项指标X，Y和Z分别表示数学推理能力、空间想象能力和语言理解能力。得分如下，选择合适的统计方法对应聘者进行分类。

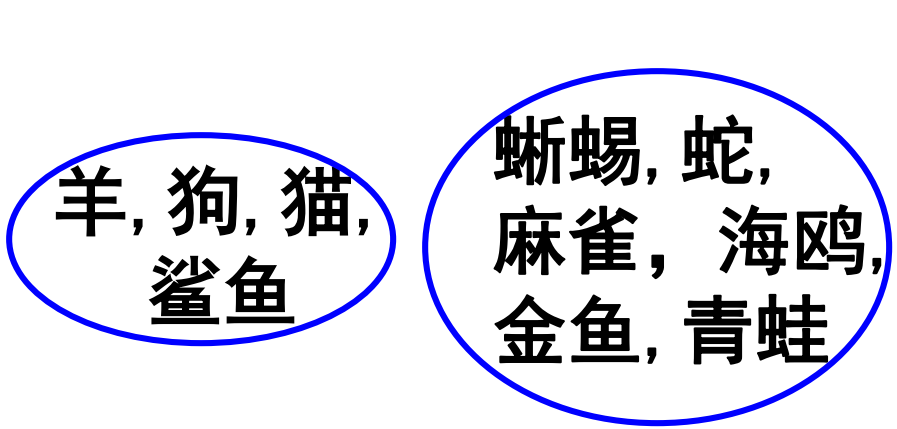
应聘者	1	2	3	4	5	6	7	8	9	10
X	28	18	11	21	26	20	16	14	24	22
Y	29	23	22	23	29	23	22	23	29	27
Z	28	18	16	22	26	22	22	24	24	24


Name of Observation or Cluster
1
5
9
10
2
4
6
7
8
3
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.70
Semi-Partial R-Squared

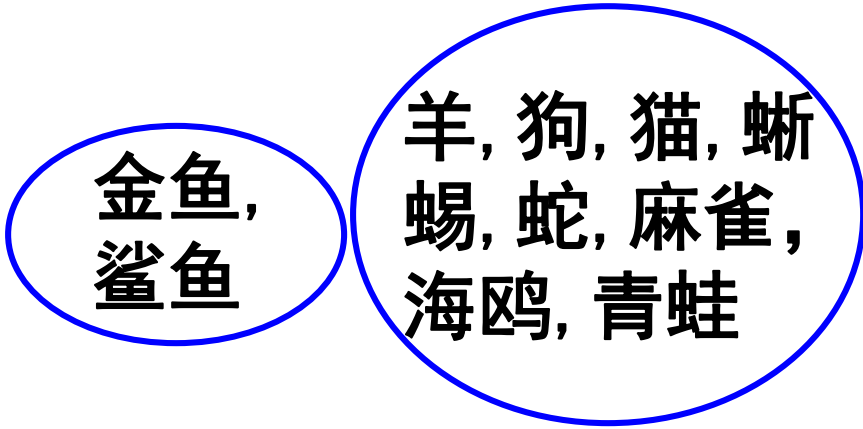

Name of Observation or Cluster
1
5
9
10
2
4
6
7
8
3
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.70
Semi-Partial R-Squared

聚类分析根据一批样本的许多观测指标，按照一定的数学公式具体地计算一些样本或一些指标的相似程度，把相似的样本或指标归为一类，把不相似的归为一类。

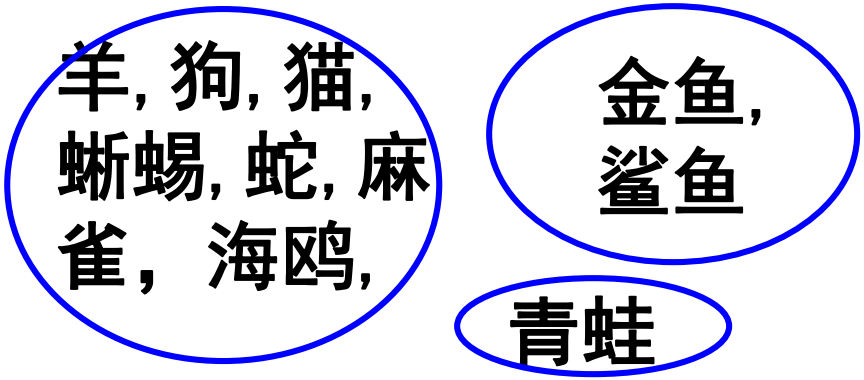
聚类准则对聚类结果的影响



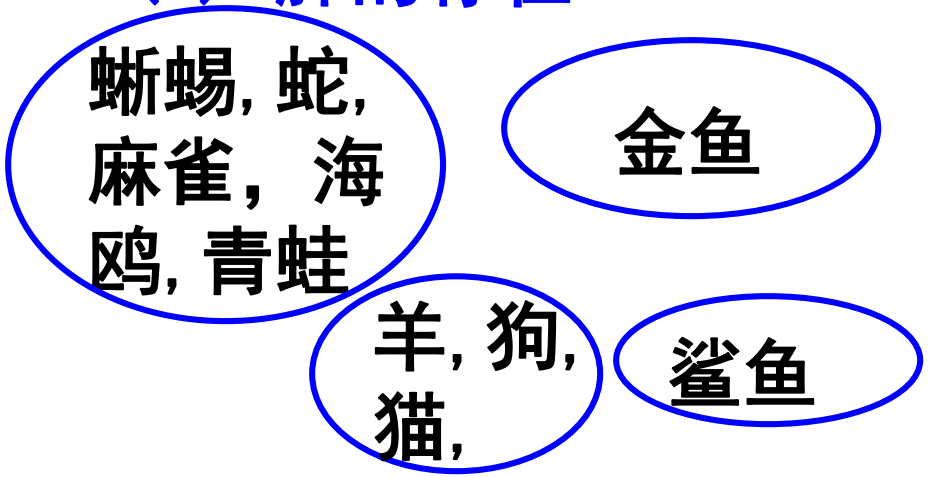
(a) 繁衍后代的方式



(b) 肺的存在

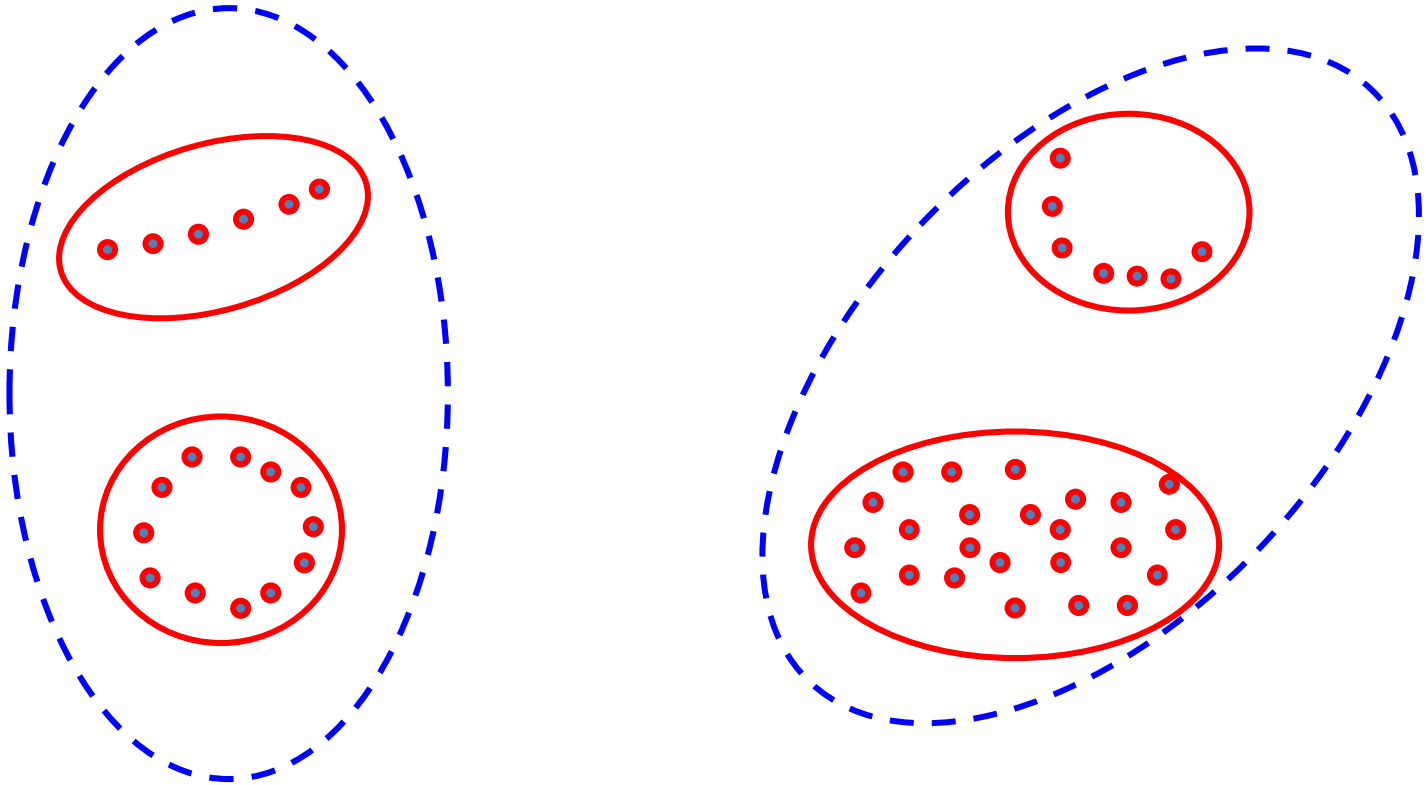


(c) 生存环境



(d) 繁衍后代的方式和是否存在肺

距离测度对聚类结果的影响



数据的粗聚类是两类，细聚类为4类

样本或变量间亲疏程度的测度

- 研究样本或变量的亲疏程度的数量指标有两种：
- 一种叫相似系数，性质越接近的变量或样本，它们的相似系数越接近于1或一1，而彼此无关的变量或样本它们的相似系数则越接近于0，相似的为一类，不相似的为不同类。
- 另一种叫距离，它是将每一个样本看作p维空间的一个点，并用某种度量测量点与点之间的距离，距离较近的归为一类，距离较远的点应属于不同的类。

- 设有n个样本单位，每个样本测得p项指标（变量），原始资料矩阵为：

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

•在聚类之前，要首先分析样品间的相似性，常用距离来测度样品之间的相似程度。每个样品有*p*个指标（变量）从不同方面描述其性质，形成一个*p*维的向量。如果把*n*个样品看成*p*维空间中的*n*个点，则两个样品间相似程度就可用*p*维空间中的两点距离公式来度量。两点距离公式可以从不同角度进行定义。

定比变量的聚类统计量：距离统计量

- 绝对距离
- 欧式距离
- 明考斯基距离
- 兰氏距离
- 马氏距离
- 切氏距离

- **1. 绝对距离（Block距离）**

$$d_{ij}(1)=\sum_{k=1}^{p}\left|x_{ik}-x_{jk}\right|$$

- **2. 欧氏距离(Euclidean distance)**

$$d_{ij}(2)=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^2\right]^{1/2}$$

- **3. 明考斯基距离(Minkowski)**

$$d_{ij}=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^{q}\right]^{1/q}$$

- **4. 兰氏距离**

$$d_{ij}(L)=\sum_{k=1}^{p}\frac{|x_{ik}-x_{jk}|}{x_{ik}+x_{jk}}$$

- **5. 马氏距离**

$$d_{ij}(M)=\left[(x_i-x_j)'S^{-1}(x_i-x_j)\right]^{1/2}$$

S:样本协方差矩阵

- **6. 切比雪夫距离(Chebychev)**

$$d_{ij}(\infty)=\max_{1\le k\le p}|x_{ik}-x_{jk}|$$

自定义距离（Customized）	$d_{ik}(q_1,q_2)=[\sum_{j=1}^{p}\mid X_{ij}-X_{kj}\mid^{q_1}]^{1/q_2}$ 在 SPSS 中由用户指定指数 q_1 和开方次数 q_2（q_1、q_2 可取 1 至 4 之间的不同值）的距离。

距离选择的原则

一般说来，同一批数据采用不同的距离公式，会得到不同的分类结果。产生不同结果的原因，主要是由于不同的距离公式的侧重点和实际意义都有不同。因此我们在进行聚类分析时，应注意距离公式的选择。通常选择距离公式应注意遵循以下的基本原则：

（1）要考虑所选择的距离公式在实际应用中有明确的意义。如欧氏距离就有非常明确的空间距离概念。马氏距离有消除量纲影响的作用。

（2）要综合考虑对样本观测数据的预处理和将要采用的聚类分析方法。如在进行聚类分析之前已经对变量作了标准化处理，则通常就可采用欧氏距离。

（3）要考虑研究对象的特点和计算量的大小。样品间距离公式的选择是一个比较复杂且带有一定主观性的问题，我们应根据研究对象的特点不同做出具体分折。实际中，聚类分析前不妨试探性地多选择几个距离公式分别进行聚类，然后对聚类分析的结果进行对比分析，以确定最合适的距离测度方法。

- 多元数据中的变量表现为向量形式，在几何上可用多维空间中的一个有向线段表示。在对多元数据进行分析时，相对于数据的大小，我们更多地对变量的变化趋势或方向感兴趣。因此，变量间的相似性，我们可以从它们的方向趋同性或“相关性”进行考察，从而得到“夹角余弦法”和“相关系数”等度量方法。

定比变量的聚类统计量：相似系数统计量

- **1. 相关系数**

$$r_{ij}=\frac{\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)^2\sum_{k=1}^{n}(x_{kj}-\bar{x}_j)^2}}$$

性质：相关系数具有坐标系平移、旋转、比例不变性。

- **2. 夹角余弦**

$$C_{ij}=\frac{\sum_{k=1}^{n}x_{ki}x_{kj}}{\left[\left(\sum_{k=1}^{n}x_{ki}^2\right)\left(\sum_{k=1}^{n}x_{kj}^2\right)\right]^{1/2}}$$

(3) 指数相关系数

$$e(\vec{x},\vec{y})=\frac{1}{n}\sum_{i=1}^{n}\exp[-\frac{3}{4}\frac{(x_i-y_i)^2}{\sigma_i^2}]$$

这里假设 $\vec{x}$ 和 $\vec{y}$ 的维数*n*相同、概率分布相同。

σ_i^2 是第 *i* 个分量的方差。

性质：不受量纲变化的影响。

聚类的类型

- 根据聚类对象的不同，分为Q型聚类和R型聚类。
- Q型聚类：样本之间的聚类即Q型聚类分析，则常用距离来测度样本之间的亲疏程度。
- R型聚类：变量之间的聚类即R型聚类分析，常用相似系数来测度变量之间的亲疏程度。

类的定义

类的划分具有人为规定性，这反映在类的定义的选取及参数的选择上。
分类结果的优劣最后只能根据实际来评价。

定义1 设集合S中任意元素x_i与x_j间的距离d_{ij}有

$$d_{ij} \le h$$

其中h为给定的阈值，称S对于阈值h组成一类。

定义2

$$\frac{1}{k-1}\sum_{x_j \in S} d_{ij} \le h$$

其中k为S中元素的个数。（类内平均距离）

类的定义

定义3 设集合S中任意元素x_i与x_j间的距离d_{ij}有

$$\frac{1}{k(k-1)}\sum_{x_i\in S}\sum_{x_j\in S} d_{ij} \le h$$

$$d_{ij} \le r$$

其中k为S中元素的个数，称S对于阈值h, r组成一类。

定义4 $\forall x_i \in S$ ，$\exists x_j \in S$，使$d_{ij} \le h$成立，则称S对于阈值h组成一类。（最近距离）

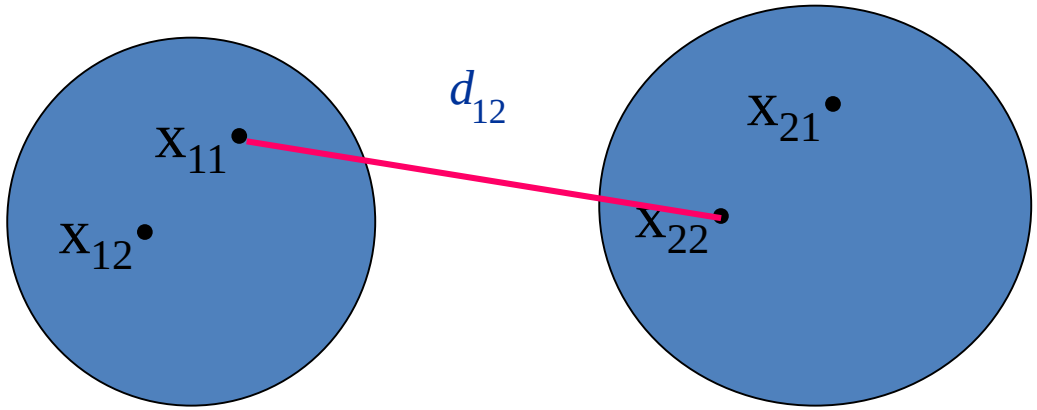
定义5 若将集合S任意分成两类S_1，S_2，这两类间的距离$D(S_1, S_2) \le h$，则称S对于阈值h组成一类。

类间距离的度量方法

- 最短距离法(Nearest Neighbor)
- 最长距离法(Further Neighbor)
- 组间平均连接法(Between-group linkage)
- 组内平均连接法(Within-group linkage)
- 重心法(Centroid clustering)
- 中位数法(Median clustering)
- 离差平方和法(Ward’s method)

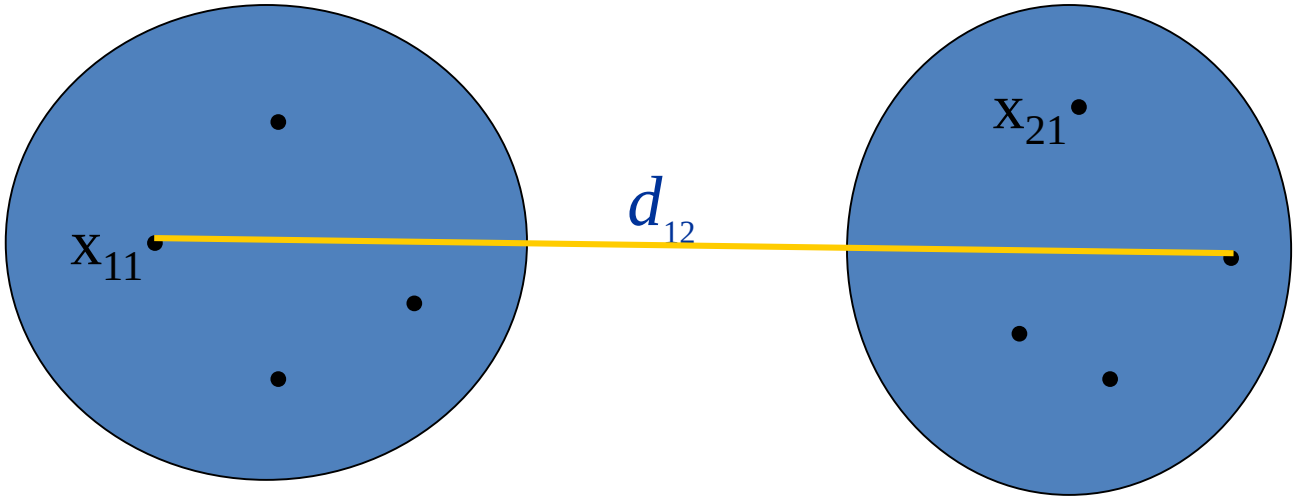
最短距离法(Nearest Neighbor)

- 以两类中距离最近的两个个体之间的距离作为类间距离。


d_{12}
x_{11}
x_{12}
x_{21}
x_{22}

最长距离法(Further Neighbor)

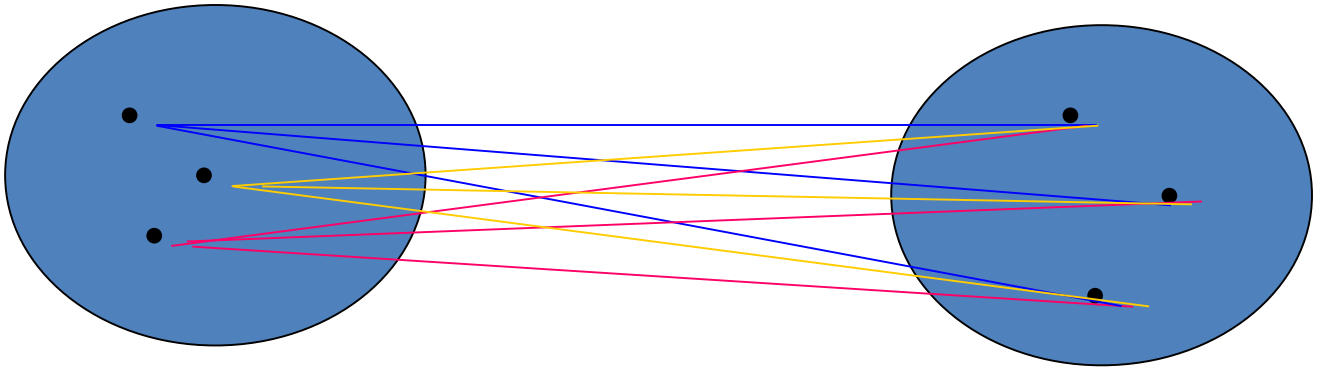
- 以两类中距离最远的两个个体之间的距离作为类间距离。


x11
d12
x21

组间平均连接法 (Between-group linkage)

- 以两类个体两两之间距离的平均数作为类间距离。

组间平均连接法（Between-group Linkage)



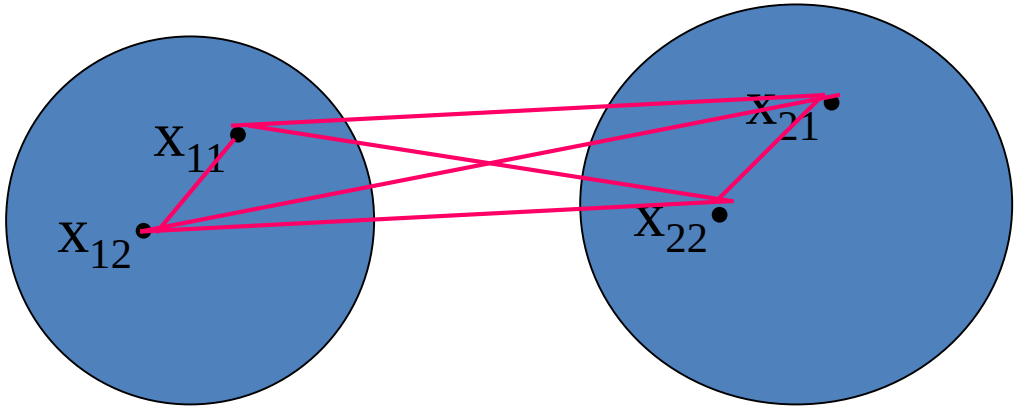
$$\frac{d_1 + d_2 + \cdots + d_9}{9}$$

组内平均连接法 (Within-group linkage)

- 将两类个体合并为一类后，以合并后类中所有个体之间的平均距离作为类间距离。

组内平均连接法（Within-group Linkage)

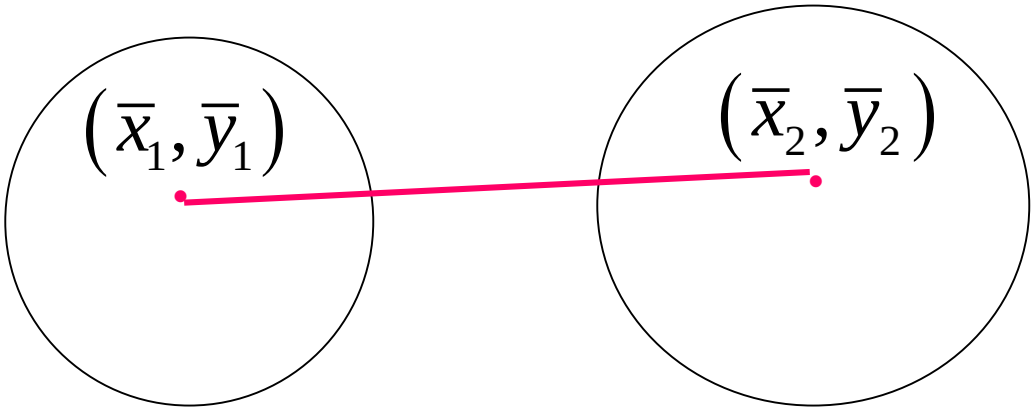
$$\frac{d_1 + d_2 + d_3 + d_4 + d_5 + d_6}{6}$$



重心法(Centroid clustering)

- 以两类变量均值（重心）之间的距离作为类间距离。

重心距离：均值点的距离



中位数法(Median clustering)

- 以两类变量中位数之间的距离作为类间距离。

有一组数据：

$$X_1, \cdots, X_N$$

将它按从小到大的顺序排序为：

$$X_{(1)}, \cdots, X_{(N)}$$

则当N为奇数时，$m_{0.5} = X_{(N+1)/2}$；当N为偶数时，$m_{0.5} = \dfrac{X_{(N/2)} + X_{(N/2+1)}}{2}$。

离差平方和法(Ward's method)

- 离差平方和法是由Ward提出的，因此也称为Ward方法。具体做法是，先将n个个体各自成一类，然后每次减少一类，随着类与类的不断聚合，类内的离差平方和必然不断增大，选择使离差平方和增加最小的两类合并，直到所有的个体归为一类为止。

主要步骤

1. **选择变量**
   （1）和聚类分析的目的密切相关
   （2）反映要分类变量的特征
   （3）在不同研究对象上的值有明显的差异
   （4）变量之间不能高度相关
2. **数据变换处理**
   为了消除各指标量纲的影响，需要对原始数据进行必要的变换处理。

3. 计算聚类统计量

聚类统计量是根据变换以后的数据计算得到的一个新数据，它用于表明各样本或变量间的关系密切程度。常用的统计量有距离和相似系数两大类。

4. 聚类

主要涉及两个问题：

（1）选择聚类的方法

（2）确定形成的类数

5. 聚类结果的解释和证实

对聚类结果进行解释是希望对各个类的特征进行准确的描述，给每类起一个合适的名称。这一步可以借助各种描述性统计量进行分析，通常的做法是计算各类在各聚类变量上的均值，对均值进行比较，还可以解释各类产生的原因。

主要的聚类分析方法

常见1，主要的聚类算法可以划分为如下几类：

（1）划分方法；

（2）层次方法；

（3）基于密度的方法；

（4）基于网格的方法；

（5）基于模型的方法。

常见2

（1） 简单聚类方法

算法运行中模式的类别及类的中心一旦确定将不会改变。

（2） 层次聚类法

算法运行中，两类合并为一类，不断重复进行。也称为谱系聚类法。

（3） 动态聚类法

算法运行中，类心不断地修正，各模式的类别的指定也不断地更改。这类方法有—C均值法、ISODATA法等。

划分方法

◆给定一个n个对象或元组的数据库，划分方法构建数据的k个划分，每个划分表示一个聚簇（类），且 $k \leq n$ 同时满足如下条件：

（1）每个聚类内至少包含一个对象；

（2）每个对象必须属于且只属于一个聚类。

◆**注意：**在模糊划分计算中第二个要求可以放宽。

◆一个好的划分的一般**准则**：

➢在同一个类内的对象间尽可能接近或相似(high intra-class similarity)；

➢不同类中的对象间尽可能远离或不同(low inter-class similarity) 。

划分方法

◆为达到全局最优，基于划分的聚类会要求穷举所有可能的划分，但实际中，绝大多数应用采用了以下两个比较流行的启发式方法：

（1）k-平均（k-means）算法：每个聚类用该聚类中对象的平均值来表示；

（2）k-中心点（k-mediods） 算法：每个聚类用接近聚类中心的一个对象来表示。

1. k-平均（k-means）聚类算法

K-平均聚类算法

◆K-平均（k-means）算法以k为参数，把n个对象分为k个簇，以使簇内对象具有较高的相似度，而簇间的相似度较低。

◆ 相似度的计算根据一个簇中对象的**平均值**（被看作簇的重心）来进行。

K-平均聚类算法

(1) *k-means* 算法

算法 6.1: 根据聚类中的均值进行聚类划分的 *k-means* 算法。

输入: 聚类个数 k，以及包含 n 个数据对象的数据库。

输出: 满足方差最小标准的 k 个聚类。

处理流程:

(1) 从 n 个数据对象任意选择 k 个对象作为初始聚类中心;

(2) 循环 (3) 到 (4) 直到每个聚类不再发生变化为止

(3) 根据每个聚类对象的均值 (中心对象), 计算每个对象与这些中心对象的距离; 并根据最小距离重新对相应对象进行划分;

(4) 重新计算每个 (有变化) 聚类的均值 (中心对象)

K-平均聚类算法

算法的基本思想：

◆ 首先，随机的选择k个对象，每个对象初始的代表了一个簇的平均值；

◆对剩余的每个对象，根据其与各个簇中心的距离，将它赋给最近的簇；

◆然后重新计算每个簇的平均值。

◆ 这个过程不断重复，直到准则函数收敛。

K-平均聚类算法

通常选择均方差作为收敛准则函数：

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

其中 E 为数据库中所有对象的均方差之和； p 为代表对象的空间中的一个点； m_i 为聚类 C_i 的均值（ p 和 m_i 均是多维的）。

这个准则试图使得生成的结果尽可能地紧凑和独立：当结果簇是密集的，且簇与簇之间区别明显时，算法的效果较好。

K-平均聚类算法

算法的**特点**：

- 只适用于聚类**均值有意义**的场合，在某些应用中，如：数据集中包含符号属性时，直接应用k-means算法就有问题；
- 用户必须事先指定**k的个数**；
- 对**噪声和孤立点数据敏感**，少量的该类数据能够对聚类**均值起到很大的影响**。

示例

示例 6.2：假设空间数据对象分布如图-6.2（a）所示，设$k=3$，也就是需要将数据集划分为三份（聚类）。

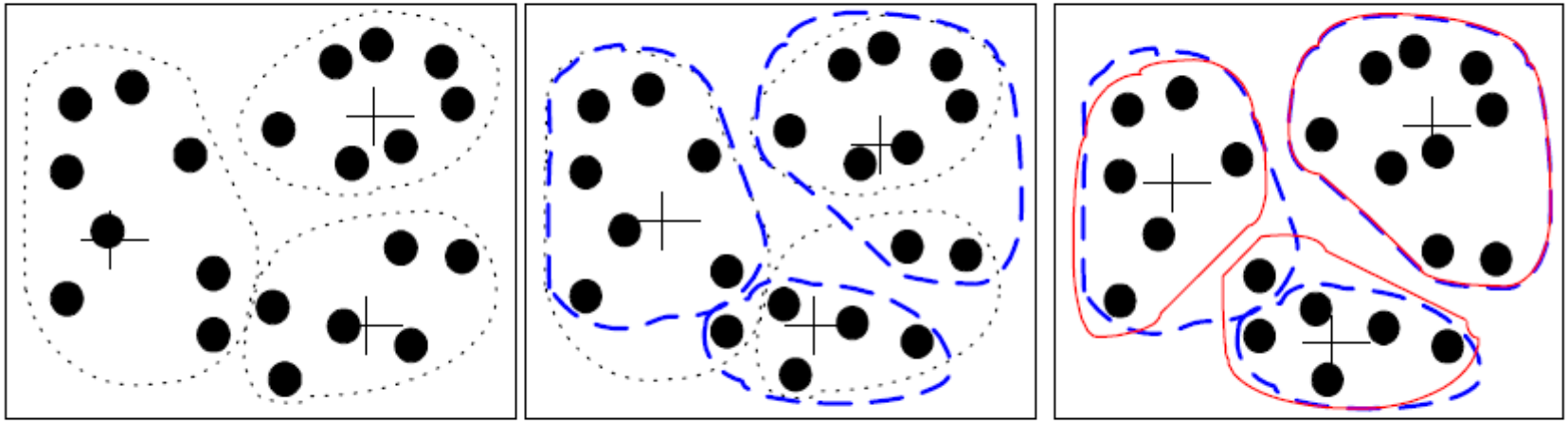
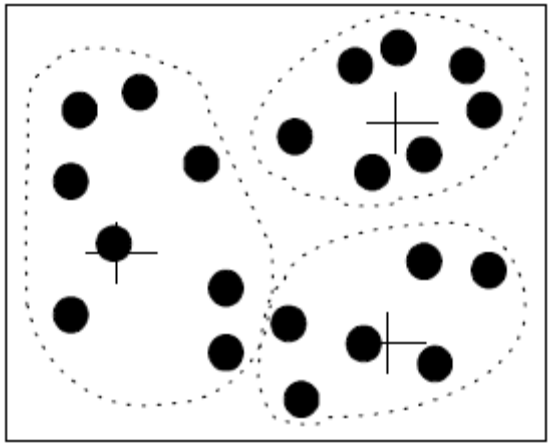


图-6.2 *k-means* 算法聚类过程示意描述

示例

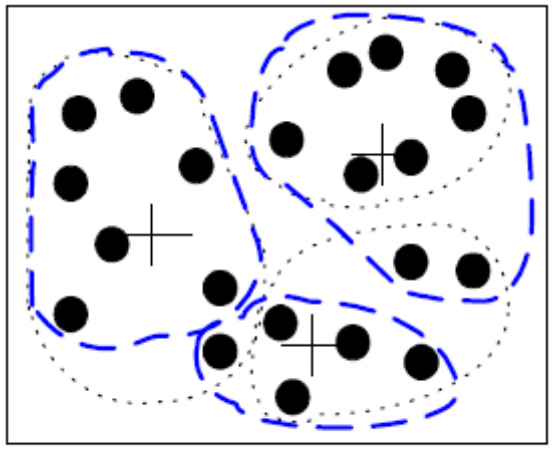
根据算法 6.1，从数据集中任意选择三个对象作为初始聚类中心（图-6.2（a）中这些对象被标上了“+”）；其余对象则根据与这三个聚类中心（对象）的距离，根据最近距离原则，逐个分别聚类到这三个聚类中心所代表的（三个）聚类中；由此获得了如图-6.2（a）所示的三个聚类（以虚线圈出）。



(a)

示例

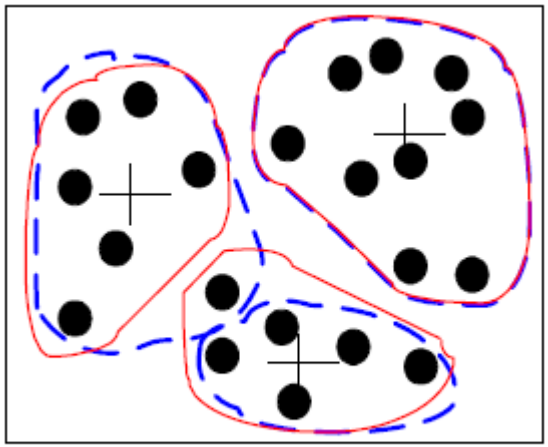
在完成第一轮聚类之后，各聚类中心发生了变化；继而更新三个聚类的聚类中心（图-6.2（b）中这些对象被标上了“+”）；也就是分别根据各聚类中的对象计算相应聚类的（对象）均值。根据所获得的三个新聚类中心，以及各对象与这三个聚类中心的距离，（根据最近距离原则）对所有对象进行重新归类。有关变化情况如图-6.2（b）所示（已用粗虚线圈出）。



(b)

示例

再次重复上述过程就可获得如图-6.2(c)所示的聚类结果(已用实线圈出)。,这时由于各聚类中的对象(归属)已不再变化，整个聚类操作结束。 ■



(c)

2. k-中心点（k-mediods）聚类算法

K-中心点聚类算法

◆K-平均（k-means）算法对于孤立点是敏感的，如何消除？

◆ **思路**：不采用簇中对象的平均值作为参照点，而选用簇中位置最中心的对象，即**中心点（mediod），**仍然基于**最小化所有对象与其参照点之间的相异度之和的原则来进行**。

◆ 这就是**k-中心点**（k-mediods）的算法基础。

K-中心点聚类算法

基本策略：

◆ 首先为每个簇随意选择一个代表对象，称为中心点，剩余的对象根据其与中心点间的距离分配给最近的一个簇。

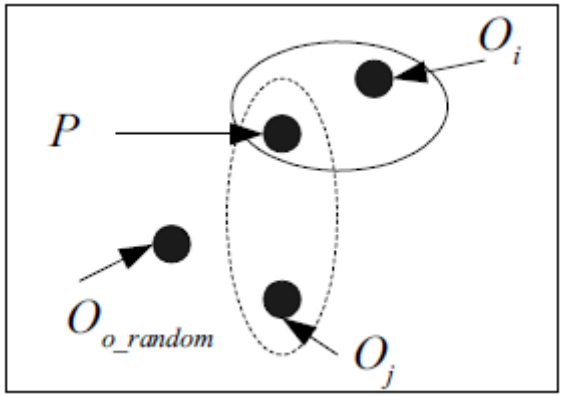
◆ 然后重复地用非中心点对象来替代中心对象，如果它改善了结果聚类的整体距离，则进行替代。

◆ 聚类结果的质量用一个代价函数来估算，该函数度量对象与其参照对象之间的平均相异度。

K-中心点聚类算法

为判定一个非代表对象 o_{random} 是否是当前代表对象O_j的一个好的替代，对于每一个非中心点对象p，考虑如下四种情况：

（1） 若对象 p 当前属于 o_j（所代表的聚类），且如果用 o_{random} 替换 o_j 作为新聚类代表，而 p 就更接近其它 o_i（$i \neq j$），那么就将 p 归类到 o_i（所代表的聚类）中；



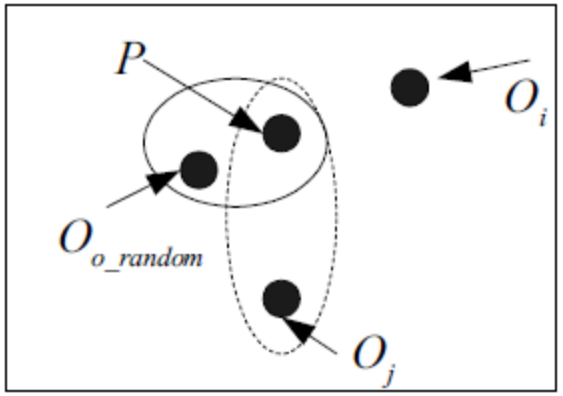
重新分配给O_i

代价函数：Cpjo=d(i,p)-d(j,p)

K-中心点聚类算法

为判定一个非代表对象 o_{random} 是否是当前代表对象Oj的一个好的替代，对于每一个非中心点对象p，考虑如下四种情况：

（2） 若对象 p 当前属于 o_j（所代表的聚类），且如果用 o_{random} 替换 o_j 作为新聚类代表，而 p 最更接近 o_{random}，那么就将 p 归类到 o_{random}（所代表的聚类）中；



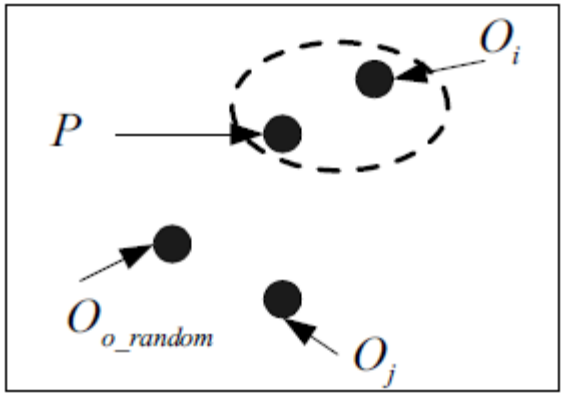
重新分配给O_{random}

代价函数：Cpjo=d(o,p)-d(j,p)

K-中心点聚类算法

为判定一个非代表对象 o_{random} 是否是当前代表对象Oj的一个好的替代，对于每一个非中心点对象p，考虑如下四种情况：

（3） 若对象 p 当前属于 o_i（所代表的聚类）（$i \neq j$），且如果用 o_{random} 替换 o_j 作为新聚类代表，而 p 仍然最接近 o_i，那么 p 归类不发生变化；



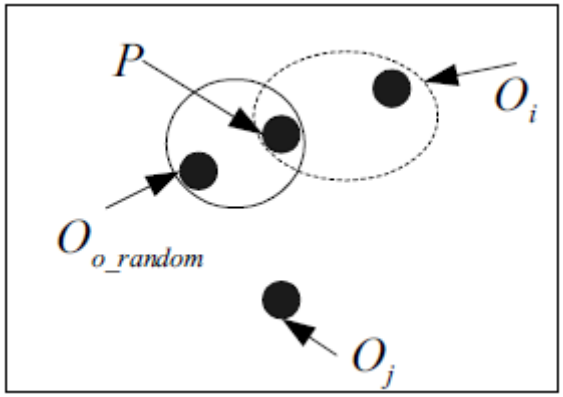
不发生变化

代价函数：Cpjo=0

K-中心点聚类算法

为判定一个非代表对象 O_{random} 是否是当前代表对象Oj的一个好的替代，对于每一个非中心点对象p，考虑如下四种情况：

（4） 若对象 p 当前属于 o_i（所代表的聚类）（$i \neq j$），且如果用 o_{random} 替换 o_j 作为新聚类代表，而 p 最更接近 o_{random}，那么就将 p 归类到 o_{random}（所代表的聚类）中；



重新分配给O_{random}

代价函数：Cpjo=d(o,p)-d(p,i)

K-中心点聚类算法

每当重新分配发生时，替换的总代价是所有非中心对象产生的代价之和：

$$TC_{jo} = \sum_{j=1}^{n} C_{pjo}$$

- 如果总代价是负的，则O_j可被O_{random}代替；
- 否则，则认为当前的中心点O_j是可接受的，在本次迭代中没有变化。

K-中心点聚类算法

算法 6.2：根据聚类的中心对象（聚类代表）进行聚类划分的 *k-medoids* 算法。

输入：聚类个数 k，以及包含 n 个数据对象的数据库。

输出：满足基于各聚类中心对象的方差最小标准的 k 个聚类。

处理流程：

（1） 从 n 个数据对象任意选择 k 个对象作为初始聚类（中心）代表；

（2） 循环（3）到（5）直到每个聚类不再发生变化为止

（3） 依据每个聚类的中心代表对象，以及各对象与这些中心对象间距离；并根据最小距离重新对相应对象进行划分；

（4） 任意选择一个非中心对象 o_{random}；计算其与中心对象 o_j 交换的整个成本 S。

（5） 若 S 为负值则交换 o_{random} 与 o_j 以构成新聚类的 k 个中心对象

两种划分方法的关系

关系：

- k-中心点方法比k-均值方法更健壮，因为其不易受到极端数据的影响；
- 但k-中心点方法比k-均值方法的执行代价高；
- 两种方法都需要用户提前指定聚类结果的数目k。

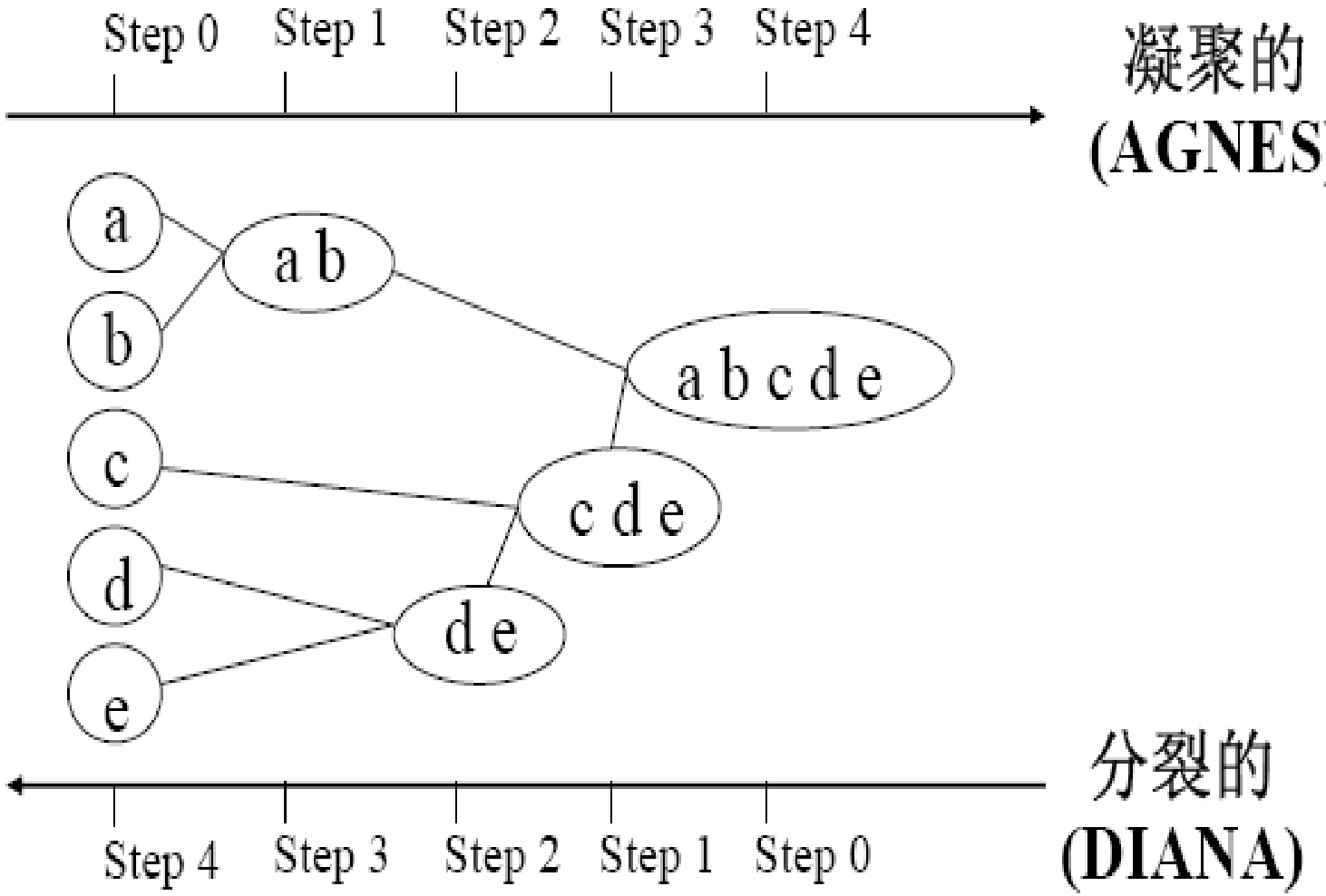
层次方法

层次方法：

该方法对给定的数据对象集合进行层次分解，根据层次分解的方式，层次的方法被分为凝聚的和分裂的：

◆**凝聚层次方法**：也称自底向上方法，一开始将每个对象作为单独的一组，然后相继地合并相近的对象或组，直到所有的组合并为一个，或达到某个终止条件，代表：AGNES算法；

◆ **分裂层次方法**：也称自顶向下方法，一开始所有对象置于一个簇中，在迭代的每一步，一个簇被分裂为更小的簇，直到最终每个对象单独为一个簇，或达到某个终止条件，代表：DIANA算法。


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

距离计算方法

四个常用的计算聚类间距离的公式说明如下：

- **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：** $d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$
- **距离均值：** $d_{mean}(C_i, C_j) = |m_i - m_j|$
- **平均距离：** $d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|$

其中 m_i 为聚类 C_i 的均值；n_i 为 C_i 中的对象数；$|p - p'|$ 为两个数据对象或点 p 和 p' 之间的距离。

AGNES算法

- AGNES 算法：最初将每个对象作为一个簇，然后这些簇根据某些准则被一步步地合并，直到达到初始指定的簇数目。

算法9-1 AGNES（自底向上凝聚算法）

输入：包含n个对象的数据库，终止条件簇的数目k。

输出：k个簇，达到终止条件规定簇数目。

(1) 将每个对象当成一个初始簇；

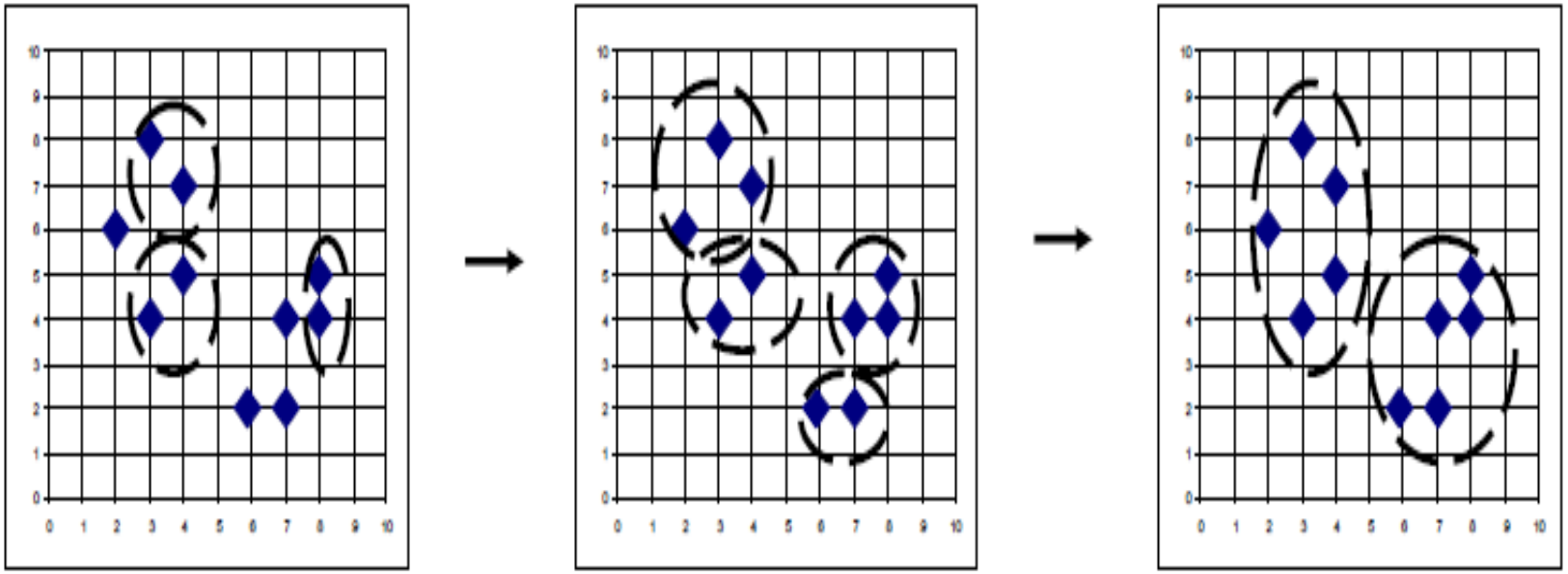
(2) REPEAT

(3) 根据两个簇中最近的数据点找到最近的两个簇；

(4) 合并两个簇，生成新的簇的集合；

(5) UNTIL 达到定义的簇的数目；

AGNES算法



AGNES算法示意图

DIANA算法

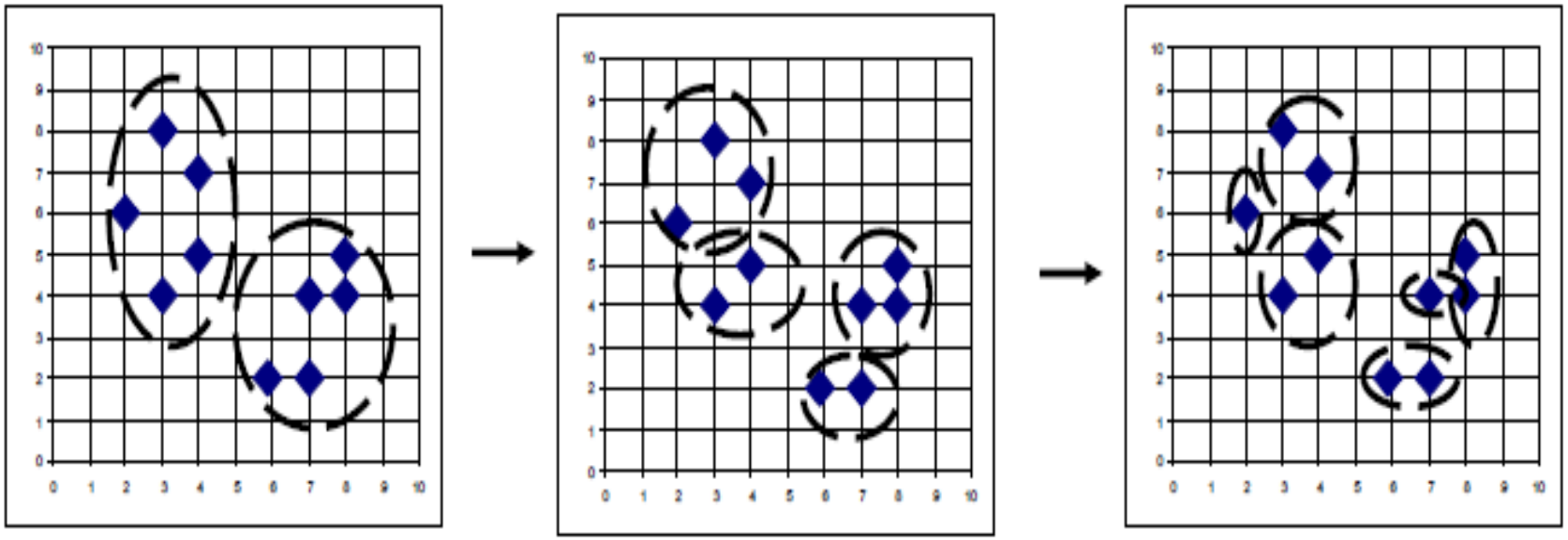
- DIANA 算法：与AGNES算法相反，初始所有节点都在一个大簇中，根据某些准则被一步步地分解，直到达到初始设定的簇数目。
- 聚类过程中，DIANA算法将用到如下两种测度方法：
  - **簇的直径：一个簇中的任意两个数据点的距离中的最大值；**
  - **平均相异度（平均距离）：**

$$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x \in C_i} \sum_{y \in C_j} |x - y|$$

DIANA算法

算法9-2 DIANA（自顶向下分裂算法）
输入：包含n个对象的数据库，终止条件簇的数目k。
输出：k个簇，达到终止条件规定簇数目。
（1）将所有对象整个当成一个初始簇；
（2） FOR （i=1; i≠k; i++) DO BEGIN
（3） 在所有簇中挑出具有最大直径的簇C；
（4） 找出C中与其它点平均相异度最大的一个点p并把p放入splinter group，剩余的放在old party中；
（5）. REPEAT
（6） 在old party里找出到最近的splinter group中的点的距离不大于到old party中最近点的距离的点，并将该点加入splinter group。
（7） UNTIL 没有新的old party的点被分配给splinter group；
（8） splinter group和old party为被选中的簇分裂成的两个簇，与其它簇一起组成新的簇集合。
（9） END.

DIANA算法



DIANA算法示意图

基于密度的聚类方法

密度方法：

- 绝大多数聚类方法基于对象之间的距离进行聚类，这样的方法只能发现球状的簇，而在发现任意形状的簇上遇到了困难。
- **基于密度的方法**：只要一个区域中点的密度（对象或数据点的数目）超过某个阈值，就将其加到与之相近的聚类中去。
- **这种方法可以过滤噪声孤立点数据，发现任意形状的簇。**
- 代表算法有：DBSCAN、OPTICS、DENCLUE算法等。

基于密度的方法：DBSCAN

DBSCAN（Density-based Spatial Clustering of Application with Noise）是一个基于密度的聚类算法。该算法将具有足够高密度的区域划分为簇，并可以在带有噪声的空间数据中发现任意形状的聚类。

在该方法中，簇被定义为密度相连的点的最大集合。

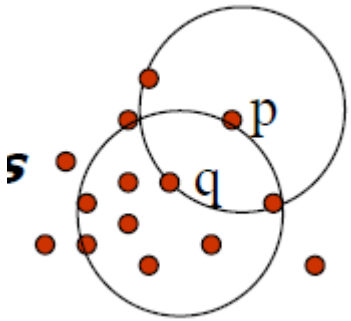
先介绍该方法中涉及到的一些基本的定义。

基于密度的方法：DBSCAN

定义 1： 对象的ε-临域：给定对象在半径ε内的区域。

定义2： 核心对象：如果一个对象的ε-临域至少包含最小数目MinPts个对象，则称该对象为核心对象。

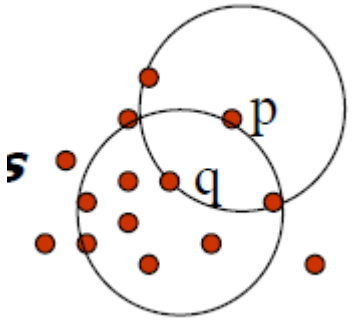
例如，在下图中，设定ε=1cm，MinPts=5，则q是一个核心对象。



基于密度的方法：DBSCAN

定义 3： 直接密度可达：给定一个对象集合D，如果p是在q的ε-邻域内，而q是一个核心对象，我们说对象p从对象q出发是直接密度可达的。

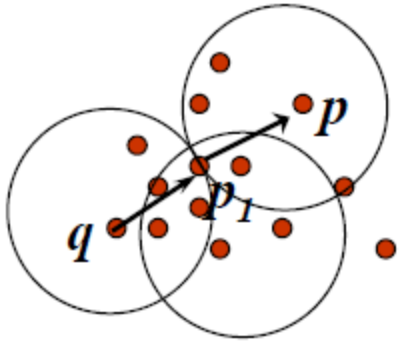
例如，在下图中，设定ε=1cm，MinPts=5， q是一个核心对象，对象p从对象q出发是直接密度可达的。



基于密度的方法：DBSCAN

定义 4： 密度可达的：如果存在一个对象链p_1，p_2，…，p_n，$p_1=q$，$p_n=p$，对$p_i \in D$，（$1<=i<=n$），p_{i+1}是从p_i关于ε和MitPts直接密度可达的，则对象p是从对象q关于ε和MinPts密度可达的。

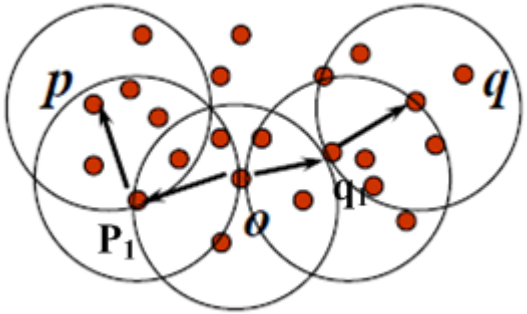
例如，在下图中，ε=1cm，MinPts=5，q是一个核心对象，p_1是从q关于ε和MitPts直接密度可达，p是从p_1关于ε和MitPts直接密度可达，则对象p从对象q关于ε和MinPts密度可达的。



基于密度的方法：DBSCAN

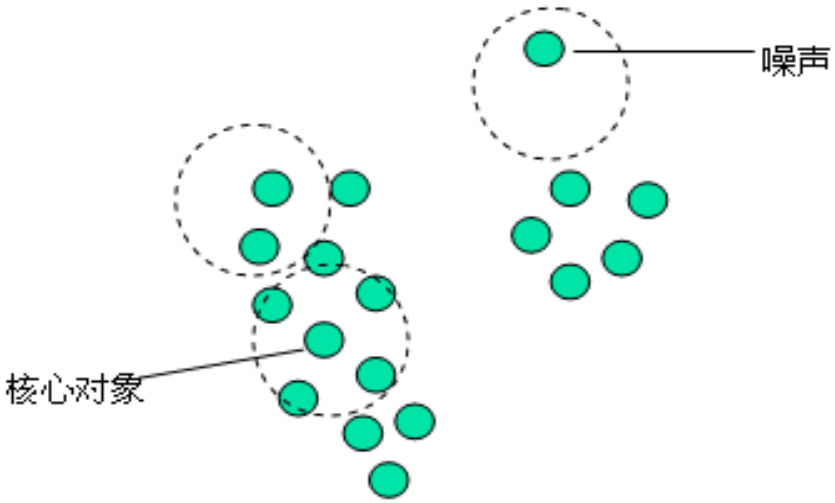
定义 5： 密度相连的：如果对象集合D中存在一个对象o，使得对象p和q是从o关于ε和MinPts密度可达的，那么对象p和q是关于ε和MinPts密度相连的。

例如，在下图中，ε=1cm，MinPts=5，o是一个核心对象，p_1是从o关于ε和MitPts直接密度可达，p是从p_1关于ε和MitPts直接密度可达，则对象p从对象q关于ε和MinPts密度可达的；同理，q也是从o关于ε和MinPts密度可达的，则，称对象p和q是关于ε和MinPts密度相连的。



基于密度的方法：DBSCAN

定义 6： 噪声：一个基于密度的簇是基于密度可达性的最大的密度相连对象的集合。不包含在任何簇中的对象被认为是“噪声”。



DBSCAN算法描述

- DBSCAN通过检查数据集中每个对象的ε-邻域来寻找聚类。
- 如果一个点p的ε-邻域包含多于MinPts个对象，则创建一个p作为核心对象的新簇。
- 然后，DBSCAN反复地寻找从这些核心对象直接密度可达的对象，这个过程可能涉及一些密度可达簇的合并。
- 当没有新的点可以被添加到任何簇时，该过程结束。

DBSCAN算法描述

DBSCAN算法描述

算法9-3 DBSCAN

输入：包含*n*个对象的数据库，半径ε，最少数目MinPts。

输出：所有生成的簇，达到密度要求。

1. REPEAT

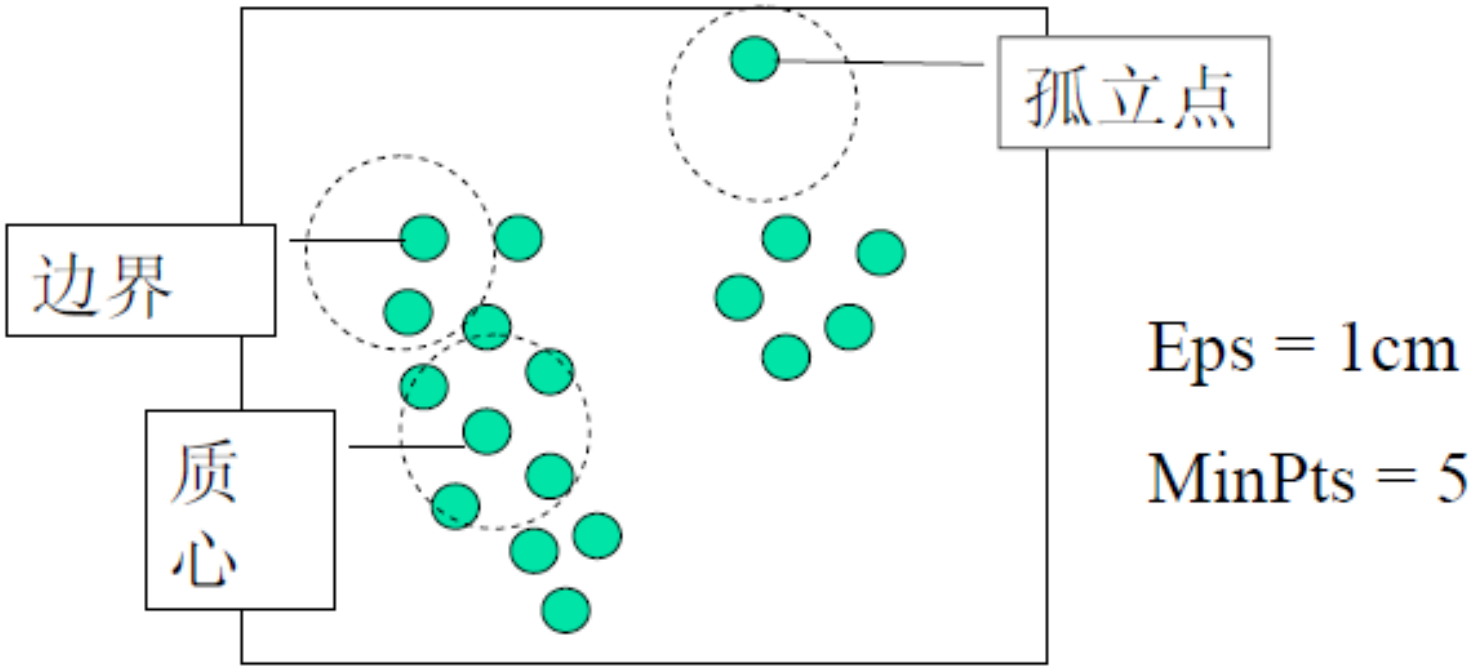
2. 从数据库中抽取一个未处理过的点；

3. IF 抽出的点是核心点 THEN找出所有从该点密度可达的对象，形成一个簇

4. ELSE 抽出的点是边缘点(非核心对象)，跳出本次循环，寻找下一点；

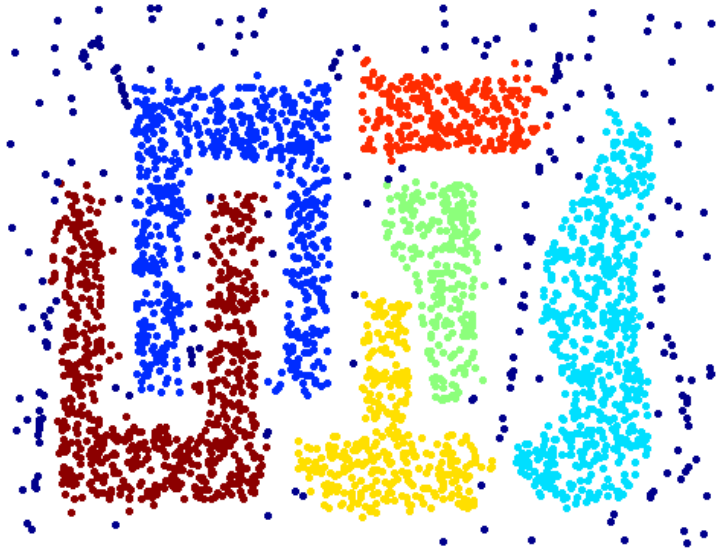
5. UNTIL 所有点都被处理；

DBSCAN算法描述



DBSCAN

Original Points



Clusters

特点：

- 抗噪声
- 能处理任意形状聚类

基于网格的聚类

- 基本思想是将每个属性的可能值分割成许多相邻的区间，创建网格单元的集合（对于的讨论我们假设属性值是序数的、区间的或者连续的）。
- 每个对象落入一个网格单元，网格单元对应的属性区间包含该对象的值。

- 优点是它的处理速度很快，其处理时间独立于数据对象的数目，只与量化空间中每一维的单元数目有关。

STING:统计信息网格

- STING是一种基于网格的多分辨率聚类技术，它将空间区域划分为矩形单元。
  - 针对不同级别的分辨率，通常存在多个级别的矩形单元，
  - 这些单元形成了一个层次结构：高层的每个单元被划分为多个低一层的单元。
  - 关于每个网格单元属性的统计信息（例如平均值、最大值和最小值）被预先计算和存储。这些统计信息用于回答查询。

STING:统计信息网格

网格中常用参数

- **count**−网格中对象数目
- **mean**−网格中所有值的平均值
- **stdev**−网格中属性值的标准偏差
- **min**−网格中属性值的最小值
- **max**−网格中属性值的最大值
- **distribution**−网格中属性值符合的分布类型。如正态分布、均匀分布、指数分布或者none（分布类型未知）

STING:统计信息网格

STING聚类的层次结构

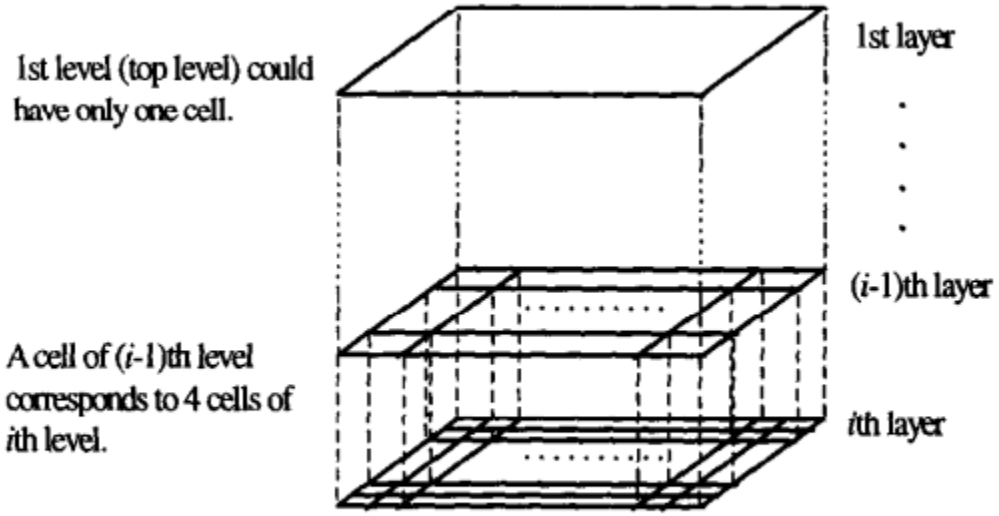
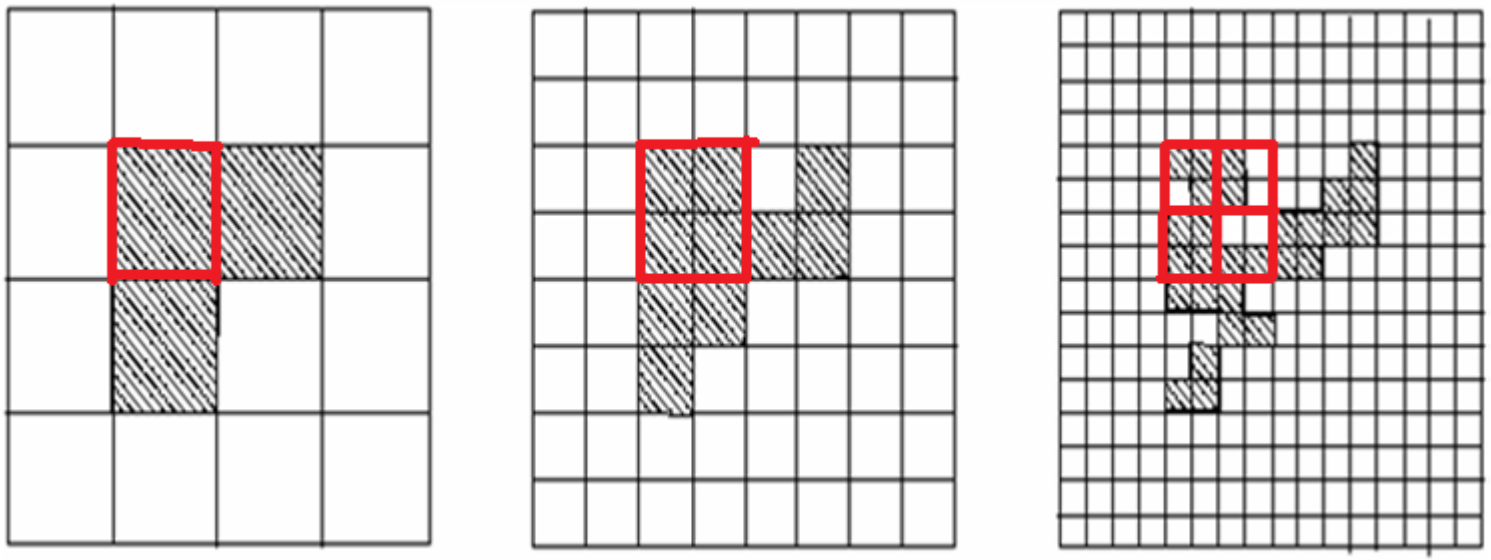


Figure 1: Hierarchical Structure

STING:统计信息网格



level i　　level i+1　　level i+2

a cell of (i-1)th level corresponds to 4 cells of (i)th level

动态聚类方法:ISODATA 简介

ISODATA算法是在k-均值算法的基础上，增加对聚类结果的“合并”和“分裂”两个操作，并设定算法运行控制参数的一种聚类算法。

“合并”操作：
当聚类结果某一类中样本数太少，或两个类间的距离太近时，进行合并。

“分裂”操作：
当聚类结果某一类中样本某个特征类内方差太大或者样本个数太多，将该类进行分裂

算法特点

1. 使用方差平方和作为基本聚类准则
2. 设定指标参数来决定是否进行“合并”或“分裂”
3. 设定算法控制参数来决定算法的运算次数
4. 具有自动调节最优类别数k的能力

参数

k_{init}：初始聚类中心数(NUMCLUS)

n_{min}:一个类中的最小数据个数(SAMPRM)

I_{max}:最大迭代次数(MAXITER)

σ_{max}:一个类别中的所有点的最大标准差（每个维度分别计算）(STDV)

L_{min}:两个聚类中心的最小距离

P_{max}:每次迭代过程中最多可以进行的“合并”操作次数(MAXPAIR)

步骤

$S = (x_1, \cdots\cdots, x_n)$表示n个待分类的点，每一个点都是一个d维向量，$x_j=(x_{j1}, \cdots\cdots, x_{jd})$，$||x||$表示x向量的欧几里德长度，k表示当前聚类中心个数

1、令$k=k_{init}$，从S中随机选取k个聚类中心$z=\{z_1, \cdots\cdots, z_k\}$

2、按照最近邻原则将样本集中的每一个点分类到某一个聚类中心中，某一个聚类中心中的所有点用S_j表示

3、根据n_{min}判断合并，如果类S_j中的样本个数小于n_{min}，则删除该聚类中心z_j，此时样本聚类中心个数k相应减少，此时转到步骤2

4、计算分类后的参数：每个类的质心、类内的平均距离($\triangle_j$)、总体平均距离（$\triangle$）

5、判断此时是否达到最大迭代次数，若达到则停止迭代，若未达到则进行聚类中心合并、聚类中心分裂？

6、分裂操作 ，若有分裂操作，则分裂完后返回第二步

7、合并操作，合并后重新标定对应的类别（如何分裂？如何合并？）

8、判断迭代停止——是否达到最大迭代次数？是否收敛？

4、计算分类后的参数：每个类的质心、类内的平均距离(△j)、总体平均距离（△）

类内的平均距离(△j)：

$$\Delta_j \leftarrow \frac{1}{n_j} \sum_{\mathbf{x} \in S_j} \|\mathbf{x} - \mathbf{z}_j\|, \quad \text{for } 1 \leq j \leq k.$$

总体平均距离（△）：

$$\Delta \leftarrow \frac{1}{n} \sum_{j=1}^{k} n_j \Delta_j .$$

5、判断此时迭代停止、聚类中心合并、聚类中心分裂？

◆ 迭代停止条件：
达到最大迭代次数

◆聚类中心分裂条件：
1、$k <= k_{ini}/2$，即聚类中心的数目不到初始聚类中心的数目的一半
2、$k_{ini}/2 < k < 2k_{ini}$，且迭代次数为奇数次
（满足此条件则转到第六步）

◆聚类中心合并的条件：
1、$k > 2k_{ini}$，即聚类中心的数目大于初始聚类中心得数目的两倍
2、 $k_{ini}/2 < k < 2k_{ini}$，且迭代次数为偶数次
（满足此条件则转到第七步）

6、分裂操作

◆ 分裂条件

对于每一个聚类样本S_j,计算其中样本点到类中心的标准差向量$V_j=(V_1,\cdots\cdots,Vd)$

$$v_{ji} \leftarrow \left(\frac{1}{n_j}\sum_{\mathbf{x}\in S_j}(x_i - z_{ji})^2\right)^{1/2} \quad \text{for } 1 \le j \le k \text{ and } 1 \le i \le d.$$

其中$V_{j,max}$表示Vj中某一维的最大值

分类操作条件，对于任意Sj

$V_{j,max}$ > σmax,标准差向量中对应的某一维的最大值大于σmax

◆分裂操作

根据$V_{j,max}$的方向对该聚类中心进行分裂

7、合并操作

计算各类中心间的距离d_{ij},并将这些距离按照增序排列，将符合规定的S_i和S_j进行合并，将其中的样本点归为一类，聚类中心为合并后的类别的质心，一次迭代过程中合并的类别数不能超过Pmax

符合合并规则的类别应满足条件：

1、$d_{ij} <$ Lmin

2、Si或者Sj在此次迭代过程中没有被合并过，即每一个类别只能合并一次

模糊C均值聚类算法（FCM）

FCM算法是一种基于划分的聚类算法，它的思想就是使得被划分到同一簇的对象之间相似度最大，而不同簇之间的相似度最小。

模糊C均值算法是普通C均值算法的改进，普通C均值算法对于数据的划分是硬性的，而FCM则是一种柔性的模糊划分。在介绍FCM具体算法之前我们先介绍一些模糊集合的基本知识。

隶属度函数是表示一个对象x隶属于集合A的程度的函数，通常记做 $\mu_A(x)$，其自变量范围是所有可能属于集合A的对象（即集合A所在空间中的所有点），取值范围是[0,1]，即0<=1， $\mu_A(x)$<=1。$\mu_A(x)$=1表示x完全隶属于集合A，相当于传统集合概念上的x∈A。一个定义在空间X={x}上的隶属度函数就定义了一个模糊集合A，或者叫定义在论域X={x}上的模糊子集 。对于有限个对象x_1，x_2，……，x_n模糊集合 可以表示为：

$$\underset{\sim}{A}=\{(\mu_A(x_i),x_i)\,|\,x_i\in X\}$$

1.FCM

设聚类中心V={$v_i|i=1,2,\ldots,c$},隶属度矩阵U={$u_{ik}|i=1,2,\ldots,c;k=1,2,\ldots,n$}。

FCM聚类算法是一种模糊目标函数法，其目标函数J(U,V)定义为

$$\begin{cases} J(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m(d_{ik})^2 \\ s.t. \quad \sum_{i=1}^{c}u_{ik}=1, \quad u_{ij}\in[0,1] \end{cases}$$

其中u_{ik}是第k个样本属于第i类的隶属度，v_i为第i类的聚类中心，m的最佳范围是[1.5,2.5],是第 k个样本到第i类的欧式距离，定义为：

$$d_{ik}^2=\|x_k-v_i\|^2$$

1.FCM

聚类准则取J(U,V)极小值

$$\min\{J(U,V)\}$$

由于隶属度矩阵U的各类都是独立的，所以

$$\min\{J(U,V)\}=\min\left\{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m(d_{ik})^2\right\}$$

$$=\sum_{k=1}^{n}\min\left\{\sum_{i=1}^{c}(u_{ik})^m(d_{ik})^2\right\}$$

上述极值的约束条件为$\sum_{i=1}^{c}u_{ik}=1$, 利用拉格朗日乘数法来求解：

1.FCM

$$\begin{cases} F=\sum_{i=1}^{c}(u_{ik})^{m}(d_{ik})^{2}+\lambda(\sum_{i=1}^{c}u_{ik}-1) \\ \dfrac{\partial F}{\partial \lambda}=0 \\ \dfrac{\partial F}{\partial u_{ij}}=0 \end{cases}$$

可求得隶属度矩阵和聚类中心：

$$u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{2/(m-1)}}$$

$$v_{i}=\frac{\sum_{k=1}^{n}(u_{ik})^{m}x_{k}}{\sum_{k=1}^{n}(u_{ik})^{m}}$$

2.FCM算法步骤描述

综上所述，FCM算法步骤描述如下：

步骤 1: 设定聚类数目c，加权指数m，以及迭代终止阈值ε；
步骤 2: 初始化隶属度矩阵$U^{(0)}$
步骤 3: 设置迭代计数器b=0;
步骤 4: 按下面公式计算$v_i^{(b)}$和$U^{(b)}$：

$$v_i^{(b)} = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m}$$

$$u_{ij}^{(b)} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{2/(m-1)}}$$

步骤 5: 若 $\|U^{(b)} - U^{(b+1)}\| < \varepsilon$， 则终止迭代，获得最佳的模糊隶属度矩阵和对应的聚类中心矩阵，根据最大隶属度原则分割图像；否则令**b=b+1,** 返回步骤**4**，继续迭代运算**.**

3.图像分割

1 图像

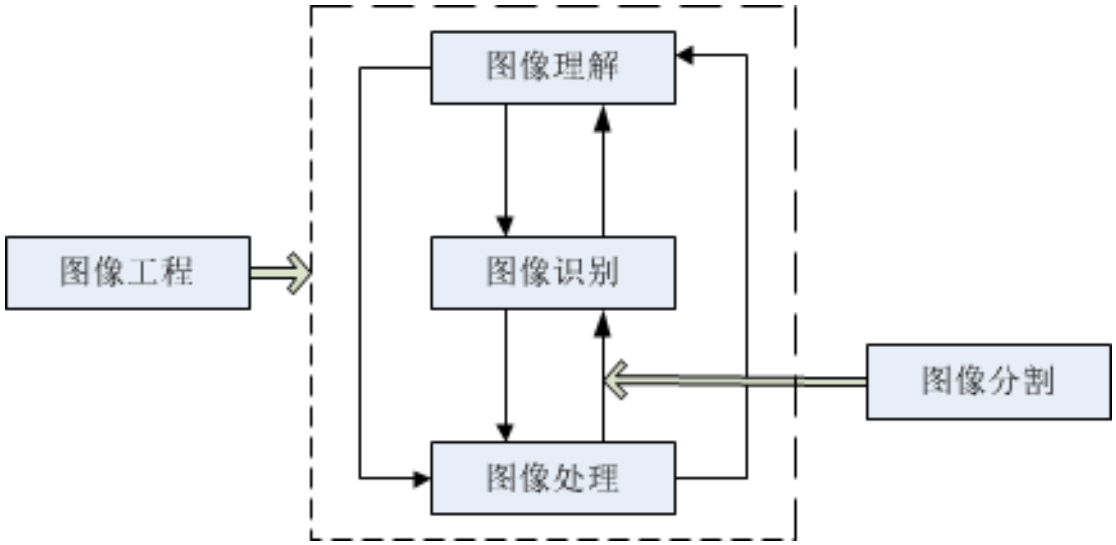
- 人类通过各种各样的信息来认识世界、了解世界并改造这个世界。这些信息种类包括文字、声音和图像等。而人类获取的这些信息中80%是来自视觉的图像信息

- 图像是人类最重要和最有效的信息获取和交流方式。所以对信息的分类实质上就是对图像信息的分类

- 图像信息的分类实质上就是对图像进行分割

2 图像分割

- 图像分割的目标是重点根据图像中的物体将图像的像素分类，并提取感兴趣目标
- 图像分割是图像识别和图像理解的基本前提步骤

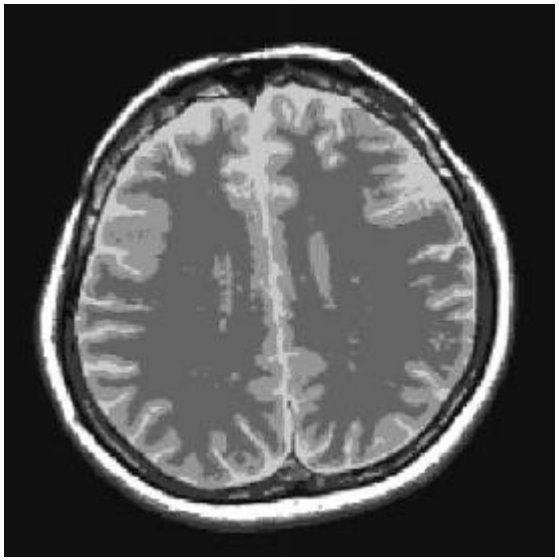


图像分割在图像工程中的地位

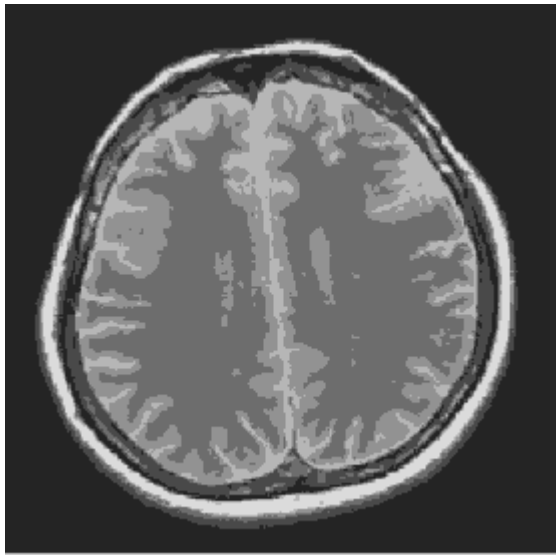
4.FCM在图像分割中应用

4.1 医学图像

MRI脑图，通过分割区分灰质、白质、骨骼



图像分割



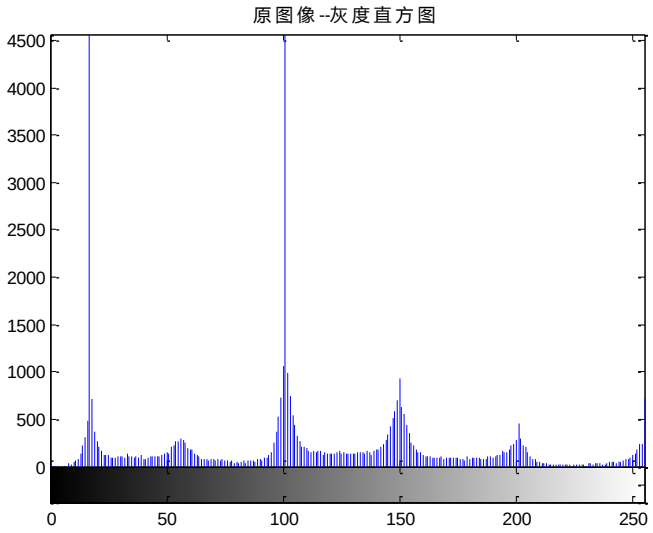
分割前的MRI图　　分割后的MRI图

计算脑图的灰质，白质，骨骼的占脑部比例。 brain.m

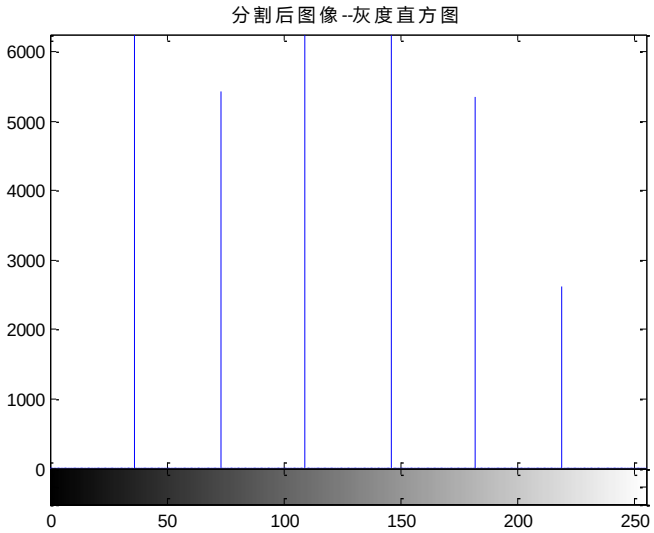
4.FCM在图像分割中应用

4.1 医学图像

图像分割前后2维直方图



图像分割



分割前MRI的直方图

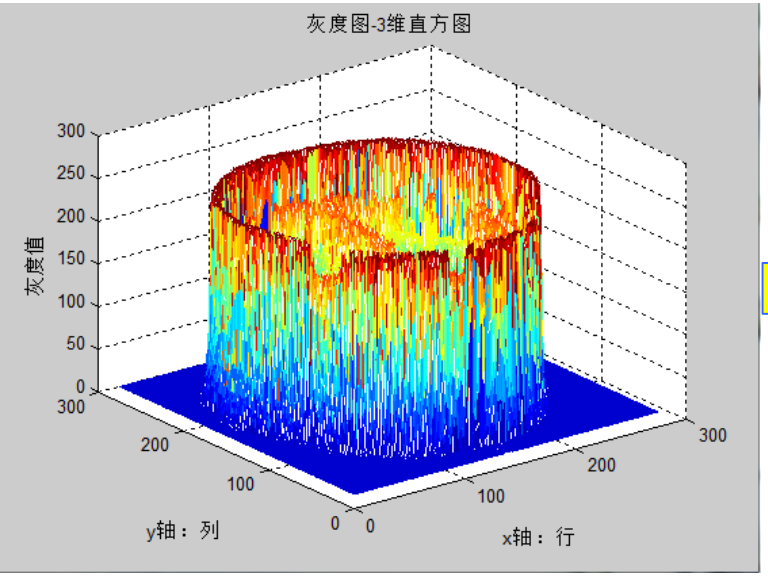
分割后MRI的直方图

计算脑图的灰质，白质，骨骼的占脑部比例。 brain.m

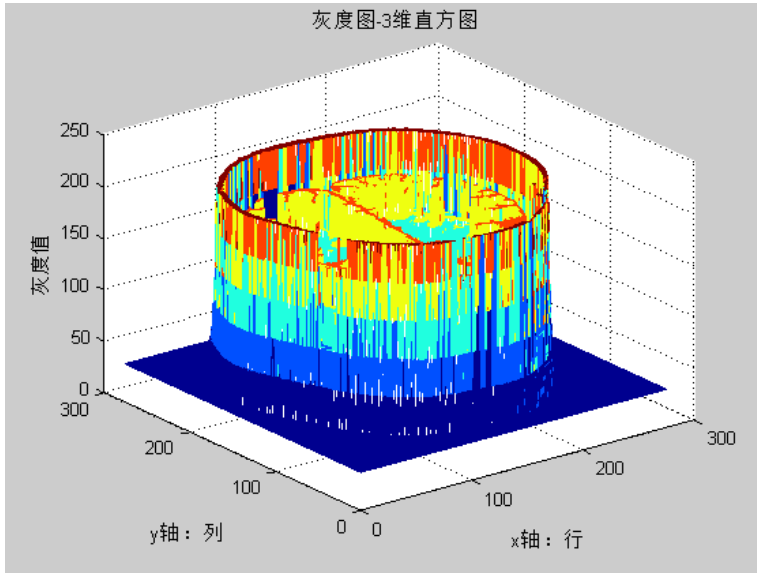
4.FCM在图像分割中应用

4.1 医学图像

图像分割前后三维直方图



图像分割



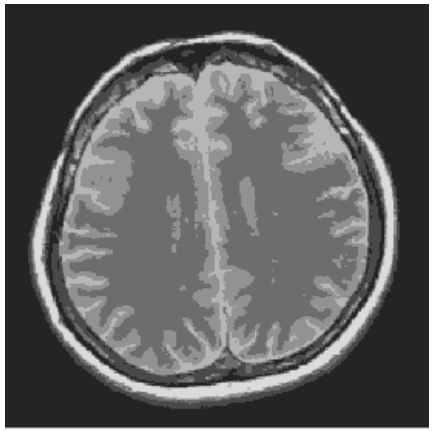
分割前MRI的直方图

分割后MRI的直方图

计算脑图的灰质，白质，骨骼的占脑部比例。 brain.m

4.FCM在图像分割中应用

4.1 医学图像



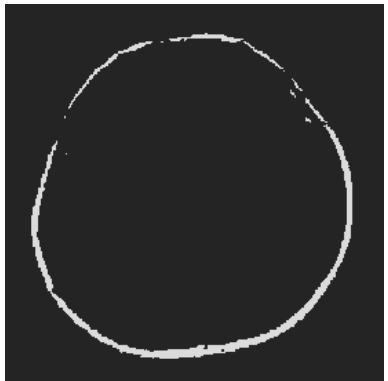
分割后的MRI图

特征提取

WM of Brain

GM of Brain

Brone of Brain



MRI脑图分割后，各部分占比例

brain占整幅图像的比例：0.57509

gray matter(GM)占brain的比例： 0.44423

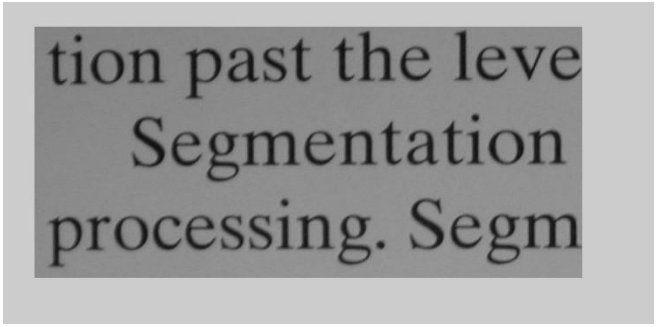
white matter(WM)占brain的比例：0.12268

bone占brain的比例： 0.059978

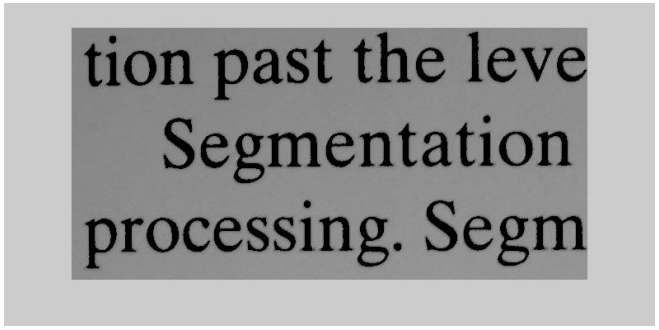
others占brain的比例：0.37311

4.FCM在图像分割中应用

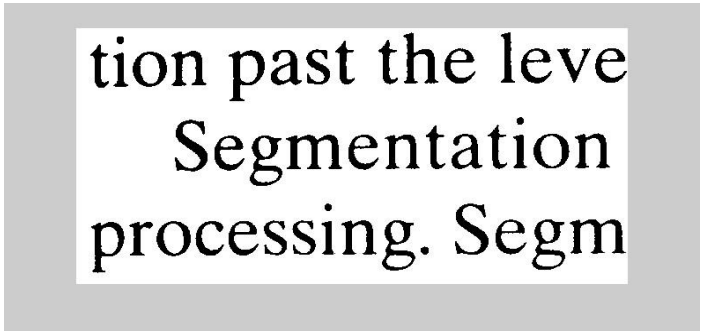
4.2 提取文本的字符 (word_code.m)



原图



分割图



设定阈值，
去除背景

4.FCM在图像分割中应用

4.3 目标提取

仿射传播聚类方法(AP聚类)

Affinity Propagation Clustering

Clustering by Passing Messages Between Data Points. Brendan J. Frey and Delbert Dueck. Science 315, 972~976, 2007.02.

算法评估

聚类评估主要包括如下任务：

- 估计聚类趋势
- 确定数据集中的簇数
- 测定聚类质量

算法评估—估计聚类趋势

- 聚类趋势评估确定给定的数据集是否具有可以导致有意义的聚类的非随机结构。

✓ 聚类要求数据的非均匀分布。

- “如何评估数据集的聚类趋势?”直观地看，我们可以评估数据集被均匀分布产生的概率。这可以通过空间随机性的统计检验来实现。为了解释这一思想，我们考虑一种简单但有效的统计量—霍普金斯统计量。
- 霍普金斯统计量是一种空间统计量，检验空间分布的变量的空间随机性。给定数据集D，它可以看做随机变量O的一个样本，我们想要确定O在多大程度上不同于数据空间中的均匀分布。

算法评估—估计聚类趋势

霍普金斯统计量-计算

（1）均匀地从D的空间中抽取n个点$p_1, \cdots, p_n$。也就是说，D的空间中的每个点都以 相同的概率包含在这个样本中。对于每个点$p_i(1 \le i \le n)$，我们找出

p_i在D中的最邻近，并令x_i为p_i与它在D中的最近邻之间的距离，即

$x_i=\min\{dist(p_i, v)\}$(其中$v \in D$)

（2）均匀地从D中抽取n个点$q_1, \cdots q_n$. 对于每个点$q_i(1 \le i \le n)$，我们找出q_i在$D-\{q_i\}$中的最邻近，并令y_i为q_i它在$D-\{q_i\}$中的最近邻之间的距离，即

$y_i=\min\{dist(q_i, v)\}$(其中$v \in D, v \neq q_i$)

（3）计算霍普金斯统计量H:

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

算法评估—估计聚类趋势

- “霍普金斯统计量告诉我们数据集D有多大可能遵循数据空间的均匀分布?”如果D是均匀分布的，则$\sum y_i$和$\sum x_i$将会很接近，因而H大约为0.5。然而，如果D是高度倾斜的，则$\sum y_i$将显著地小于$\sum x_i$，因而H将接近0。

- 我们的假设是同质假设——D是均匀分布的，因而不包含有意义的簇。非均匀假设(即D不是均匀分布，因而包含簇)是备择假设。我们可以迭代地进行霍普金斯统计量检验，使用0.5作为拒绝备择假设阈值，即如果H>0.5，则D不大可能具有统计显著的簇。

算法评估—确定簇数

- 确定数据集中”正确的”簇数是重要的，因为合适的簇数可以控制适当的聚类分析粒度，这可以看做在聚类分析的可压缩性与准确性之间寻找好的平衡点。
- 简单的经验方法：对于n个点的数据集，设置簇数p大约为√n/2.在期望情况下，每个簇大约有√2n个点。
- 肘方法：给点k>0,我们可以使用一种像k-均值这样的算法对数据集聚类，并计算簇内方差和—var(k).然后，我们绘制var关于k的曲线。曲线的第一个(或者最显著的)拐点暗示”正确的”簇数。

还有一些其他的方法，可以依情况选择合适的方法。

算法评估—测定聚类质量

- 对于测定聚类的质量，我们有几种方法可供选择。一般而言，根据是否有基准可用，这些方法可以分成两类。这里，基准是一种理想的聚类，通常由专家构建。
- 如果有基准可用，则外在方法可以使用它。外在方法比较聚类结构和基准。如果没有基准可用，则我们可以使用内在方法，通过考虑簇的分离情况评估聚类的好坏。基准可以看做一种簇标号形式的监督。因此，外在方法又称为监督方法，而内在方法是无监督方法。

算法评估—外在方法

外在方法核心：给定基准Cg，对聚类C赋予一个评分Q(C,Cg),一种外在方法是否有效很大程度上依赖于该方法使用的度量Q,度量Q应满足：

- 簇的同质性：要求聚类中的簇越纯，聚类越好
- 簇的完全性：要求对于聚类来说，根据基准如果两个对象属于相同的类别，则他们应该被分配到相同的簇。
- 小簇保持性：把小类别划分成小片比将大类别划分成小片更有害。

算法评估—内在方法

- 当没有数据集的基准可用时，我么必须使用内在方法来评估聚类的质量。一般而言，内在方法通过考察簇的分离情况和簇的紧凑情况来评估聚类。许多内在方法都利用数据集的对象之间的相似性度量。
- 轮廓系数：对于n个对象的数据集D，假设D被划分成k个簇$C_1,\cdots,C_K$. 对于每个对象o与o所属的簇的其他对象之间的平均距离y(o). 类似地，b(o)是o到不属于o的所有簇的最小平均距离。假设$o\in C_i(1\le i\le k)$,则

  $y(o)=(\sum dist(o,o'))/(|C_i|-1)\ (o'\in C_i, o\neq o')$ 而$b(o)=min\{\sum dist(o,o')/|C_i|\}\ (C_{j:1\le j\le k, j\neq i})$

  对象o的轮廓系数定义为s(o)=(b(o)-y(o))/max{y(o),b(o)}，其值在-1和1之间。
- y(o)反应o所属的簇的紧凑性，值越小越好；b(o)捕获o与其它簇的分离程度，值越大越好。当s(o)的值接近1时，包含o的簇是紧凑的并且远离其它簇，可取情况；当其值为负时，这意味在期望情况下，o距离其它簇的对象比距离与自己同在簇的对象更近，不可取情况。

作业

1、简述什么聚类？及常用聚类统计量。

2、简述k-均值与k-中心点方法的原理以及两种方法各自的优缺点。

3、实现一种常见的无监督聚类方法应用范例。