聚类分析

1 什么是聚类分析?

2 聚类分析中的数据类型

3 基于划分的聚类方法

4 基于层次的聚类方法

5 基于密度的聚类方法

学习目的

- 理解聚类与分类数据挖掘的区别。
- 掌握聚类的常用方法。

1 什么是聚类分析

聚类(Clustering):

◆聚类是一个将数据集划分为若干组（class）或类（cluster）的过程，并使得同一个组内的数据对象具有较高的相似度；而不同组中的数据对象是不相似的。

◆ 相似或不相似是基于数据描述属性的取值来确定的，通常利用各数据对象间的距离来进行表示。

◆聚类分析尤其适合用来探讨样本间的相互关联关系从而对一个样本结构做一个初步的评价。

示 例

表中给出9个顾客的购买信息，包括购买的商品的数量及价格，根据此两个特征量，将顾客聚类成3类（购买大量的高价产品；购买少量的高价产品；购买少量的低价产品）。

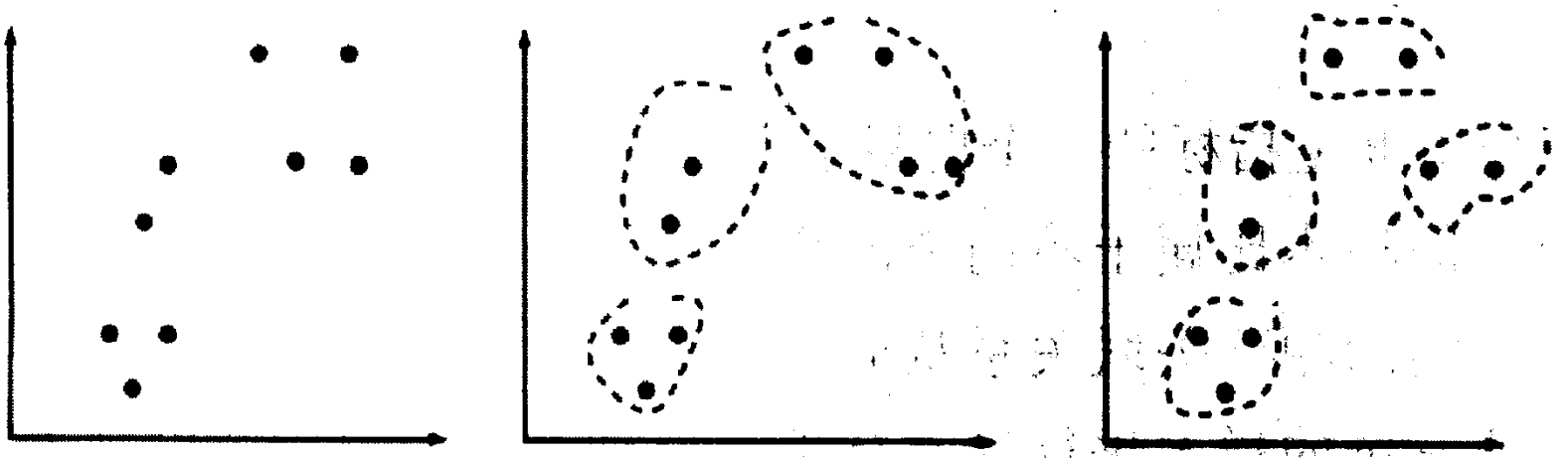
包含相似对象的类的样本集

	商品的数量	价 格
	2	1700
类 1	3	2000
	4	2300
	10	1800
类 2	12	2100
	11	2500
	2	100
类 3	3	200
	3	350

示 例

聚类是一个非常困难的事情，因为在一个n维样本空间中，数据可以以不同的形状和大小揭示类。

如在二维欧几里得空间中，上面数据可以分类三个类也可以分为四个类，类的数量的任意性是聚类过程中的主要问题。



a) 初始数据　　b) 数据的三个类　　c) 数据的四个类

二维空间的点的聚类分析

1 什么是聚类分析

聚类与分类的区别：

◆聚类是一 种**无（教师）监督**的学习方法。与分类不同，其不依赖于事先确定的数据类别，以及标有数据类别的学习训练样本集合。

◆ 因此，聚类是观察式学习，而不是示例式学习。

1 什么是聚类分析

聚类分析的应用：

◆**市场分析**：帮助市场分析人员从客户基本库中发现不同的客户群，并用购买模式刻画不同的客户群的特征；

◆**万维网**：对WEB日志的数据进行聚类，以发现相同的用户访问模式；

◆**图像处理**；

◆**模式识别**；

◆**孤立点检测等**。

1 什么是聚类分析

什么是好的聚类：

◆一个好的聚类方法将产生以下的高聚类：

➢最大化类内的相似性；

➢最小化类间的相似性。

◆聚类结果的质量依靠所使用度量的相似性和它的执行。

◆聚类方法的质量也可以用它发现一些或所有隐含模式的能力来度量。

2 聚类分析中的数据类型

- 基本的数据结构；
- 区间标度变量；
- 二元变量；
- 符号型、顺序型和比例数值型变量；
- 混合数据类型。

1. 基本的数据结构?

基本的数据结构

许多基于内存的聚类算法选择如下两种具有代表性的数据结构：

（1）数据矩阵；

（2）相异度矩阵。

（1）数据矩阵

数据矩阵：

是一个对象—属性结构，由n个对象组成，如：人；每个对象利用p个属性加以描述，如：年龄、身高、体重等。数据矩阵采用关系表形式或n*p矩阵来表示：

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

（2）相异度矩阵

相异度矩阵（差异矩阵）：

是一个对象—对象结构，存放n个对象两两之间的近似性（差异性），采用n*n的矩阵形式表示：

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \cdots & \cdots & \cdots & \cdots \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

（2）相异度矩阵

相异度矩阵（差异矩阵）：

其中$d(i,j)$表示对象i和对象j之间的差异（或不相似程度）。通常$d(i,j)$为一个非负数；当对象i和对象j非常相似或彼此“接近”时，该数值接近 0；该数值越大，就表示对象i和对象j越不相似。由于有$d(i,j)=d(j,i)$且$d(i,i)=0$，

所以，矩阵呈现出上三角或下三角的形式。

注意：

◆数据矩阵通常称为双模（two-mode）矩阵：行和列分布表示不同的实体；

◆相异度矩阵常被称为单模（one-mode）矩阵：行和列表示同一实体。

◆许多聚类算法都是以**相异度矩阵**为基础计算的，所以如果数据是以数据矩阵的形式给出的，则需要首先转换为相异度矩阵，才可以利用聚类算法来处理。

2. 区间标度变量?

什么是区间标度变量

区间标度变量（间隔数值变量）：

- 基本呈**直线比例的连续变量**，如：重量、高度和温度等。

为什么标准化？

- 通常，选用的度量单位将直接影响聚类分析的结果，如：将高度的度量单位由“米”变为“英尺”，或将重量的单位由“千克”变为“英镑”，可能会产生非常不同的聚类结构。
- 一般，度量单位越小，变量可能的值域越大，对聚类结果的影响也越大。因此，为避免对度量单位选择的依赖，数据应当标准化。

度量值的标准化

为了实现标准化，一种方法是将初始测量值转换为无单位变量。给定一个属性变量*f*，可用如下公式对其进行标准化：

（1）计算平均的绝对偏差

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

其中$x_{1f}, x_{2f}, \cdots, x_{nf}$是变量f的n个测量值；$m_f$为变量f的均值，即：

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + ... + x_{nf})$$

（2）计算标准化测量（z-score）：

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

度量值的标准化

在标准化之后，或在无需标准化的特定应用中，由间隔数值所描述对象之间的差异（或相似）程度可以通过计算相应两个对象之间距离来确定。最常用的距离计算公式就是欧氏距离（Euclidean distance），具体公式内容如下：

$$d(i,j)=\sqrt{(|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+\cdots+|x_{ip}-x_{jp}|^2)}$$

其中 $i=(x_{i1},x_{i2},\ldots,x_{ip})$；$j=(x_{j1},x_{j2},\ldots,x_{jp})$；它们分别表示一个 $p-$维数据对象。

度量值的标准化

另一个常用的距离计算方法就是Manhattan距离，它的具体计算公式定义如下：

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\cdots+|x_{ip}-x_{jp}|$$

欧氏距离和Manhattan距离均满足距离函数的有关数学性质（要求）：

- $d(i,j)\geq 0$，这表示对象之间距离为非负数的一个数值；
- $d(i,i)=0$；这表示对象自身之间距离为零；
- $d(i,j)=d(j,i)$；这表示对象之间距离是对称函数；
- $d(i,j)\leq d(i,h)+d(h,j)$；这表示对象自身之间距离满足“两边之和不小于第三边”的性质；若将两个对象之间距离用一条边来表示的话；其中h为第三个对象。

度量值的标准化

Minkowski 距离是欧式距离和 Manhattan 距离的一个推广，它的计算公式定义如下：

$$d(i,j)=(|x_{i1}-x_{j1}|^q+|x_{i2}-x_{j2}|^q+\cdots+|x_{ip}-x_{jp}|^q)^{1/q}$$

其中 q 为一个正整数；当 $q=1$ 时，它代表 Manhattan 距离计算公式；而当 $q=2$ 时，它代表欧氏距离计算公式。

若每个变量均可被赋予一个权值，以表示其所代表属性的重要性。那么带权的欧氏距离计算公式就是：

$$d(i,j)=\sqrt{w_1|x_{i1}-x_{j1}|^2+w_2|x_{i2}-x_{j2}|^2+\cdots+w_p|x_{ip}-x_{jp}|^2)}$$

同样，Manhattan 距离和 Minkowski 距离也可以引入权值进行计算。

3. 二元变量?

什么是二元变量

二元变量（二值变量）：

- 一个二元变量只有两个状态：0或者1。其中0代表变量所表示的状态不存在；1则代表相应的状态存在。
- 如：给定变量smoker，用以描述一个病人是否吸烟的情况，如用smoker为1表示病人吸烟；若smoker为0表示病人不吸烟。

二元变量的相异度计算

差异矩阵法：

如果假设所有的二元变量有相同的权重，则可以得到一个两行两列（2*2）的条件表。

对象i \ 对象j	1	0	合计
1	q	r	$q+r$
0	s	t	$s+t$
合计	$q+s$	$r+t$	p

二元变量的相异度计算

其中：

- q表示在对象i和对象j中均取1的二值变量个数；
- r表示在对象i取1但对象j中取0的二值变量个数；
- s表示在对象i中取0而在对象j中取1的二值变量个数；
- t则表示在对象i和对象j中均取0的二值变量个数。
- 二值变量的总数为p，则：p=q+r+s+t。

对称？不对称？

- 如果一个二值变量取0或1所表示的内容同等价值，且有相同的权重，则该二元变量是对称的。如，属性“性别”，有两个值“女性”和“男性”，两个取值都没有优先权 。
- 基于对称二元变量的相似度，称为恒定的相似度。
- 对恒定相似度而言，评价量对象i和j间相异度的最著名的系数是简单匹配系数：

$$d(i,j)=\frac{r+s}{q+r+s+t}$$

对称？不对称？

- 如果一个二值变量的两个取值的重要性不同等重要，则该二元变量就是不对称的。如一个疾病disease的测试结果positive或negative，显然这两个测试结果的重要性是不一样的：
- 通常将比较重要的输出结果，编码为1；而将另一结果编码为0.
- 给定一个二元变量，如果认为取0值比取0值所表示的情况更重要，则这样的二元变量被认为是单性的（好像只有一个状态）。

对称？不对称？

- 基于这样的二元变量的相似度被称为非恒定的相似度。
- 对非恒定相似度，最常见的描述对象i和对象j间差异度的参数是Jaccard相关系数：

$$d(i,j)=\frac{r+s}{q+r+s}$$

- 在计算过程中，负匹配的数目t被认为是不重要的，因此被忽略。
- 若一个数据集中既包括对称二元变量，又包含不对称二元变量，可以用混合变量方法来处理。

示例

示例 二值变量的差异性。假设一个病人记录表如表 1 所示；表中所描述的属性（变量）分别为 *name*、*gender*、*fever*、*cough*、*test-1*、*test-2*、*test-3* 和 *test-4*；其中 *name* 作为（病人）对象的标识； *gender*（性别）是一个对称二值变量。其它变量则均为非对称变量。

name	*gender*	*fever*	*cough*	*test-1*	*test-2*	*test-3*	*test-4*
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

表1 一个包含许多二值属性的关系数据表示意描述

示例

对于非对称属性（变量）值，可将其 Y 和 P 设为 1；N 设为 0。根据非对称变量计算不同对象（病人）间的距离（差异性），就可以利用 Jaccard 相关系数计算公式（6.10）进行，具体计算结果如下：

$$d(Jack, Mary) = \frac{0+1}{2+0+1} = 0.33;$$

$$d(Jack, Jim) = \frac{1+1}{1+1+1} = 0.67;$$

$$d(Jim, Mary) = \frac{1+2}{1+1+2} = 0.75;$$

上述计算值表明：Jim 和 Mary，由于他们之间距离值（差异性）三个中最大，因此不太可能得的是相似的病；而 Jack 和 Mary，由于他们之间距离值（差异性）三个中是最小，因此可能得的就是相似的病。 ■

4. 符号、顺序和比例数值变量?

（1）符号变量

符号变量（标称变量）：

- 符号变量是二元变量的推广，可具有多于两个的状态值，如颜色变量（红、橙、黄、绿、蓝等）。
- 设一个符号变量所取的状态个数为M，其中的状态可以用字母、符号，或一个整数集合来表示，如1,2,...M。此处的整数仅是为方便数据处理而采用的，并不代表任何的特定的顺序。

（1）符号变量

对于符号变量，最常用的计算对象i和对象j之间差异（程度）的方法就是简单匹配方法.

$$d(i,j)=\frac{p-m}{p}$$

其中m表示对象i和对象j中取同样状态的符号变量个数（匹配数）；p为所有的符号变量个数。

（2）顺序变量

顺序变量（序数型变量）：

- 一个离散的顺序变量类似于符号变量，但不同的是顺序变量的M个状态是以有意义的顺序进行排列的。
- 如专业等级是一个顺序变量，是按照助教、讲师、副教授和教授的顺序排列的。
- 一个连续的顺序变量，值的相对位置要比它的实际数值有意义的多，如某个比赛的相对排名（金牌、银牌和铜牌）可能比实际得分更重要。

顺序变量的相异度

顺序变量的处理与区间标度变量非常类似，假设f是用于描述n个对象的一组顺序变量之一，关于f的相异度计算如下：

- 第 i 个对象的 f 变量值标记为 x_{if}，变量 f 有 M_f 个有序状态，可以利用等级 1,2,…, M_f 分别替换相应的 x_{if}，得到相应的 r_{if}，$r_{if} \in \{1,2,\ldots,M_f\}$；
- 由于每个顺序变量的状态个数可能不同。因此有必要将每个顺序变量的取值范围映射到[0 –1]区间，以便使每个变量的权值相同。可以通过将第 i 个对象中的第 f 个变量的 r_{if} 用以下所计算得到的值来替换：

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

接下来就可以用区间标度变量中所描述的任意一组距离度量方法进行计算相异度。

（3）比例数值变量

比例数值变量（比例标度型变量）：

➢一个比例数值变量指在非线性的标度上取正的度量值的变量，如指数比例：

$$Ae^{Bt} \text{ 或 } Ae^{-Bt}$$

其中 A 和 B 为正的常数。典型例子包括：细菌繁殖增长的数目描述，或放射元素的衰减。

（3）比例数值变量

在计算比例数值变量所描述对象间的距离时，有两种处理方法：

1）将比例数值变量看作区间标度变量，采用相同的方法处理，但不佳，因为比例尺度是非线性的；

2）采用对数变换 $y_{if} = \log(x_{if})$ 对比例数值变量进行处理，然后将y_{if}当做区间标度变量来处理。

5. 混合数据类型?

混合数据类型

混合数据类型：

➢在实际数据库中，数据对象往往是用复合数据类型来描述的，而且常常包括以上六种数据类型：区间标度变量、对称二元变量、不对称二元变量、符号类型、顺序类型和比例数值类型。

如何计算相异度？

➢ 一种方法是将变量按类型分组，对每种类型的变量单独聚类分析，如果分析得对兼容的结果，这种方法可行，但实际中，往往不可行。

➢ 一种更可取的方法是将所有的变量一起处理，只进行一次聚类分析。

混合数据类型

- 一种技术是将不同类型的变量组合在单个相异度矩阵中，把所有有意义的变量转换到共同的值域区间[0,1]上。
- 假设数据集包含p个不同类型的变量，对象i和j间的相异度d(i,j)定义为：

$$d(i,j)=\frac{\sum_{f=1}^{p}\delta_{ij}^{(f)}d_{ij}^{(f)}}{\sum_{f=1}^{p}\delta_{ij}^{(f)}}$$

其中如果（1）x_{if}或x_{jf}数据不存在（对象i或对象j的变量f无测量值）；或（2）$x_{if}=x_{jf}=0$且变量f为非对称二值变量，则标记$\delta_{ij}^{(f)}=0$；否则$\delta_{ij}^{(f)}=1$。

混合数据类型

变量f对i和j直接相异度的计算方式与其具体类型有关：

（1） 若变量 f 为二值变量或符号变量，则如果 $x_{if} = x_{jf}$，那么 $d_{ij}^{(f)} = 0$；否则 $d_{ij}^{(f)} = 1$。

（2） 若变量 f 为间隔数值变量，则 $d_{ij}^{(f)} = \dfrac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$；其中 h 为变量 f 所有可能的对象。

（3） 若变量 f 为顺序变量或比例数值变量，则计算顺序 r_{if} 和 $z_{if} = \dfrac{r_{if} - 1}{M_f - 1}$，并将 z_{if} 当作间隔数值变量来进行计算处理。

4 基于划分的聚类方法

目前，在文献报道中有大量的聚类算法，算法的选择主要取决于数据的类型、聚类的目的和应用。如果聚类分析被用作描述或探索性的攻击，则可以对同样的数据尝试多种算法，以发现数据可能揭示的结果。

主要的聚类分析方法

大体上，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。

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　：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。

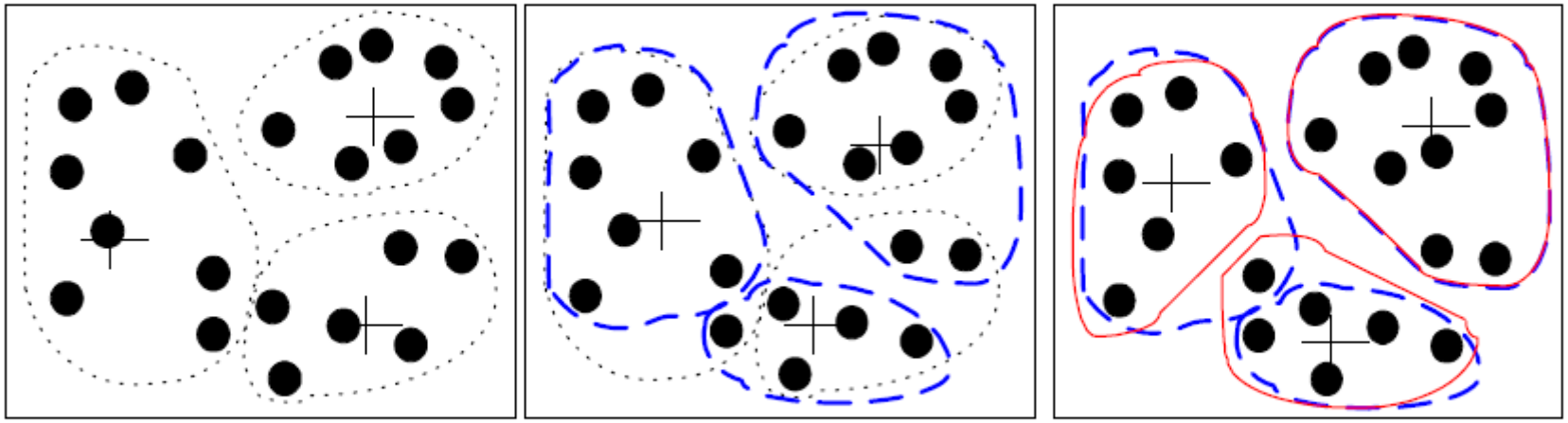
K-平均聚类算法

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

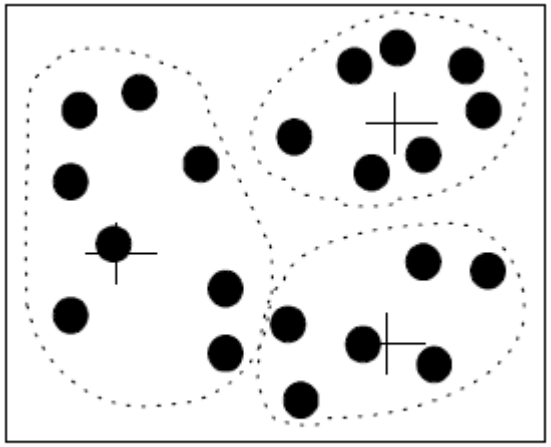
示例 假设空间数据对象分布如图 （a）所示，设$k=3$，也就是需要将数据集划分为三份（聚类）。



k-means 算法聚类过程示意描述

示例

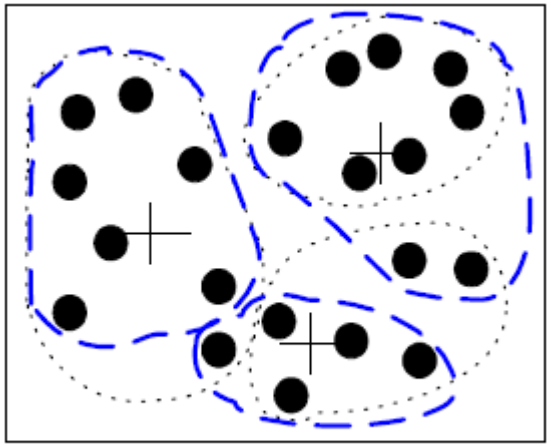
根据算法　　，从数据集中任意选择三个对象作为初始聚类中心（图　　（a）中这些对象被标上了“+”）；其余对象则根据与这三个聚类中心（对象）的距离，根据最近距离原则，逐个分别聚类到这三个聚类中心所代表的（三个）聚类中；由此获得了如图　　（a）所示的三个聚类（以虚线圈出）。



(a)

示例

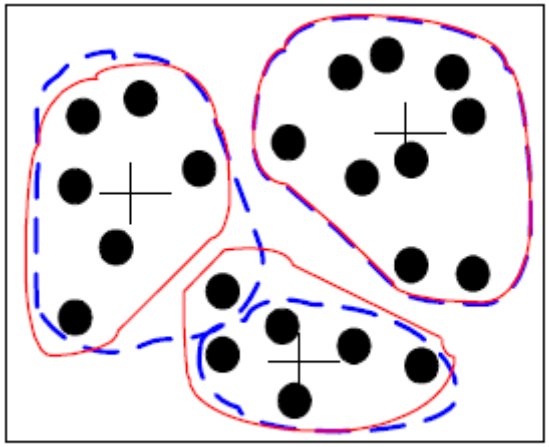
在完成第一轮聚类之后，各聚类中心发生了变化；继而更新三个聚类的聚类中心（图　　（b）中这些对象被标上了“+”）；也就是分别根据各聚类中的对象计算相应聚类的（对象）均值。根据所获得的三个新聚类中心，以及各对象与这三个聚类中心的距离，（根据最近距离原则）对所有对象进行重新归类。有关变化情况如图　　（b）所示（已用粗虚线圈出）。



(b)

示例

再次重复上述过程就可获得如图　（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。

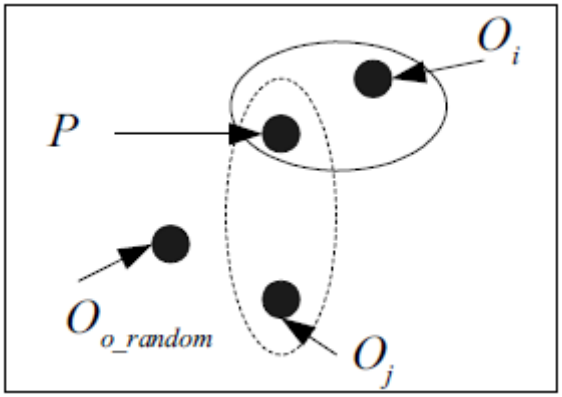
◆ 然后重复地用非中心点对象来替代中心对象，如果它改善了结果聚类的整体距离，则进行替代。

◆ 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；



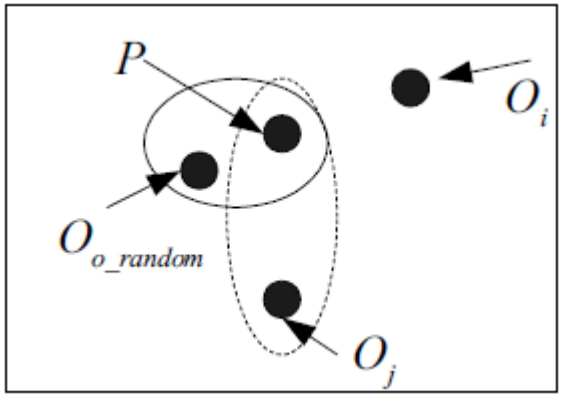
重新分配给O_i

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；



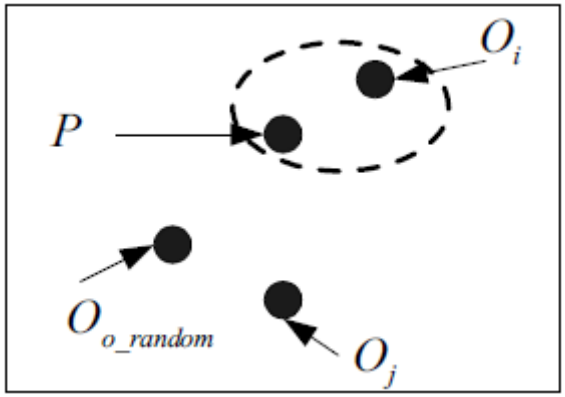
重新分配给O_{random}

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；



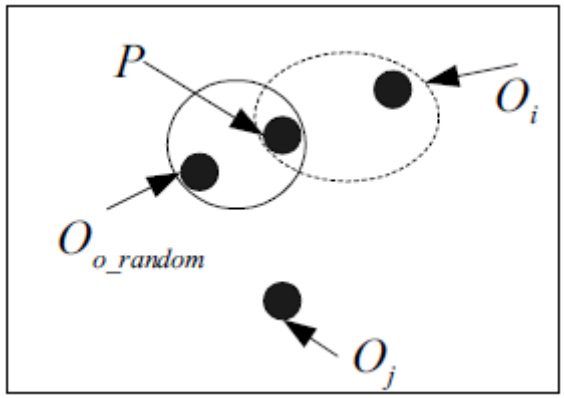
不发生变化

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　　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。

基于层次的聚类方法

大体上，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。

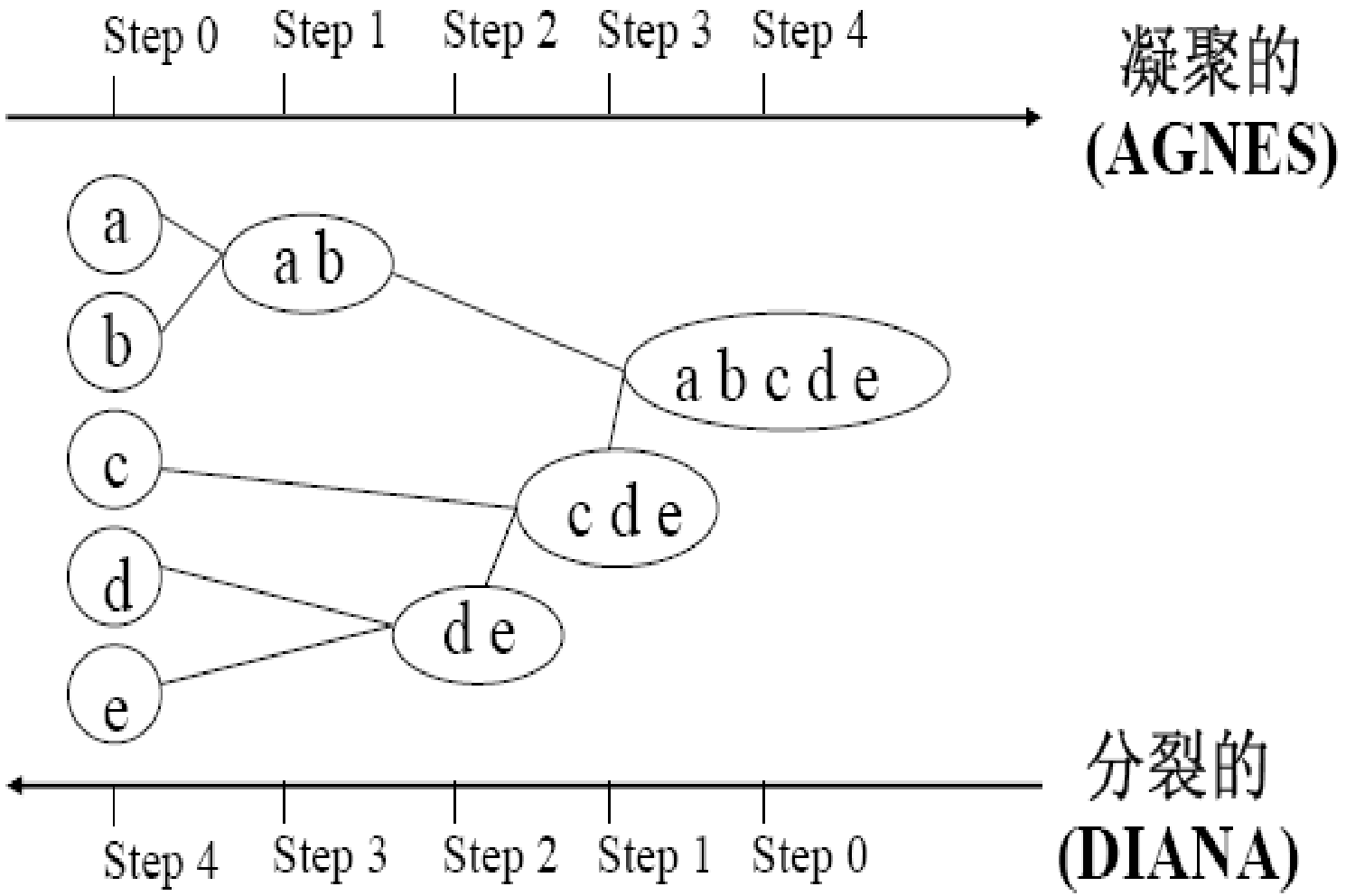
层次方法

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

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；

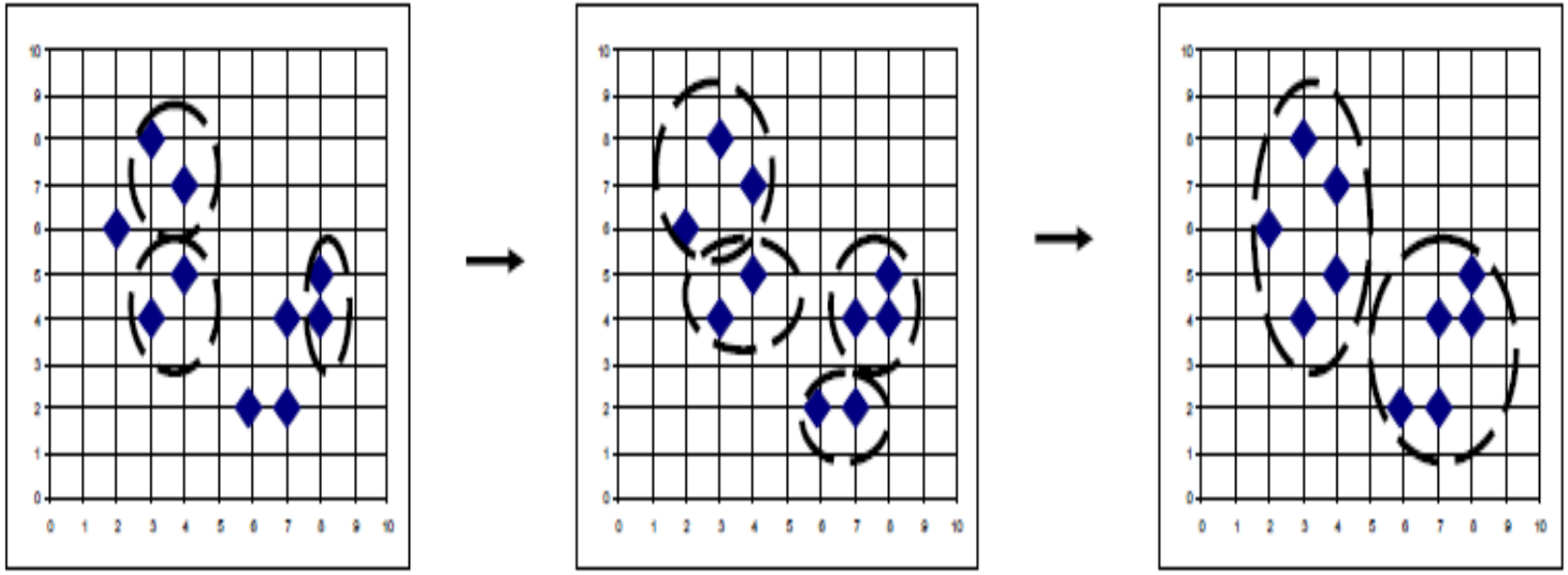
(2) REPEAT

(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

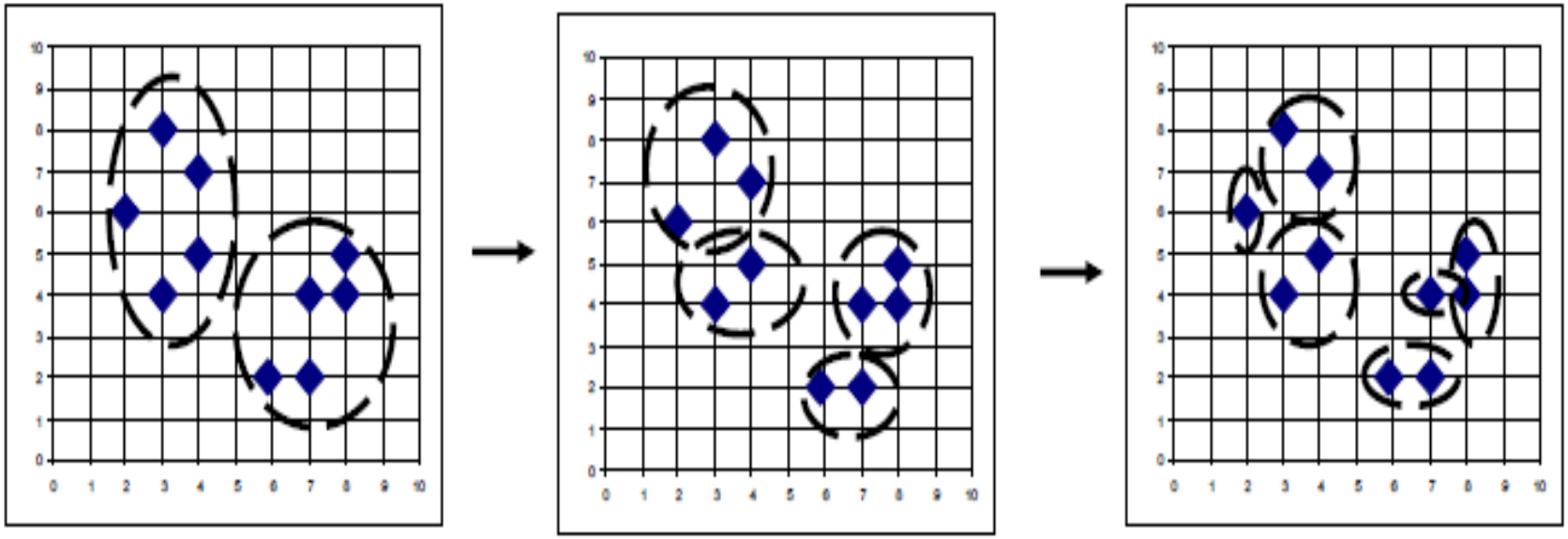
- DIANA 算法：与AGNES算法相反，初始所有节点都在一个大簇中，根据某些准则被一步步地分解，直到达到初始设定的簇数目。
- 聚类过程中，DIANA算法将用到如下两种测度方法：
  - **簇的直径**：一个簇中的任意两个数据点的距离中的最大值；
  - **平均相异度**（平均距离）：

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

DIANA算法

算法 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.

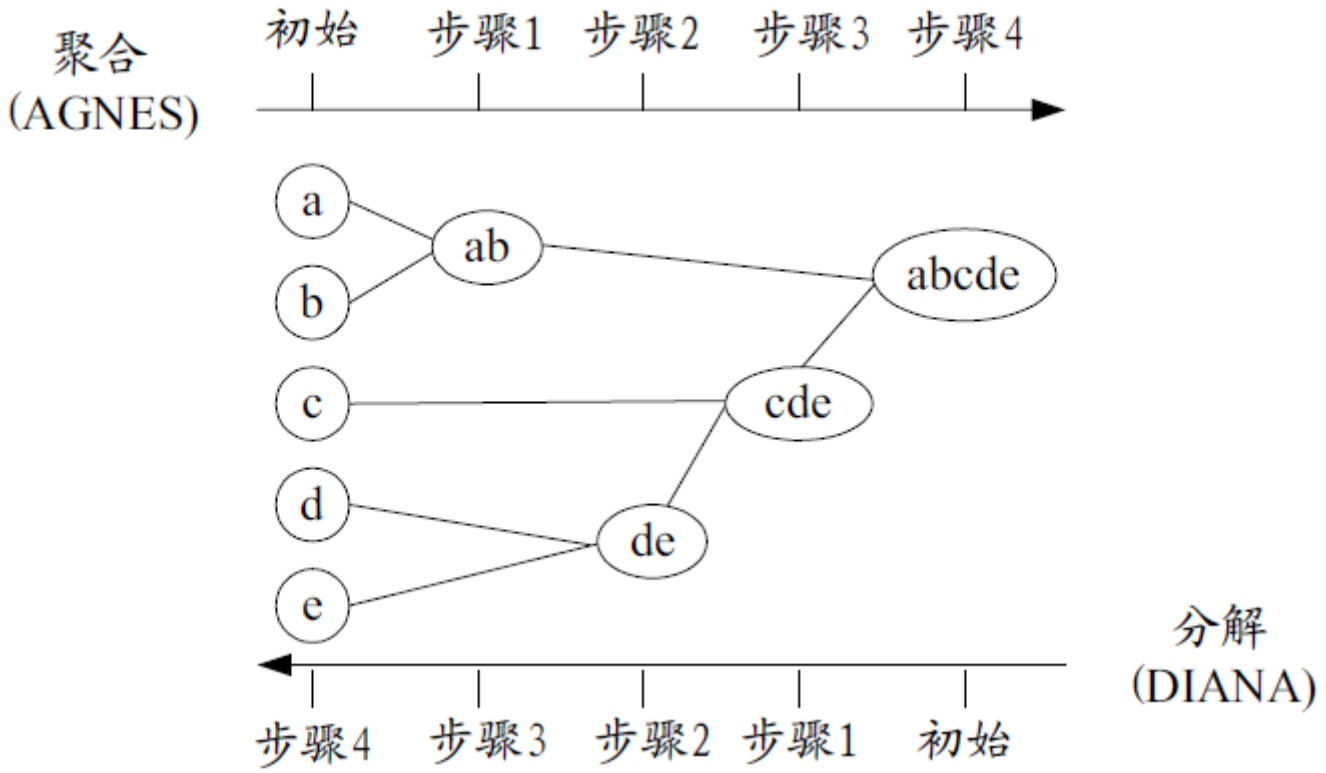
DIANA算法



DIANA算法示意图

示例

示例　　如图　　所示，就分别是一个自下而上聚合层次聚类方法 AGNES（AGglomerative NESting）和一个自顶而下分解层次聚类方法 DIANA（DIvsia ANAlysia）的应用示例。其中数据集为 $\{a,b,c,d,e\}$，共有 5 个对象。



示例

1）最初AGNES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合并。

- 如，如果簇C_1中的一个对象和簇C_2中的一个对象之间的距离是所有属于不同簇的对象间欧氏距离最小的， C_1和C_2可能被合并。
- 这是一种单链接方法：每个簇可以被簇中所有对象代表，两个簇间的相似度由这两个不同簇中距离最近的数据点对的相似度确定。
- 聚类的合并过程反复进行直到所有对象最终合并形成一个簇。

示例

2）在分裂层次DIANA方法中，所有对象初始都放在一个簇中，根据一些原则（如簇中对象的最大欧氏距离），将该簇分裂。簇的分裂过程反复进行，直到最终每个新的簇只包含一个对象。

注意：

在这两种层次方法中，用户可以定义一个希望得到的簇的数目来作为结束条件。

基于层次的聚类方法

大体上，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。

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。

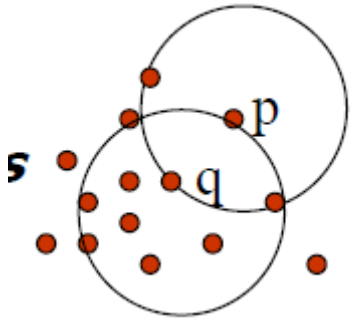
先介绍该方法中涉及到的一些基本的定义。

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。

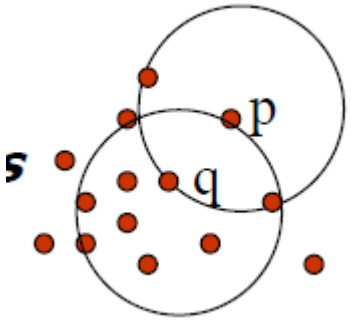
例如，在下图中，设定ε=1cm，MinPts=5，则q是一个核心对象。



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。

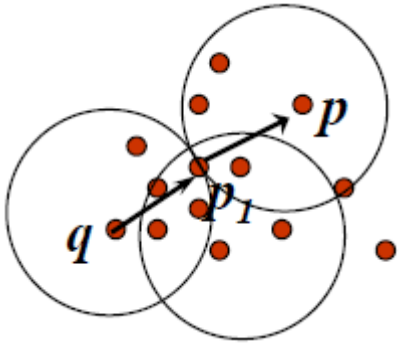
例如，在下图中，设定ε=1cm，MinPts=5， q是一个核心对象，对象p从对象q出发是直接密度可达的。



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。

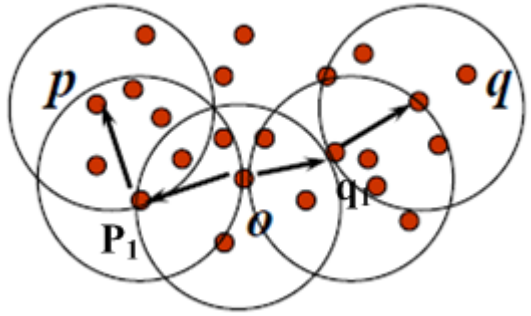
例如，在下图中，ε=1cm，MinPts=5，q是一个核心对象，p_1是从q关于ε和MitPts直接密度可达，p是从p_1关于ε和MitPts直接密度可达，则对象p从对象q关于ε和MinPts密度可达的。



基于密度的方法：DBSCAN

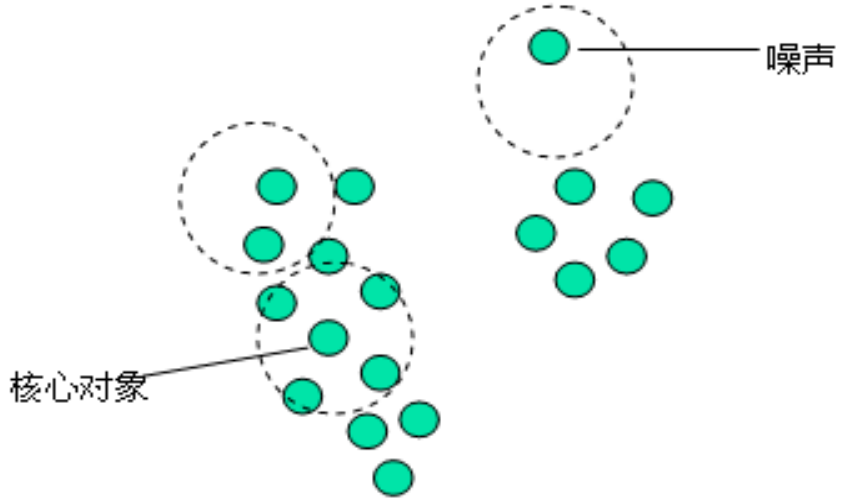
定义 5： 密度相连的：如果对象集合D中存在一个对象o，使得对象p和q是从o关于ε和MinPts密度可达的，那么对象p和q是关于ε和MinPts密度相连的。

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

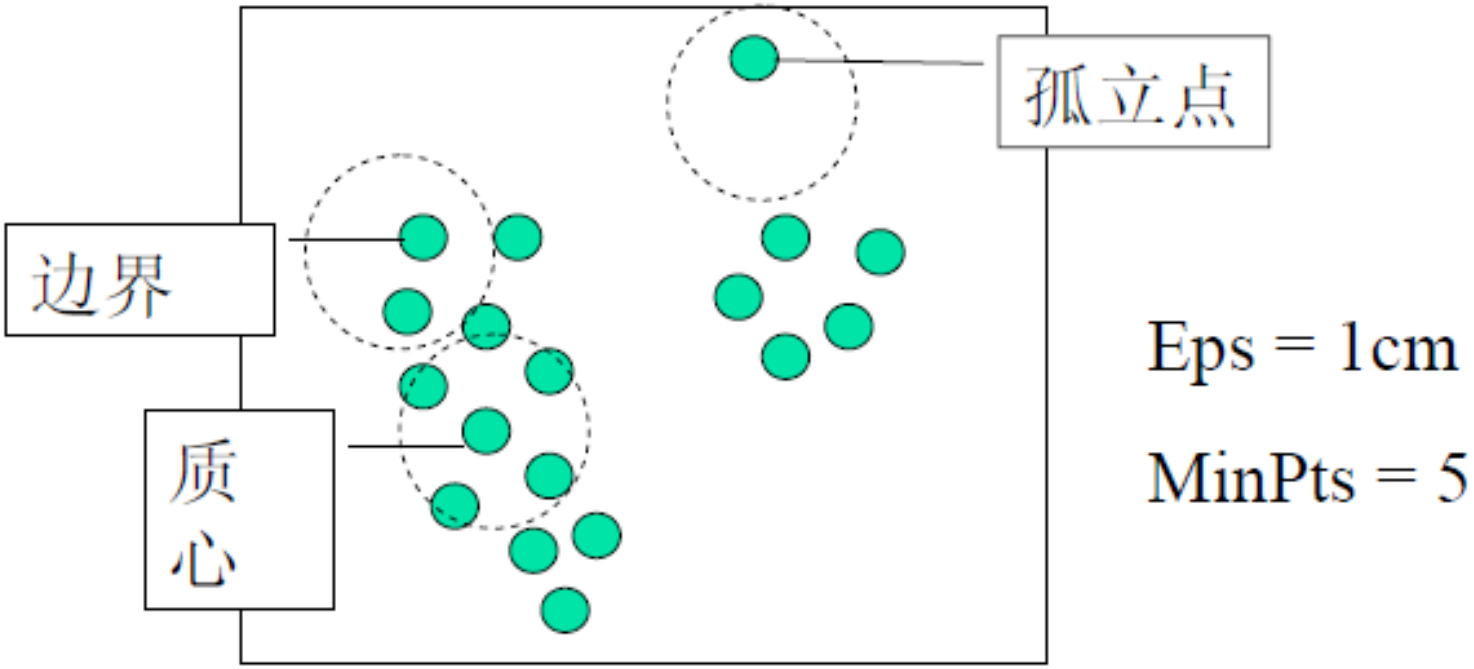
算法 DBSCAN

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



复习与思考问题

1. 基于划分的聚类方法？
2. 基于层次的聚类方法？
3. 基于密度的聚类方法？