



第六章

聚类分析（模式分析）

任课教师：柳欣老师

email: starxliu@163.com



第六章 聚类分析 (模式分析)

(Clustering Analysis)

- 6. 1 聚类分析的概念
- 6. 2 模式相似性测度
- 6. 3 类的定义与类间距离
- 6. 4 聚类的算法



6.1 聚类分析的概念

一、聚类分析的基本思想

- ★相似的归为一类。
- ★模式相似性的度量和聚类算法。
- ★无监督分类 (Unsupervised) 。

二、特征量的类型

- ★物理量----(重量、长度、速度)
- ★次序量----(等级、技能、学识)
- ★名义量----(性别、状态、种类)

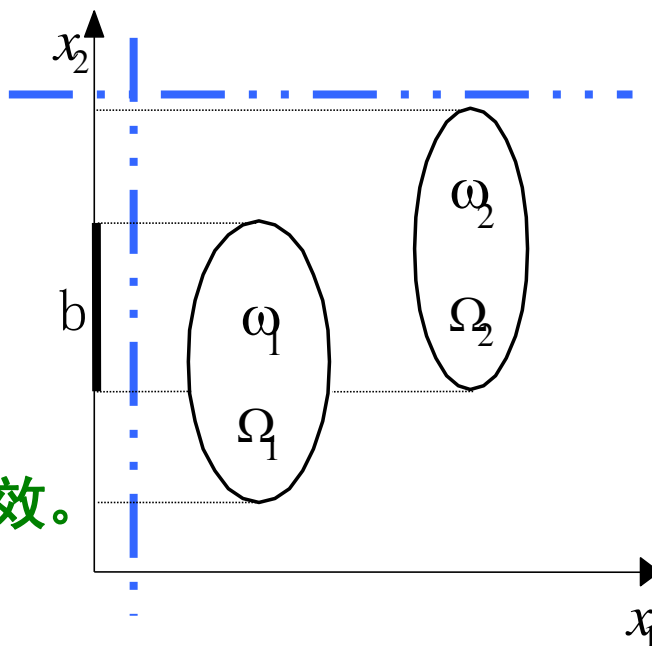
6.1 聚类分析的概念

三、方法的有效性

取决于分类算法和特征点分布情况的匹配。

分类无效时的情况

1. 特征选取不当使分类无效。



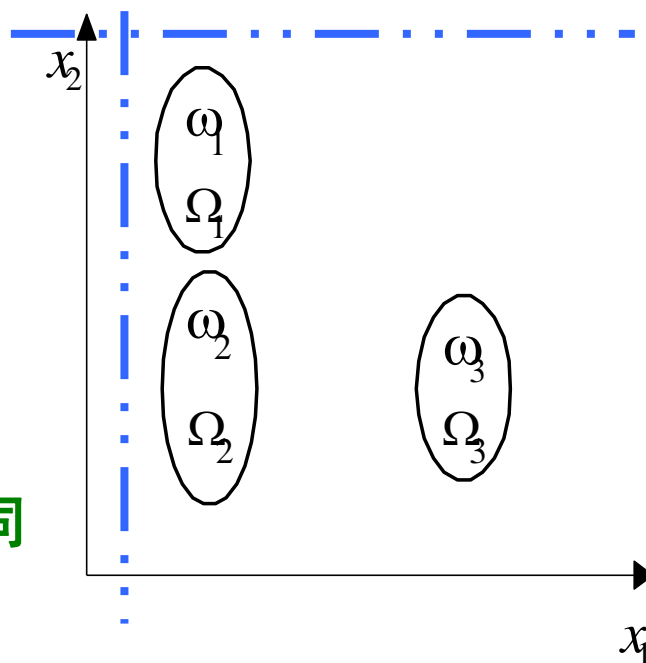
6.1 聚类分析的概念

三、方法的有效性

取决于分类算法和特征点分布情况的匹配。

分类无效时的情况

2. 特征选取不足可能使不同类别的模式判为一类。



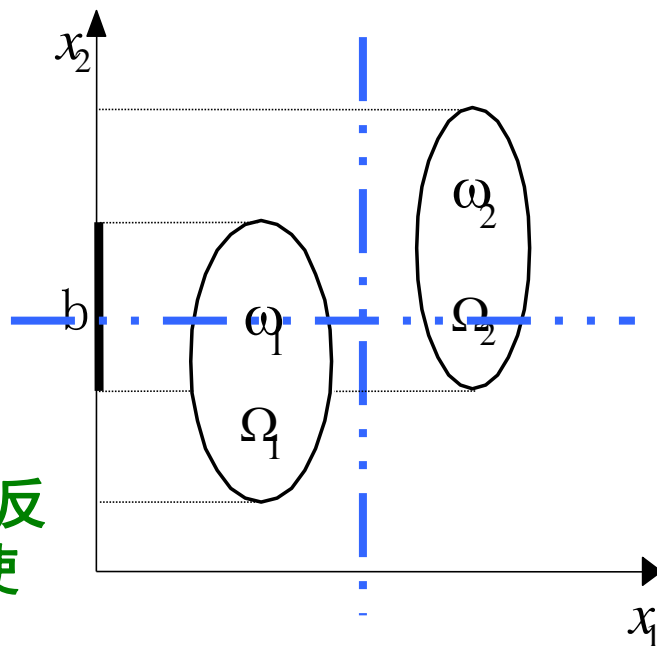
6.1 聚类分析的概念

三、方法的有效性

取决于分类算法和特征点分布情况的匹配。

分类无效时的情况

3. 特征选取过多可能无益反而有害, 增加分析负担并使分析效果变差。



特征选取不同对聚类结果的影响

下列是一些动物的名称：

羊 (sheep)

狗 (dog)

蓝鲨 (blue shark)

蜥蜴 (lizard)

毒蛇 (viper)

猫 (cat)

麻雀 (sparrow)

海鸥 (seagull)

金鱼 (gold fish)

绯鲉鲤 (red-mullet)

蛙 (frog)

要对这些动物进行分类，则不同的特征有不同的分法：

特征选取不同对聚类结果的影响

(a) 按繁衍后代的方式分

羊, 狗, 猫
蓝鲨

哺乳动物

蜥蜴, 毒蛇,
麻雀, 海鸥, 金鱼,
绯鲋鲤, 青蛙

非哺乳动物

特征选取不同对聚类结果的影响

(b) 按肺是否存在分

金鱼
鲱鳎鲤
蓝鲨

无肺

羊, 狗, 猫
蜥蜴, 毒蛇
麻雀, 海鸥
青蛙

有肺

特征选取不同对聚类结果的影响

(c) 按生活环境分

羊, 狗, 猫
蜥蜴, 毒蛇
麻雀, 海鸥

陆地

金鱼
绯鲋鲤
蓝鲨

水里

青
蛙

两栖

特征选取不同对聚类结果的影响

(d) 按繁衍后代方式和肺是否存在分

蜥蜴, 毒蛇
麻雀, 海鸥
青蛙

非哺乳且有肺

金鱼
鲱鲉
鲚

非哺乳且无肺

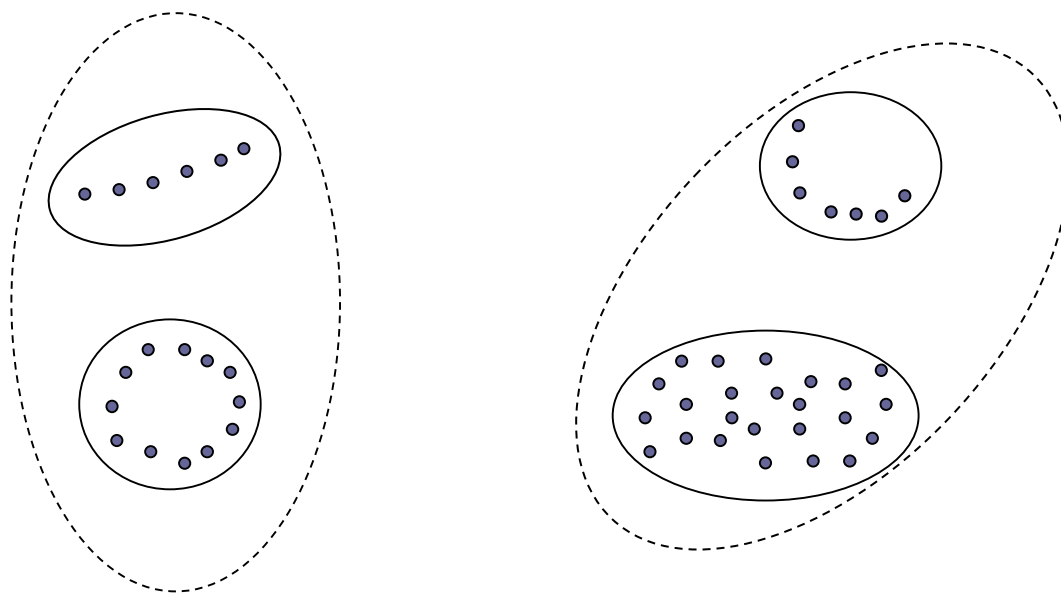
羊, 狗, 猫

哺乳且有肺

蓝鲨

哺乳且无肺

距离测度不同,聚类结果也不同



数据的粗聚类是两类,细聚类为4类



特征作用

选择什么特征？

选择多少个特征？

选择什么样的量纲？

选择什么样的距离测度？

这些对分类结果都会产生极大影响。



聚类过程遵循的基本步骤

- 一、特征选择(feature selection)
尽可能多地包含任务关心的信息
- 二、近邻测度(proximity measure)
定量测定两特征如何“相似”或“不相似”
- 三、聚类准则 (clustering criterion)
以蕴涵在数据集中类的类型为基础
- 四、聚类算法 (clustering algorithm)
接近邻测度和聚类准则揭示数据集的聚类结构
- 五、结果验证 (validation of the results)
常用逼近检验验证聚类结果的正确性
- 六、结果判定 (interpretation of the results)
由专家用其他方法判定结果的正确性



聚类应用的四个基本方向

一、减少数据

许多时候，当数据量 N 很大时，会使数据处理变得很费力。因此可使用聚类分析的方法将数据分成几组可判断的聚类 m ($m \ll N$) 来处理，每一个类可当作独立实体来对待。从这个角度看，数据被压缩了。

二、假说生成

在这种情况下，为了推导出数据性质的一些假说，对数据集进行聚类分析。因此，这里使用聚类作为建立假说的方法，然后用其他数据集验证这些假说。



聚类应用的四个基本方向

三、假说检验

用聚类分析来验证指定假说的有效性。

例如：考虑这样的假说“**大公司在海外投资**”。

要验证这个假说是否正确，就要对大公司和有代表性的公司按规模、海外活跃度、成功完成项目的能力等进行聚类分析。从而来支持这个假说。

四、基于分组的预测

对现有数据进行聚类分析，形成模式的特征，并用特征表示聚类，接下来，对于一个未知模式，就可以用前面的聚类来确定是哪一类？

例如：考虑被同种疾病感染的病人数据集。先按聚类分析进行分类，然后对新的病人确定他适合的聚类，从而判断他病情。



6.2 模式相似性测度

用于描述各模式之间特征的相似程度

- 距离测度
- 相似测度
- 匹配测度

6.2 模式相似性测度

一、距离测度(差值测度)

测度基础：两个矢量矢端的距离

测度数值：两矢量各相应分量之差的函数。

两矢量的距离定义应满足下面的公理：

设矢量 \vec{x} 和 \vec{y} 的距离记为 $d(\vec{x}, \vec{y})$,

(1) $d(\vec{x}, \vec{y}) \geq 0$ ，当且仅当 $\vec{y} = \vec{x}$ 时，等号成立；

(2) $d(\vec{x}, \vec{y}) = d(\vec{y}, \vec{x})$

(3) $d(\vec{x}, \vec{y}) \leq d(\vec{x}, \vec{z}) + d(\vec{z}, \vec{y})$

6.2 模式相似性测度

常用的距离测度有： $\bar{x}=(x_1, x_2, \dots, x_n)'$, $\bar{y}=(y_1, y_2, \dots, y_n)'$

1. 欧氏 (Euclidean) 距离

$$d(\bar{x}, \bar{y}) = \|\bar{x} - \bar{y}\| = \left[\sum_{i=1}^n (x_i - y_i)^2 \right]^{1/2}$$

距离越小，越相似。

注意：

- 1) 各特征向量对应的维上应当是相同的物理量；
注意物理量的单位。

某些维上物理量采用的单位发生变化，会导致对同样的点集出现不同聚类结果的现象。

6.2 模式相似性测度

2. 绝对值距离(街坊距离或Manhattan距离)

$$d(\bar{x}, \bar{y}) = \sum_{i=1}^n |x_i - y_i|$$

3. 切氏(Chebyshev)距离

$$d(\bar{x}, \bar{y}) = \max_i |x_i - y_i|$$

6.2 模式相似性测度

4. 明氏 (Minkowski) 距离

$$d(\vec{x}, \vec{y}) = \left[\sum_{i=1}^n |x_i - y_i|^m \right]^{1/m}$$

5. 马氏距离

设 n 维向量 \vec{x}_i 和 \vec{x}_j 是向量集 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m\}$ 中的两个向量, 马氏距离 d 定义为

$$d^2(\vec{x}_i, \vec{x}_j) = (\vec{x}_i - \vec{x}_j)' V^{-1} (\vec{x}_i - \vec{x}_j)$$

其中

$$V = \frac{1}{m-1} \sum_{i=1}^m (\vec{x}_i - \bar{\vec{x}})(\vec{x}_i - \bar{\vec{x}})'$$
$$\bar{\vec{x}} = \frac{1}{m} \sum_{i=1}^m \vec{x}_i$$

注意! 马氏距离对一切非奇异线性变换都是不变的, 这说明它不受特征量纲选择的影响, 并且是平移不变的。

上面的 V 的含义是这个向量集的协方差阵的统计量, 故马氏距离加入了对特征的相关性的考虑。

6.2 模式相似性测度

一般地，设 \vec{x} 、 \vec{y} 是从期望矢量为 $\bar{\mu}$ 、协方差矩阵 Σ 的母体 G 中抽取的两个样本，它们间的马氏距离定义：

$$d^2(\vec{x}, \vec{y}) = (\vec{x} - \vec{y})' \Sigma^{-1} (\vec{x} - \vec{y})$$

当将 \vec{x} 和 \vec{y} 视作两个数据集中的样本时，设 C 是它们的互协方差阵，这种情况的马氏距离的定义是：

$$d^2(\vec{x}, \vec{y}) = (\vec{x} - \vec{y}) C^{-1} (\vec{x} - \vec{y})$$

例子

已知一个二维正态母体G的分布为 $N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}\right)$

求点 $A: \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ 和 $B: \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ 至均值点 $M: \bar{\mu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 的距离。

解：由题设，可得 $\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$ $\Sigma^{-1} = \frac{1}{0.19} \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}$

从而马氏距离

$$d_M^2(A, M) = (1 \ 1) \Sigma^{-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0.2 / 0.19 \quad d_M^2(B, M) = (1 \ -1) \Sigma^{-1} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = 3.8 / 0.19$$

它们之比达 $\sqrt{19}$ 倍，若用欧氏距离，算得的距离值相同。

$$d_E^2(A, M) = 2 \quad d_E^2(B, M) = 2$$

由分布函数知，A、B两点的概率密度分别为

$$p(1, 1) = 0.2157 \quad p(1, -1) = 0.00001658$$

6.2 模式相似性测度

二、相似测度

测度基础：以两矢量的方向是否相近作为考虑的基础, 矢量长度并不重要。设

$$\vec{x} = (x_1, x_2, \dots, x_n)', \vec{y} = (y_1, y_2, \dots, y_n)'$$

1. 角度相似系数(夹角余弦)

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x}'\vec{y}}{\|\vec{x}\|\|\vec{y}\|} = \frac{\vec{x}'\vec{y}}{[(\vec{x}'\vec{x})(\vec{y}'\vec{y})]^{1/2}}$$

注意：坐标系的旋转和尺度的缩放是不变的, 但对一般的线形变换和坐标系的平移不具有不变性。

6.2 模式相似性测度

二、相似测度

2. 相关系数

它实际上是数据中心化后的矢量夹角余弦。

$$r(\vec{x}, \vec{y}) = \frac{(\vec{x} - \bar{\vec{x}})'(\vec{y} - \bar{\vec{y}})}{\left[(\vec{x} - \bar{\vec{x}})'(\vec{x} - \bar{\vec{x}})(\vec{y} - \bar{\vec{y}})'(\vec{y} - \bar{\vec{y}})\right]^{1/2}}$$

3. 指数相似系数

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

式中 σ_i^2 为相应分量的协方差， n 为矢量维数。
它不受量纲变化的影响。

6.2 模式相似性测度

二、匹配测度

当特征只有两个状态（0，1）时，常用匹配测度。

0表示无此特征 1表示有此特征。故称之为**二值特征**。

对于给定的 x 和 y 中的某两个相应分量 x_i 与 y_j

若 $x_i=1, y_j=1$ ，则称 x_i 与 y_j 是 **(1-1) 匹配**；

若 $x_i=1, y_j=0$ ，则称 x_i 与 y_j 是 **(1-0) 匹配**；

若 $x_i=0, y_j=1$ ，则称 x_i 与 y_j 是 **(0-1) 匹配**；

若 $x_i=0, y_j=0$ ，则称 x_i 与 y_j 是 **(0-0) 匹配**。

6.2 模式相似性测度

三、匹配测度

对于二值 n 维特征矢量可定义如下相似性测度

令 $a = \sum_i x_i y_i$ 为 \vec{x} 与 \vec{y} 的 (1-1) 匹配的特征数目

$b = \sum_i y_i (1 - x_i)$ (0-1) 匹配的特征数目

$c = \sum_i x_i (1 - y_i)$ (1-0) 匹配的特征数目

$e = \sum_i (1 - x_i)(1 - y_i)$ (0-0) 匹配的特征数目

6.2 模式相似性测度

三、匹配测度

(1) Tanimoto测度

$$s(\vec{x}, \vec{y}) = \frac{a}{a + b + c} = \frac{\vec{x}'\vec{y}}{\vec{x}'\vec{x} + \vec{y}'\vec{y} - \vec{x}'\vec{y}}$$

○例4.2.2

设 $\vec{x} = (0, 1, 0, 1, 1, 0)'$ $\vec{y} = (0, 0, 1, 1, 0, 1)'$

则 $\vec{x}'\vec{x} = 3$, $\vec{y}'\vec{y} = 3$, $\vec{x}'\vec{y} = 1$

$$s(\vec{x}, \vec{y}) = \frac{1}{3 + 3 - 1} = \frac{1}{5}$$

可以看出，它等于共同具有的特征数目与分别具有的特征种类总数之比。这里只考虑(1-1)匹配而不考虑(0-0)匹配。

6.2 模式相似性测度

例6.2.2

设 $\vec{x} = (0, 1, 0, 1, 1, 0)'$ $\vec{y} = (0, 0, 1, 1, 0, 1)'$

则 $\vec{x}'\vec{x} = 3$, $\vec{y}'\vec{y} = 3$, $\vec{x}'\vec{y} = 1$

$$s(\vec{x}, \vec{y}) = \frac{1}{3 + 3 - 1} = \frac{1}{5}$$

可以看出，它等于共同具有的特征数目与分别具有的特征种类总数之比。这里只考虑(1-1)匹配而不考虑(0-0)匹配。

6.2 模式相似性测度

三、匹配测度

(2) Rao测度

$$s(\vec{x}, \vec{y}) = \frac{a}{a + b + c + e} = \frac{\vec{x}'\vec{y}}{n}$$

注：(1-1)匹配特征数目和所选用的特征数目之比。

(3) 简单匹配系数

$$m(\vec{x}, \vec{y}) = \frac{a + e}{n}$$

注：上式分子为(1-1)匹配特征数目与(0-0)匹配特征数目之和，分母为所考虑的特征数目。

6.3 类的定义与类间距离

类的定义

类的定义有很多种，类的划分具有人为规定性，这反映在定义的选取及参数的选择上。一个分类结果的优劣最后只能根据实际来评价。

定义之1 设集合S中任意元素 x_i 与 y_j 间的距离 d_{ij} 有

$$d_{ij} \leq h$$

其中h为给定的阈值，称S对于阈值h组成一类。

书中的其它定义方法请大家自行参考学习



6.3 类的定义与类间距离

类间距离测度方法

- (1) 最近距离法
- (2) 最远距离法
- (3) 中间距离法
- (4) 重心距离法
- (5) 平均距离法
- (6) 离差平方和法

6.3 类的定义与类间距离

类间距离测度方法

- (1) 最近距离法
- (2) 最远距离法
- (3) 中间距离法
- (4) 重心距离法
- (5) 平均距离法
- (6) 离差平方和法

$$D_{kl} = \min_{i,j} [d_{ij}]$$

式中 d_{ij} 表示 $\vec{x}_i \in \omega_k$
和 $\vec{x}_j \in \omega_l$ 之间的距离。

6.3 类的定义与类间距离

类间距离测度方法

- (1) 最近距离法
- (2) 最远距离法
- (3) 中间距离法
- (4) 重心距离法
- (5) 平均距离法
- (6) 离差平方和法

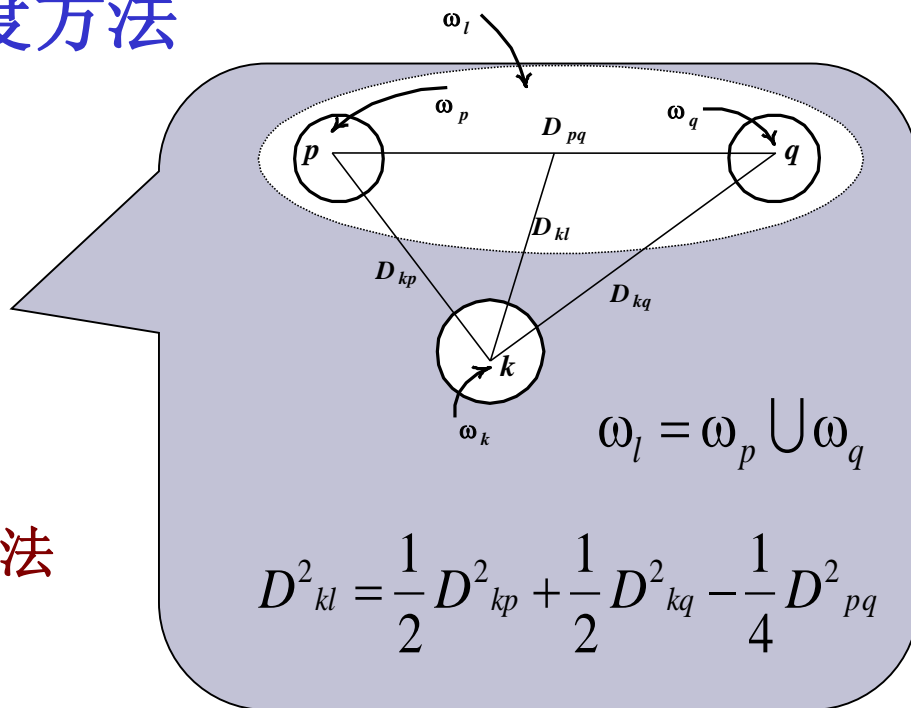
$$D_{kl} = \max_{i,j} [d_{ij}]$$

式中 d_{ij} 表示 $\vec{x}_i \in \omega_k$
和 $\vec{x}_j \in \omega_l$ 之间的距离。

6.3 类的定义与类间距离

类间距离测度方法

- (1) 最近距离法
- (2) 最远距离法
- (3) 中间距离法
- (4) 重心距离法
- (5) 平均距离法
- (6) 离差平方和法



6.3 类的定义与类间距离

类间距离测度方法

- (1) 最近距离法
- (2) 最远距离法
- (3) 中间距离法
- (4) 重心距离法
- (5) 平均距离法
- (6) 离差平方和法

$$D_{kl}^2 = \frac{n_p}{n_p + n_q} D_{kp}^2 + \frac{n_q}{n_p + n_q} D_{kq}^2 - \frac{n_p n_q}{(n_p + n_q)^2} D_{pq}^2$$

n_p, n_q 分别为类 ω_p 和 ω_q 的样本个数

6.3 类的定义与类间距离

类间距离测度方法

- (1) 最近距离法
- (2) 最远距离法
- (3) 中间距离法
- (4) 重心距离法
- (5) 平均距离法
- (6) 离差平方和法

$$D_{pq}^2 = \frac{1}{n_p n_q} \sum_{\substack{\vec{x}_i \in \omega_p \\ \vec{x}_j \in \omega_q}} d_{ij}^2$$

6.3 类的定义与类间距离

类间距离测度方法

- (1) 最近距离法
- (2) 最远距离法
- (3) 中间距离法
- (4) 重心距离法
- (5) 平均距离法
- (6) 离差平方和法

类内离差平方和

$$s_t = \sum_{\vec{x}_i \in \omega_t} (\vec{x}_i - \vec{x}_t)' (\vec{x}_i - \vec{x}_t)$$

$$\omega_l = \omega_p \cup \omega_q \quad D_{pq}^2 = s_l - s_p - s_q$$

$$D_{pq}^2 = \frac{n_p n_q}{n_p + n_q} (\vec{x}_p - \vec{x}_q)' (\vec{x}_p - \vec{x}_q)$$

\vec{x}_t \vec{x}_p \vec{x}_q 分别为对应类的重心

递推公式为:

$$D_{kl}^2 = \frac{n_k + n_p}{n_k + n_l} D_{kp}^2 + \frac{n_k + n_q}{n_k + n_l} D_{kq}^2 - \frac{n_k}{n_k + n_l} D_{pq}^2$$

$$D_{kl}^2 = \alpha_p D_{kp}^2 + \alpha_q D_{kq}^2 + \beta D_{pq}^2 + \gamma |D_{kp}^2 - D_{kq}^2|$$

	α_p	α_q	β	γ
最近距离法	1/2	1/2	0	-1/2
最远距离法	1/2	1/2	0	1/2
中间距离法	1/2	1/2 <1	-1/4	0
重心距离法	$\frac{n_p}{n_p + n_q}$	$\frac{n_q}{n_p + n_q}$	$-\alpha_p \alpha_q$	0
平均距离法	$\frac{n_p}{n_p + n_q}$	$\frac{n_q}{n_p + n_q}$	0	0
可变平均法	$(1-\beta) \frac{n_p}{n_p + n_q}$	$(1-\beta) \frac{n_q}{n_p + n_q}$	<1	0
可变法	$\frac{1-\beta}{2}$	$\frac{1-\beta}{2}$		0
离差平方和法	$\frac{n_k + n_p}{n_k + n_l}$	$\frac{n_k + n_q}{n_k + n_l}$	$-\frac{n_k}{n_k + n_l}$	0



6.3 类的定义与类间距离

聚类的准则函数

判别分类结果好坏的一般标准：

类内距离小，类间距离大。

某些算法需要一个能对分类过程或分类结果的优劣进行评估的准则函数。如果聚类准则函数选择得好，聚类质量就会高。聚类准则往往是和类的定义有关的，是类的定义的某种体现。

6.3 类的定义与类间距离

聚类的准则函数

一、类内距离准则

设有待分类的模式集 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ 在某种相似性测度基础上被划分为 c 类, $\{\vec{x}_i^{(j)}; j=1, 2, \dots, c; i=1, 2, \dots, n_j\}$ 类内距离准则函数 J_W 定义为: (\vec{m}_j 表示 ω_j 类的模式均值矢量。)

$$J_W = \sum_{j=1}^c \sum_{i=1}^{n_j} \left\| \vec{x}_i^{(j)} - \vec{m}_j \right\|^2$$



6.3 类的定义与类间距离

$$J_W = \sum_{j=1}^c \sum_{i=1}^{n_j} \left\| \vec{x}_i^{(j)} - \vec{m}_j \right\|^2$$

我们的目标是使 J_W 取最小, 即 $J_W \Rightarrow \min$, 这种方法也称为误差平方和准则。

显然, J_W 是模式 \vec{x}_i 和类心 \vec{m}_j 的函数, 在样本集 $\{\vec{x}_i\}$ 给定条件下, J_W 的值取决于类心的选取。



加权类内距离准则 J_{WW} :

$$J_{WW} = \sum_{j=1}^c \frac{n_j}{N} \bar{d}_j^2$$
$$\bar{d}_j^2 = \frac{2}{n_j(n_j-1)} \sum_{\substack{\vec{x}_k^{(j)} \in \omega_j \\ \vec{x}_i^{(j)} \in \omega_j}} \left\| \vec{x}_i^{(j)} - \vec{x}_k^{(j)} \right\|^2$$

式中, $\sum \left\| \vec{x}_i^{(j)} - \vec{x}_k^{(j)} \right\|^2$ 表示 ω_j 类内任两个模式距离平方和, 共有 $\frac{n_j(n_j-1)}{2}$ 个组合数, 所以 \bar{d}_j^2 表示类内两模式间的均方距离。 N 为待分类模式总数, $\frac{n_j}{N}$ 表示 ω_j 类先验概率的估计——频率。

6.3 类的定义与类间距离

二、类间距离准则

$$J_B = \sum_{j=1}^c (\vec{m}_j - \vec{m})' (\vec{m}_j - \vec{m}) \Rightarrow \max$$

这里, \vec{m}_j 为 ω_j 类的模式平均矢量, \vec{m} 为总的模式平均矢量。设 n_j 为 ω_j 类所含模式个数,

$$\vec{m}_j = \frac{1}{n_j} \sum_{\vec{x}_i \in \omega_j} \vec{x}_i \quad \vec{m} = \frac{1}{N} \sum_{i=1}^N \vec{x}_i$$



加权类间距离准则:

$$J_{WB} = \sum_{j=1}^c \frac{n_j}{N} (\vec{m}_j - \vec{m})' (\vec{m}_j - \vec{m}) \Rightarrow \max$$

对于两类问题，类间距离有时取

$$J_{B2} = (\vec{m}_1 - \vec{m}_2)' (\vec{m}_1 - \vec{m}_2)$$

J_{B2} 和 J_{WB} 的关系是

$$J_{WB} = \frac{n_1}{N} \frac{n_2}{N} J_{B2}$$

6.3 类的定义与类间距离

三、基于类内距离类间距离的准则函数

我们希望聚类结果使类内距离越小越好，类间距离越大越好。为此构造能同时反映出类内距离和类间距离的准则函数。

设待分类模式集 $\{\vec{x}_i, i = 1, 2, \dots, N\}$, 将它们分成 c 类, ω_j 类含 n_j 个模式, 分类后各模式记为

$$\{\vec{x}_i^{(j)}, j = 1, 2, \dots, c; i = 1, 2, \dots, n_j\}$$

6.3 类的定义与类间距离

ω_j 的类内离差阵定义为

$$S_W^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} (\vec{x}_i^{(j)} - \vec{m}_j)(\vec{x}_i^{(j)} - \vec{m}_j)' \quad (j=1,2,\cdots,c)$$

式中 \vec{m}_j 为类 ω_j 的模式均值矢量

$$\vec{m}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \vec{x}_i^{(j)} \quad (j=1,2,\cdots,c)$$

总的类内离差阵定义为
$$S_W = \sum_{j=1}^c \frac{n_j}{N} S_W^{(j)}$$

类间离差阵定义为
$$S_B = \sum_{j=1}^c \frac{n_j}{N} (\vec{m}_j - \vec{m})(\vec{m}_j - \vec{m})'$$

式中, \vec{m} 为所有待分
类模式的均值矢量:

$$\vec{m} = \frac{1}{N} \sum_{i=1}^N \vec{x}_i$$

总的离差阵 S_T , 定义为
$$S_T = \frac{1}{N} \sum_{i=1}^N (\vec{x}_i - \vec{m})(\vec{x}_i - \vec{m})'$$

于是有下面关系
$$S_T = S_W + S_B$$

6.3 类的定义与类间距离

例6.3.1 证明： $S_T = S_W + S_B$

$$\begin{aligned} S_T &= \frac{1}{N} \sum_{i=1}^N (\vec{x}_i - \vec{m})(\vec{x}_i - \vec{m})' = \frac{1}{N} \sum_{j=1}^c \sum_{i=1}^{n_j} (\vec{x}_i^{(j)} - \vec{m})(\vec{x}_i^{(j)} - \vec{m})' \\ S_T &= \sum_{j=1}^c \frac{n_j}{N} \frac{1}{n_j} \sum_{i=1}^{n_j} (\vec{x}_i^{(j)} - \vec{m})(\vec{x}_i^{(j)} - \vec{m})' \\ &= \sum_{j=1}^c \frac{n_j}{N} \left[\frac{1}{n_j} \sum_{i=1}^{n_j} (\vec{x}_i^{(j)} - \vec{m}_j)(\vec{x}_i^{(j)} - \vec{m}_j)' + (\vec{m}_j - \vec{m})(\vec{m}_j - \vec{m})' \right] \\ &= S_W + S_B \end{aligned}$$

6.3 类的定义与类间距离

聚类的基本目的是使 $\text{Tr}[S_B] \Rightarrow \max$
或 $\text{Tr}[S_W] \Rightarrow \min$ 。利用线形代数有关矩阵的迹和行列式的性质,可以定义如下4个聚类的准则函数:

$$J_1 = \text{Tr}[S_W^{-1} S_B] \quad J_2 = |S_W^{-1} S_B|$$

$$J_3 = \text{Tr}[S_W^{-1} S_T] \quad J_4 = |S_W^{-1} S_T|$$

由它们的构造可以看出,为得到好的聚类结果,应该使它们尽量的大。这类准则也大量用在特征提取和选择中。

LDA参考网址: <http://blog.csdn.net/yihaizhiyan/article/details/7579506>



6.4 聚类的算法准则

聚类准则：根据相似性测度确定的，衡量模式之间是否相似的标准。即把不同模式聚为一类还是归为不同类的准则。

确定聚类准则的两种方式：

1. 阈值准则：根据规定的距离阈值进行分类的准则。
2. 函数准则：利用聚类准则函数进行分类的准则。

误差平方和函数：
$$J = \sum_{j=1}^c \sum_{X \in S_j} \|X - M_j\|^2$$

式中： c 为聚类类别的数目，

$$M_j = \frac{1}{N_j} \sum_{X \in S_j} X \text{ 为属于 } S_j \text{ 集的样本的均值向量,}$$

N_j 为 S_j 中样本数目。



6.4 聚类的算法

聚类的技术方案

聚类分析有很多具体的算法,有的比较简单,有的相对复杂和完善,但归纳起来就是三大类:

- 1、按最小距离原则简单聚类方法
- 2、按最小距离原则进行两类合并的方法
- 3、依据准则函数动态聚类方法

(1) 简单聚类方法

针对具体问题确定相似性阈值,将模式到各聚类中心间的距离与阈值比较,当大于阈值时该模式就作为另一类的类心,小于阈值时按最小距离原则将其分划到某一类中。

这类算法运行中模式的类别及类的中心一旦确定将不会改变。



6.4 聚类的算法

(2) 按最小距离原则进行两类合并的方法

首先视各模式自成一类,然后将距离最小的两类合并成一类,不断地重复这个过程,直到成为两类为止。

这类算法运行中,类心不断地修正,但模式类别一旦指定后就不再改变,就是模式一旦划为一类后就不再被分划开,这类算法也称为谱系聚类法。

(3) 依据准则函数动态聚类法

设定一些分类的控制参数,定义一个能表征聚类结果优劣的准则函数,聚类过程就是使准则函数取极值的优化过程。

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。

6.4 聚类的算法——简单聚类方法

根据相似性阈值和最小距离原则的简单聚类方法

1. 条件及约定

设待分类的模式为 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ ，选定类内距离门限 T 。

2. 算法思想

计算模式特征矢量到聚类中心的距离并和门限 T 比较，决定归属该类或作为新的一类中心。这种算法通常选择欧氏距离。

6.4 聚类的算法——简单聚类方法

3. 算法原理步骤

(1) 取任意的一个模式特征矢量作为第一个聚类中心。例如，令 ω_1 类的中心 $\vec{z}_1 = \vec{x}_1$ 。

(2) 计算下一个模式特征矢量 \vec{x}_2 到 \vec{z}_1 的距离 d_{21} 。若 $d_{21} > T$ ，则建立新的一类 ω_2 ，其中心 $\vec{z}_2 = \vec{x}_2$ 。若 $d_{21} \leq T$ ，则 $\vec{x}_2 \in \omega_1$ 。

6.4 聚类的算法——简单聚类方法

3. 算法原理步骤

(3) 假设已有聚类中心 $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_k$ ，计算尚未确定类别的模式特征矢量 \vec{x}_i 到各聚类中心 \vec{z}_j ($j = 1, 2, \dots, k$) 的距离 d_{ij} 。如果 $d_{ij} > T$ ($j = 1, 2, \dots, k$)，则 \vec{x}_i 作为新的一类 ω_{k+1} 的中心， $\vec{z}_{k+1} = \vec{x}_i$ ；
否则，如果 $d_{il} = \min_j [d_{ij}]$ ，则指判 $\vec{x}_i \in \omega_l$ 。检查是否所有的模式都分划完类别，如果都分划完了则结束；否则返到(3)。



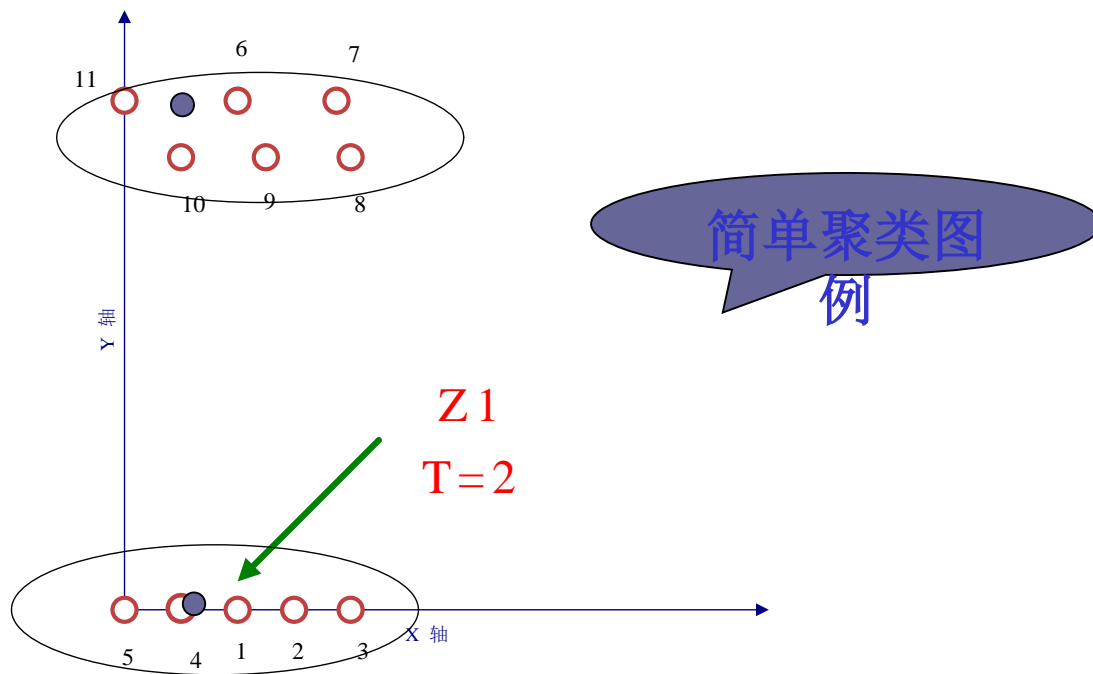
6.4 聚类的算法——简单聚类方法

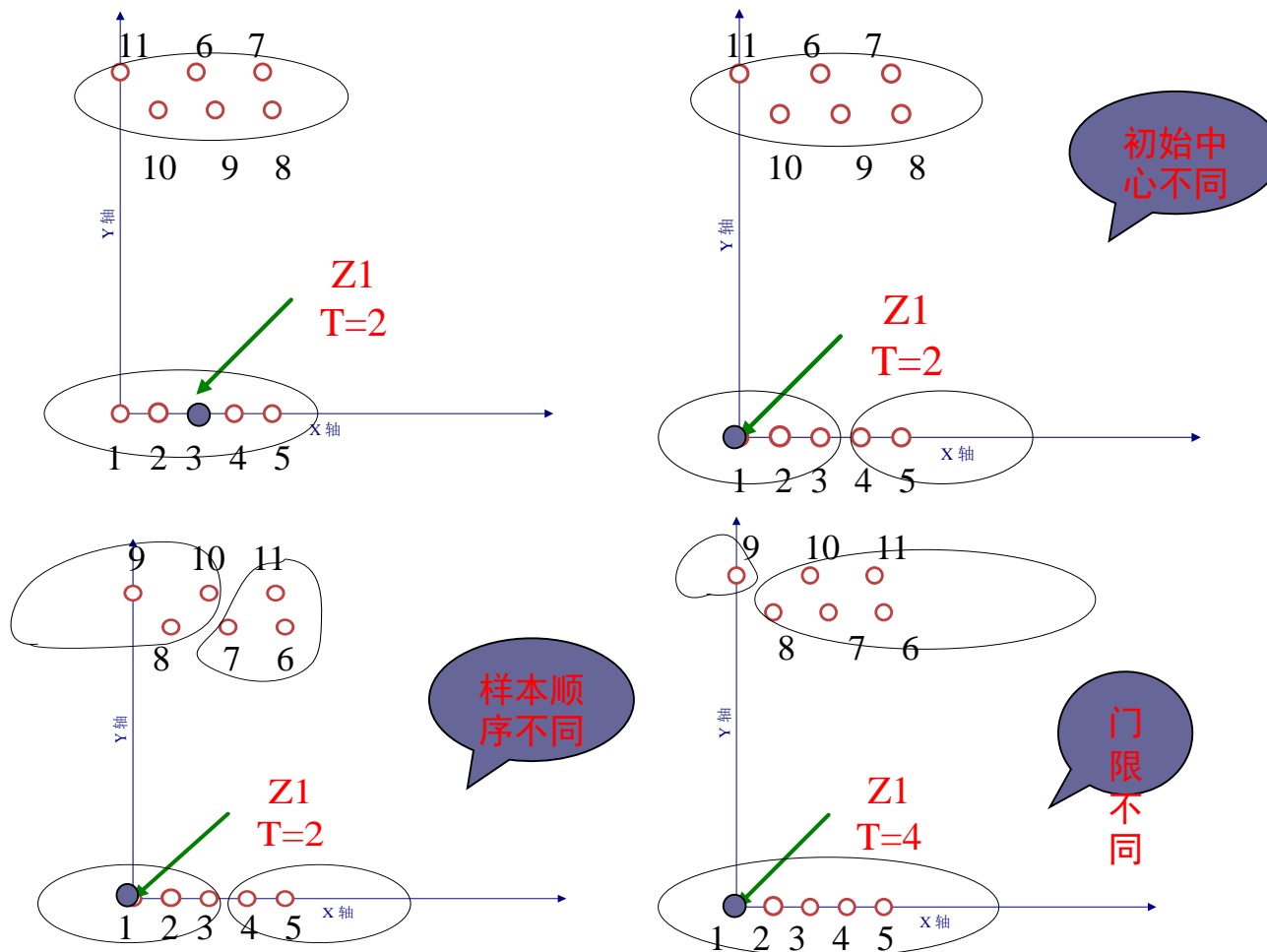
算法特点:

这类算法的突出优点是算法简单。但聚类过程中，类的中心一旦确定将不会改变，模式一旦指定类后也不再改变。

从算法的过程可以看出，该算法结果很大程度上依赖于距离门限 T 的选取及模式参与分类的次序。如果能有先验知识指导门限 T 的选取，通常可获得较合理的效果。也可考虑设置不同的 T 和选择不同的次序，最后选择较好的结果进行比较。

6.4 聚类的算法——简单聚类方法





例：初始条件不同的简单聚类结果

6.4 聚类的算法——最大最小距离法

1. 条件及约定

设待分类的模式为 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$, 选定比例系数 θ 。

2. 算法思想

在模式特征矢量集中以最大距离原则选取新的聚类中心。以最小距离原则进行模式归类。这种方法通常也使用欧氏距离。

6.4 聚类的算法——最大最小距离法

3. 算法原理步骤

(1) 选任一模式特征矢量作为第一个聚类中心 \vec{z}_1

例如, $\vec{z}_1 = \vec{x}_1$ 。

(2) 从待分类矢量集中选距离 \vec{z}_1 最远的特征矢量
作为第二个聚类中心 \vec{z}_2 。

6.4 聚类的算法——最大最小距离法

- (3) 计算未被作为聚类中心的各模式特征矢量 $\{\vec{x}_i\}$ 与 \vec{z}_1 、 \vec{z}_2 之间的距离，并求出它们之中的最小值，即

$$d_{ij} = \|\vec{x}_i - \vec{z}_j\| \quad (j = 1, 2)$$

$$d_i = \min[d_{i1}, d_{i2}] \quad (i = 1, 2, \dots, N)$$

为表述简洁，虽然某些模式已选做聚类中心，但上面仍将所有模式下角标全部列写出来，因这并不影响算法的正确性。

6.4 聚类的算法——最大最小距离法

(4) 若

$$d_l = \max_i [\min(d_{i1}, d_{i2})] > \theta \|\vec{z}_1 - \vec{z}_2\|$$

则相应的特征矢量 \vec{x}_l 作为第三个聚类中心, $\vec{z}_3 = \vec{x}_l$

然后转至(5); 否则, 转至最后一步(6)。

(5) 设存在 k 个聚类中心, 计算未被作为聚类中心的各特征矢量到各聚类中心的距离 d_{ij} , 并算出

$$d_l = \max_i [\min[d_{i1}, d_{i2}, \dots, d_{ik}]]$$

如果 $d_l > \theta \|\vec{z}_1 - \vec{z}_2\|$, 则 $\vec{z}_{k+1} = \vec{x}_l$ 并转至(5);

否则, 转至最后一步(6)。

6.4 聚类的算法——最大最小距离法

- (6) 当判断出不再有新的聚类中心之后，将模式特征矢量 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ 按最小距离原则分到各类中去，即计算

$$d_{ij} = \|\vec{x}_i - \vec{z}_j\| \quad (j=1,2,\dots; i=1,2,\dots,N)$$

当 $d_{il} = \min_j [d_{ij}]$ ，则判 $\vec{x}_i \in \omega_l$ 。

这种算法的聚类结果与参数 θ 以及第一个聚类中心的选取有关。如果没有先验知识指导 θ 和 \vec{z}_1 的选取，可适当调整 θ 和 \vec{z}_1 ，比较多次试探分类结果，选取最合理的一种聚类。

6.4 聚类的算法 谱系聚类法

层次聚类法 (Hierarchical Clustering Method)(系统聚类法、谱系聚类法)

按最小距离原则不断进行两类合并

1. 条件及约定

设待分类的模式特征矢量为 $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$, $G_i^{(k)}$ 表示第 k 次合并时的第 i 类。

2. 算法思想

首先将 N 个模式视作各自成为一类，然后计算类与类之间的距离，选择距离最小的一对合并成一个新类，计算在新的类别分划下各类之间的距离，再将距离最近的两类合并，直至所有模式聚成两类为止。

6.4 聚类的算法 谱系聚类法步骤

(1) 初始分类。令 $k=0$ ，每个模式自成一类，
即

$$G_i^{(0)} = \{\vec{x}_i\} \quad (i=1,2,\dots,N)$$

(2) 计算各类间的距离 D_{ij} ，由此生成一个对称的距离矩阵 $D^{(k)} = (D_{ij})_{m \times m}$ ， m 为类的个数（初始时 $m = N$ ）。

(3) 找出前一步求得的矩阵 $D^{(k)}$ 中的最小元素，
设它是 $G_i^{(k)}$ 和 $G_j^{(k)}$ 间的距离，将 $G_i^{(k)}$ 和 $G_j^{(k)}$
两类合并成一类，于是产生新的聚类 $G_1^{(k+1)}, G_2^{(k+1)}, \dots$
令 $k = k + 1, m = m - 1$

(4) 检查类的个数。如果类数 m 大于2，转至(2)；
否则，停止。

例：给出6个五维模式样本如下，按最短距离准则进行系统聚类分类。

$$\mathbf{X}_1 = [0, 3, 1, 2, 0]^T \quad \mathbf{X}_2 = [1, 3, 0, 1, 0]^T \quad \mathbf{X}_3 = [3, 3, 0, 0, 1]^T$$

$$\mathbf{X}_4 = [1, 1, 0, 2, 0]^T \quad \mathbf{X}_5 = [3, 2, 1, 2, 1]^T \quad \mathbf{X}_6 = [4, 1, 1, 1, 0]^T$$

解：（1）将每一样本看作单独一类，得：

$$G_1(0) = \{\mathbf{X}_1\} \quad G_2(0) = \{\mathbf{X}_2\} \quad G_3(0) = \{\mathbf{X}_3\}$$

$$G_4(0) = \{\mathbf{X}_4\} \quad G_5(0) = \{\mathbf{X}_5\} \quad G_6(0) = \{\mathbf{X}_6\}$$

计算各类间欧氏距离：

$$\begin{aligned} D_{12}(0) &= \|\mathbf{X}_1 - \mathbf{X}_2\| = \left[(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + (x_{13} - x_{23})^2 + (x_{14} - x_{24})^2 + (x_{15} - x_{25})^2 \right]^{1/2} \\ &= [1 + 0 + 1 + 1 + 0]^{1/2} = \sqrt{3} \end{aligned}$$

$$\begin{aligned} D_{13}(0) &= [3^2 + 0 + 1 + 2^2 + 1]^{1/2} = \sqrt{15} \quad , \quad D_{14}(0), \quad D_{15}(0), \quad D_{16}(0), \\ D_{23}(0) \quad D_{24}(0) \quad D_{25}(0) \quad D_{26}(0) \quad ; \quad & D_{34}(0) \quad D_{35}(0) \quad D_{36}(0) \quad \dots \end{aligned}$$

得距离矩阵 $\mathbf{D}(0)$:

$\mathbf{D}(0)$	$G_1(0)$	$G_2(0)$	$G_3(0)$	$G_4(0)$	$G_5(0)$	$G_6(0)$
$G_1(0)$	0					
$G_2(0)$	$\sqrt{3}$	0				
$G_3(0)$	$\sqrt{15}$	$\sqrt{6}$	0			
$G_4(0)$	$\sqrt{6}$	$\sqrt{5}$	$\sqrt{13}$	0		
$G_5(0)$	$\sqrt{11}$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{7}$	0	
$G_6(0)$	$\sqrt{21}$	$\sqrt{14}$	$\sqrt{8}$	$\sqrt{11}$	$\sqrt{4}$	0

(2) 将最小距离 $\sqrt{3}$ 对应的类 $G_1(0)$ 和 $G_2(0)$ 合并为1类, 得新的分类。

$$G_{12}(1) = \{G_1(0), G_2(0)\}$$

$$G_3(1) = \{G_3(0)\} \quad G_4(1) = \{G_4(0)\}$$

$$G_5(1) = \{G_5(0)\} \quad G_6(1) = \{G_6(0)\}$$

计算聚类后的距离矩阵 $\mathbf{D}(1)$:

由 $\mathbf{D}(0)$ 递推出 $\mathbf{D}(1)$ 。

$D(0)$	$G_1(0)$	$G_2(0)$	$G_3(0)$	$G_4(0)$	$G_5(0)$	$G_6(0)$
$G_1(0)$	0					
$G_2(0)$	$\sqrt{3}$	0				
$G_3(0)$	<u>$\sqrt{15}$</u>	<u>$\sqrt{6}$</u>	0			
$G_4(0)$	<u>$\sqrt{6}$</u>	<u>$\sqrt{5}$</u>	$\sqrt{13}$	0		
$G_5(0)$	<u>$\sqrt{11}$</u>	<u>$\sqrt{8}$</u>	$\sqrt{6}$	$\sqrt{7}$	0	
$G_6(0)$	<u>$\sqrt{21}$</u>	<u>$\sqrt{14}$</u>	$\sqrt{8}$	$\sqrt{11}$	$\sqrt{4}$	0

$D(1)$	$G_{12}(1)$	$G_3(1)$	$G_4(1)$	$G_5(1)$	$G_6(1)$
$G_{12}(1)$	0				
$G_3(1)$	$\sqrt{6}$	0			
$G_4(1)$	$\sqrt{5}$	$\sqrt{13}$	0		
$G_5(1)$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{7}$	0	
$G_6(1)$	$\sqrt{14}$	$\sqrt{8}$	$\sqrt{11}$	$\sqrt{4}$	0

$D(2)$	$G_{12}(2)$	$G_3(2)$	$G_4(2)$	$G_{56}(2)$
$G_{12}(2)$	0			
$G_3(2)$	$\sqrt{6}$	0		
$G_4(2)$	$\sqrt{5}$	$\sqrt{13}$	0	
$G_{56}(2)$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{7}$	0

(3) 将 $D(1)$ 中最小值 $\sqrt{4}$ 对应的类合为一类，得 $D(2)$ 。

(4) 将 $D(2)$ 中最小值 $\sqrt{5}$ 对应的类合为一类，得 $D(3)$ 。

$D(2)$	$G_{12}(2)$	$G_3(2)$	$G_4(2)$	$G_{56}(2)$
$G_{12}(2)$	0			
$G_3(2)$	<u>$\sqrt{6}$</u>	0	$\sqrt{13}$	
$G_4(2)$	* $\sqrt{5}$	$\sqrt{13}$	0	
$G_{56}(2)$	<u>$\sqrt{8}$</u>	$\sqrt{6}$	<u>$\sqrt{7}$</u>	0

$D(3)$	$G_{124}(3)$	$G_3(3)$	$G_{56}(3)$
$G_{124}(3)$	0		
$G_3(3)$	$\sqrt{6}$	0	
$G_{56}(3)$	$\sqrt{7}$	$\sqrt{6}$	0

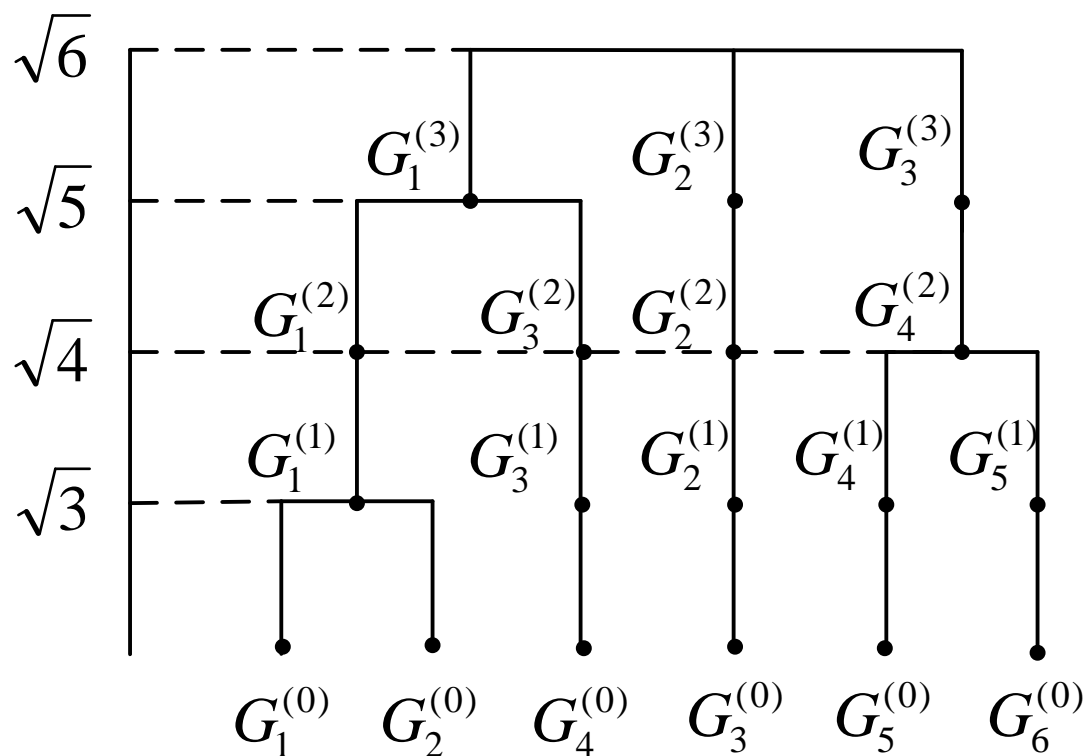
若给定的阈值为 $T = \sqrt{5}$ ， $D(3)$ 中的最小元素 $\sqrt{6} > T$ ，聚类结束。

$$G_1 = \{X_1, X_2, X_4\} \quad G_2 = \{X_3\} \quad G_3 = \{X_5, X_6\}$$

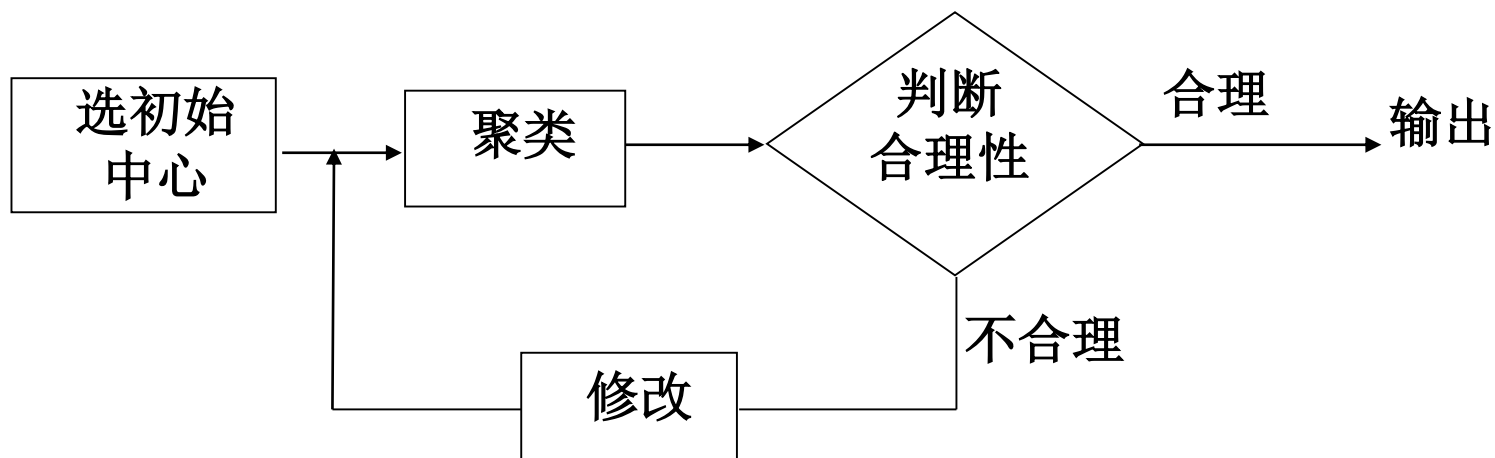
若无阈值，继续分下去，最终全部样本归为一类。可给出聚类过程的树状表示图。

系统聚类的树状表示

- 根据不同的距离阈值，可以确定不同的聚类。



6.5 聚类的算法 动态聚类法步骤



两种常用算法:

- * K-均值算法(或C-均值算法)
- * 迭代自组织的数据分析算法(ISODATA, iterative self-organizing data analysis techniques algorithm)



K—均值算法

- 条件及约定:

- 设待分类的模式特征矢量集为 $\{x_1, x_2, \dots, x_N\}$;
- 类的数目 K 是事先取定的。

- 基本思想:

- 首先任意选取 K 个聚类中心，按最小距离原则将各模式分配到 K 类的某一类;
- 不断计算聚类中心和调整各模式的类别，最终使各模式到其判属类别中心的距离平方之和最小。

基于使聚类准则函数最小化，

准则函数：聚类集中每一样本点到该类中心的距离平方和。

对于第 j 个聚类集，准则函数定义为

$$J_j = \sum_{i=1}^{N_j} \| \mathbf{X}_i - \mathbf{Z}_j \|^2, \quad \mathbf{X}_i \in S_j$$

S_j : 第 j 个聚类集（域），聚类中心为 \mathbf{Z}_j ；

N_j : 第 j 个聚类集 S_j 中所包含的样本个数。

对所有 K 个模式类有

$$J = \sum_{j=1}^K \sum_{i=1}^{N_j} \| \mathbf{X}_i - \mathbf{Z}_j \|^2, \quad \mathbf{X}_i \in S_j$$

K-均值算法的聚类准则：聚类中心的选择应使准则函数 J 极小，
即使 J_j 的值极小。



应有 $\frac{\partial J_j}{\partial \mathbf{Z}_j} = 0$

即
$$\frac{\partial}{\partial \mathbf{Z}_j} \sum_{i=1}^{N_j} \|\mathbf{X}_i - \mathbf{Z}_j\|^2 = \frac{\partial}{\partial \mathbf{Z}_j} \sum_{i=1}^{N_j} (\mathbf{X}_i - \mathbf{Z}_j)^T (\mathbf{X}_i - \mathbf{Z}_j) = 0$$

可解得
$$\mathbf{Z}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{X}_i, \quad \mathbf{X}_i \in S_j$$

上式表明， S_j 类的聚类中心应选为该类型样本的均值。

(1) 任选 K 个模式特征矢量作为初始聚类中心:

$z_1(1), z_2(1), \dots, z_K(1)$ 。括号内的序号表示迭代次数。

(2) 将待分类的模式特征矢量集 $\{x\}$ 中的模式逐个按最小距离原则分划给 K 类中的某一类。

如果 $D_j(k) = \min\{\|x - z_i(k)\|\}$, $i=1, 2, \dots, K$

则判 $x \in S_j(k)$

(3) 计算重新分类后的各聚类中心 $z_j(k+1)$,
即求各聚类域中所包含样本的均值向量:

$$z_j(k+1) = \frac{1}{N_j} \sum_{x \in S_j(k)} x, \quad j=1, 2, \dots, K$$

以均值向量作新的聚类中心, 可得新的准则函数:

$$J_j = \sum_{x \in S_j(k)} \|x - z_j(k+1)\|^2, \quad j=1, 2, \dots, K$$

(4) 如果 $z_j(k+1) = z_j(k)$ ($j=1, 2, \dots, K$), 则结束; 否则, $k=k+1$, 转(2)



“动态”聚类法

?

聚类过程中，
聚类中心位置或个数发生变化。

2. 算法讨论

结果受到所选聚类中心的个数和其初始位置，以及模式样本的几何性质及读入次序等的影响。实际应用中需要试探不同的K值和选择不同的聚类中心起始值。

例：已知20个模式样本如下，试用K-均值算法分类。

$$\mathbf{X}_1 = [0,0]^T \quad \mathbf{X}_2 = [1,0]^T \quad \mathbf{X}_3 = [0,1]^T \quad \mathbf{X}_4 = [1,1]^T$$

$$\mathbf{X}_5 = [2,1]^T \quad \mathbf{X}_6 = [1,2]^T \quad \mathbf{X}_7 = [2,2]^T \quad \mathbf{X}_8 = [3,2]^T$$

$$\mathbf{X}_9 = [6,6]^T \quad \mathbf{X}_{10} = [7,6]^T \quad \mathbf{X}_{11} = [8,6]^T \quad \mathbf{X}_{12} = [6,7]^T$$

$$\mathbf{X}_{13} = [7,7]^T \quad \mathbf{X}_{14} = [8,7]^T \quad \mathbf{X}_{15} = [9,7]^T \quad \mathbf{X}_{16} = [7,8]^T$$


$$\mathbf{X}_{17} = [8,8]^T \quad \mathbf{X}_{18} = [9,8]^T \quad \mathbf{X}_{19} = [8,9]^T \quad \mathbf{X}_{20} = [9,9]^T$$

解：① 取 $K=2$ ，并选： $\mathbf{Z}_1(1) = \mathbf{X}_1 = [0,0]^T$ $\mathbf{Z}_2(1) = \mathbf{X}_2 = [1,0]^T$

② 计算距离，聚类：

$$\mathbf{X}_1: \left. \begin{array}{l} D_1 = \|\mathbf{X}_1 - \mathbf{Z}_1(1)\| = 0 \\ D_2 = \|\mathbf{X}_1 - \mathbf{Z}_2(1)\| = \sqrt{(0-1)^2 + (0-0)^2} = \sqrt{1} \end{array} \right\} \Rightarrow D_1 < D_2 \Rightarrow \mathbf{X}_1 \in S_1(1)$$

$$\mathbf{X}_2: \left. \begin{array}{l} D_1 = \|\mathbf{X}_2 - \mathbf{Z}_1(1)\| = \sqrt{1} \\ D_2 = \|\mathbf{X}_2 - \mathbf{Z}_2(1)\| = 0 \end{array} \right\} \Rightarrow D_2 < D_1 \Rightarrow \mathbf{X}_2 \in S_2(1)$$



$$X_3: \left. \begin{array}{l} D_1 = \|X_3 - Z_1(1)\| = \sqrt{(0-0)^2 + (1-0)^2} = \sqrt{1} \\ D_2 = \|X_3 - Z_2(1)\| = \sqrt{(0-1)^2 + (1-0)^2} = \sqrt{2} \end{array} \right\} \Rightarrow D_1 < D_2 \Rightarrow X_3 \in S_1(1)$$

$$X_4: \left. \begin{array}{l} D_1 = \|X_4 - Z_1(1)\| = \sqrt{(1-0)^2 + (1-0)^2} = \sqrt{2} \\ D_2 = \|X_4 - Z_2(1)\| = \sqrt{(1-1)^2 + (1-0)^2} = \sqrt{1} \end{array} \right\} \Rightarrow D_2 < D_1 \Rightarrow X_4 \in S_2(1)$$

....., 可得到: $S_1(1) = \{X_1, X_3\} \quad N_1 = 2$

③ 计算新的聚类中: $S_2(1) = \{X_2, X_4, X_5, \dots, X_{20}\} \quad N_2 = 18$

$$Z_1(2) = \frac{1}{N_1} \sum_{X \in S_1(1)} X = \frac{1}{2} (X_1 + X_3) = \frac{1}{2} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$

$$Z_2(2) = \frac{1}{N_2} \sum_{X \in S_2(1)} X = \frac{1}{18} (X_2 + X_4 + \dots + X_{20}) = \begin{bmatrix} 5.67 \\ 5.33 \end{bmatrix}$$

④ 判断: $\because Z_j(2) \neq Z_j(1) \quad j=1,2$, 故返回第②步。

② 从新的聚类中心得:

$$X_1: \left. \begin{array}{l} D_1 = \|X_1 - Z_1(2)\| = \dots \\ D_2 = \|X_1 - Z_2(2)\| = \dots \end{array} \right\} \Rightarrow X_1 \in S_1(2)$$

⋮

$$X_{20}: \left. \begin{array}{l} D_1 = \|X_{20} - Z_1(2)\| = \dots \\ D_2 = \|X_{20} - Z_2(2)\| = \dots \end{array} \right\} \Rightarrow X_{20} \in S_2(2)$$

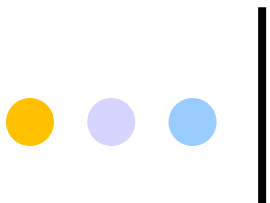
$$\text{有: } S_1(2) = \{X_1, X_2, \dots, X_8\} \quad N_1 = 8$$

$$S_2(2) = \{X_9, X_{10}, \dots, X_{20}\} \quad N_2 = 12$$

③ 计算聚类中心:

$$Z_1(3) = \frac{1}{N_1} \sum_{X \in S_1(2)} X = \frac{1}{8} (X_1 + X_2 + \dots + X_8) = \begin{bmatrix} 1.25 \\ 1.13 \end{bmatrix}$$

$$Z_2(3) = \frac{1}{N_2} \sum_{X \in S_2(2)} X = \frac{1}{12} (X_9 + X_{10} + \dots + X_{20}) = \begin{bmatrix} 7.67 \\ 7.33 \end{bmatrix}$$



④ $\because \mathbf{Z}_j(3) \neq \mathbf{Z}_j(2) \quad j=1,2$

返回第②步，以 $\mathbf{z}_1(3)$ ， $\mathbf{z}_2(3)$ 为中心进行聚类。

② 以新的聚类中心分类，求得的分类结果与前一次迭代结果相同：
 $S_1(3) = S_1(2) \quad S_2(3) = S_2(2)$

③ 计算新聚类中心向量值，聚类中心与前一次结果相同，即：

$$\mathbf{Z}_j(4) = \mathbf{Z}_j(3), \quad j=1,2$$

④ $\because \mathbf{Z}_j(4) = \mathbf{Z}_j(3)$ ，故算法收敛，得聚类中心为

$$\mathbf{Z}_1 = \begin{bmatrix} 1.25 \\ 1.13 \end{bmatrix}, \quad \mathbf{Z}_2 = \begin{bmatrix} 7.67 \\ 7.33 \end{bmatrix}$$

结果图示：

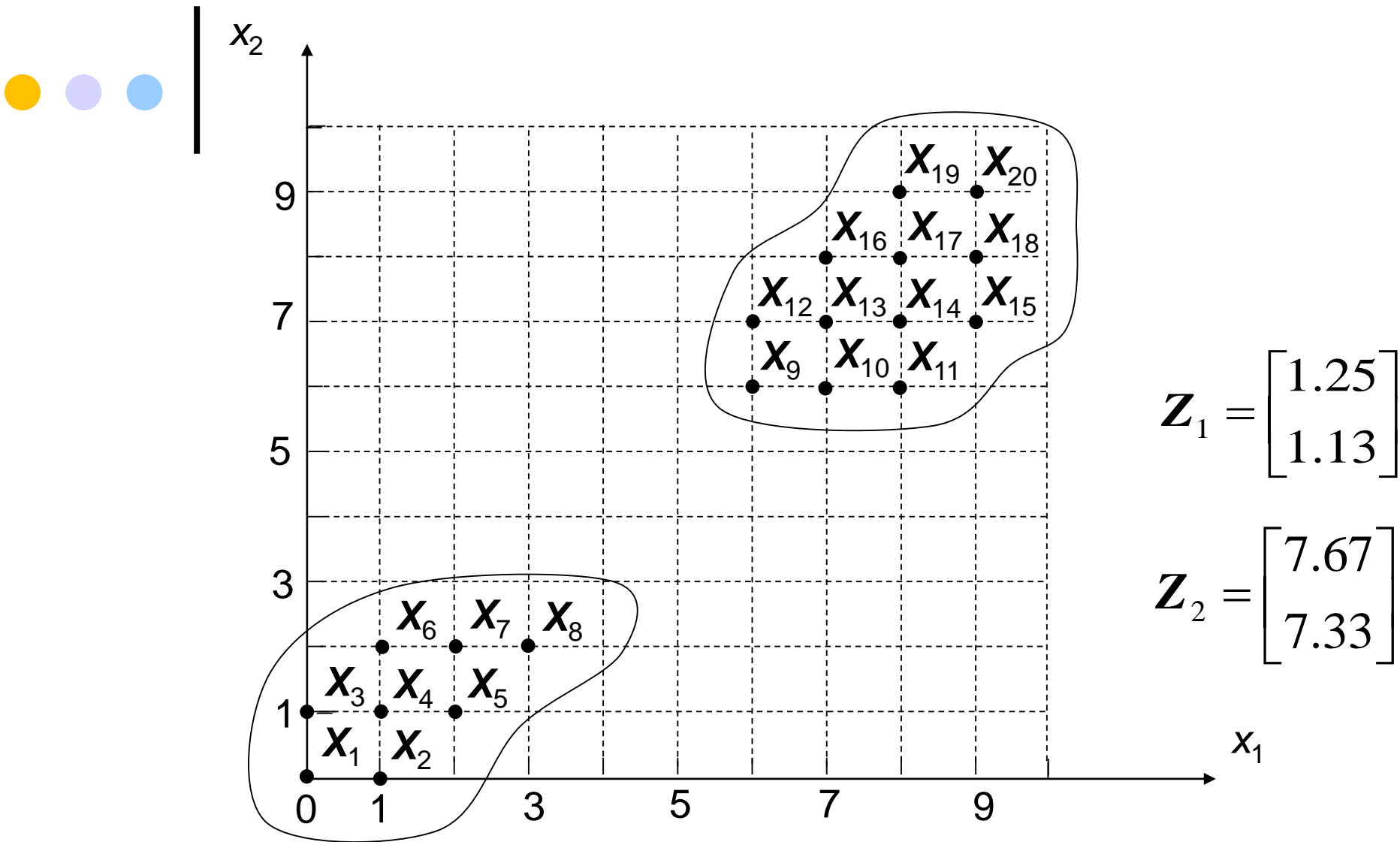
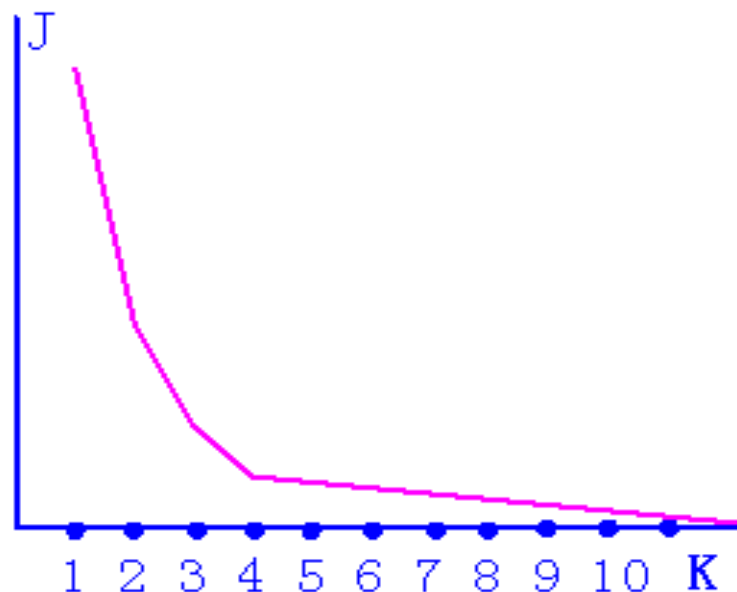


图2.10 K-均值算法聚类结果

类别数目未知情况下如何使用？

- 在类别数未知情况下使用K—均值算法时，可以假设类别数是逐步增加的。显然准则函数是随K的增加而单调地减少的。
- 如果样本集的合理聚类数为K类，当类别数从1增加到K时准则函数迅速减小，当类别数超过K时，准则函数虽然继续减少但会呈现平缓趋势。





如何避免初始聚类中心的影响？

- 多次运行K均值算法，例如**50~1000**次，每次随机选取不同的初始聚类中心。
- 聚类结束后计算准则函数值。
- 选取准则函数值最小的聚类结果为最后的结果。
- 该方法一般适用于聚类数目小于**10**的情况。

ISODATA算法的提出

(iterative self-organizing data analysis techniques algorithm, ISODATA)

- **K—均值**算法比较简单，但它的自我调整能力也比较差。这主要表现在类别数不能改变，受代表点初始选择的影响也比较大。
- **ISODATA**算法的功能与**K—均值**算法相比，在下列几方面有改进：
 - 可以改变类别数目。通过类别的合并与分裂来实现。
 - 合并主要发生在某一类内样本个数太少的情况，或两类聚类中心之间距离太小的情况。为此设有**最小类内样本数限制**，以及**类间中心距离参数**。
 - 分裂则主要发生在某一类别的某分量出现类内方差过大的现象，因而宜分裂成两个类别，以维持合理的类内方差。给出一个对**类内分量方差的限制参数**，用以决定是否需要将某一类分裂成两类。
 - 由于算法有自我调整的能力，因而需要设置若干个控制用参数。，如聚类数期望值**K**、每次迭代允许合并的最大聚类对数**L**、及允许迭代次数**I**等。

○ 基本步骤和思路

- (1) 选择初始控制参数。可选不同的指标，也可在迭代过程中人为修改，以将N个模式样本按指标分配到各个聚类中心中去。
- (2) 计算各类中诸样本的距离指标函数。
- (3) ~ (5) 按给定的要求，将前一次获得的聚类集进行分裂和合并处理（(4)为分裂处理，(5)为合并处理），从而获得新的聚类中心。
- (6) 重新进行迭代运算，计算各项指标，判断聚类结果是否符合要求，如不符合，返回(2)。经过多次迭代后，若结果收敛，则运算结束。

ISODATA算法的步骤

○ 步骤1(确定控制参数及设置代表点)

需确定的控制参数为：

K： 聚类期望数；

θ_N ： 一个聚类中的最少样本数；

θ_C ： 类间距离控制参数，用于控制合并；

θ_S ： 标准偏差控制参数，用于控制分裂；

L： 每次迭代允许合并的最大聚类对数；

I： 允许迭代的次数。

设定初始聚类数为 N_c ，任意选定初始的聚类中心 z_i ， $i = 1, 2, \dots, N_c$ 。

ISODATA算法的步骤(续)

○ 步骤2(分类)

对所有样本，按给定的 N_c 个聚类中心，以最小距离进行分类，即

若 $D_j = \min(\|x - z_i\|)$, $i = 1, 2, \dots, N_c$, 则 $x \in S_j$

○ 步骤3(撤消类内样本数过小类别)

若有任何一个类 S_j ，其样本数 $N_j < \theta$ ，则舍去 S_j ，令 $N_c = N_c - 1$ ，将原样本分配至其它类；

ISODATA算法的步骤(续)

步骤4(更新均值向量)

按现有样本分类结果，调整均值参数

$$z_j = \frac{1}{N_j} \sum_{x \in S_j} x, \quad i = 1, 2, \dots, N_c$$

步骤5(计算类内平均距离)

每类各样本到均值的平均距离

$$\tilde{D}_j = \frac{1}{N_j} \sum_{x \in S_j} \|x - z_j\|, \quad j = 1, 2, \dots, N_c$$

步骤6(计算全部样本集到相应均值的平均距离)

$$\tilde{D} = \frac{1}{N} \sum_{j=1}^{N_c} N_j \tilde{D}_j$$

ISODATA算法的步骤(续)

步骤7(入口选择, 判断分裂、合并及迭代运算)

如这是最后一次迭代(取决于 l), 则转步骤11, 并设置 $\theta_c = 0$, 防止合并发生。

如果 $N_c \leq K/2$, 则转向步骤8, 执行分裂步骤;

如果此时迭代运算次数是偶数次, 或 $N_c \geq 2K$, 则转向步骤11, 执行合并步骤, 否则继续执行, 进行分裂。

步骤8(分裂步骤1: 求各类内各分类标准偏差)

对每个聚类 j , 求其标准偏差 $\sigma_j = (\sigma_{j1} \quad \sigma_{j2} \quad \cdots \quad \sigma_{jn})^t$

$$\sigma_{ji} = \sqrt{\frac{1}{N_j} \sum_{y_k \in S_j} (y_{ki} - z_{ji})^2}$$

式中, σ_{ji} 是第 j 个聚类第 i 个分量的标准偏差, y_{ki} 是第 j 类中第 k 个样本的第 i 分量, z_{ji} 是均值向量 z_j 的第 i 分量, n 是样本特征维数。



l : 允许迭代的次数。

ϑ_c : 两聚类中心之间的最小距离。

N_c : 预选的聚类中心数。

K : 希望的聚类中心的数目。

ISODATA算法的步骤(续)

步骤9(分裂步骤2: 求每类具有最大标准偏差的分量)

求出每类具有最大标准偏差的分量 $\sigma_{j_{\max}}, j = 1, 2, \dots, c$ 。

步骤10(分裂步骤3: 执行分裂)

若任一个 $\sigma_{j_{\max}}, j = 1, 2, \dots, c$ 有 $\sigma_{j_{\max}} > \theta_s$,

并且有(a) $\tilde{D}_j > \tilde{D}$ 且 $N_j > 2\theta_N + 1$ 或有(b) $N_c \leq K/2$,

则把 S_j 分裂成两个聚类, 其中心相应为 z_j^+ 与 z_j^- ,

把原来的 S_j 取消, 且令 $N_c = N_c + 1$ 。转第2步。

ISODATA算法的步骤(续)

- 步骤11(合并步骤1: 计算类间聚类中心距离)
i类与j类的类间距离

$$D_{ij} = \|z_i - z_j\|, \quad i = 1, 2, \dots, N_c - 1, \quad j = i + 1, i + 2, \dots, N_c$$

- 步骤12(合并步骤2: 列出类间距离过近者)
比较 D_{ij} 与 θ_c 并将小于 θ_c 的按上升次序排列

$$D_{i_1 j_1} < D_{i_2 j_2} < \dots < D_{i_l j_l}, \quad l \leq L$$

该队列最大个数是控制合并对数的参数L。

ISODATA算法的步骤(续)

○ 步骤13(合并步骤3: 执行合并)

从类间距离最小的两类开始执行合并过程, 此时需将 z_{kl} 与 z_{jl} 合并, 得

$$z_l = \frac{1}{N_{il} + N_{jl}} [N_{il} z_{il} + N_{jl} z_{jl}]$$

且 $N_c = N_c - 1$ 。

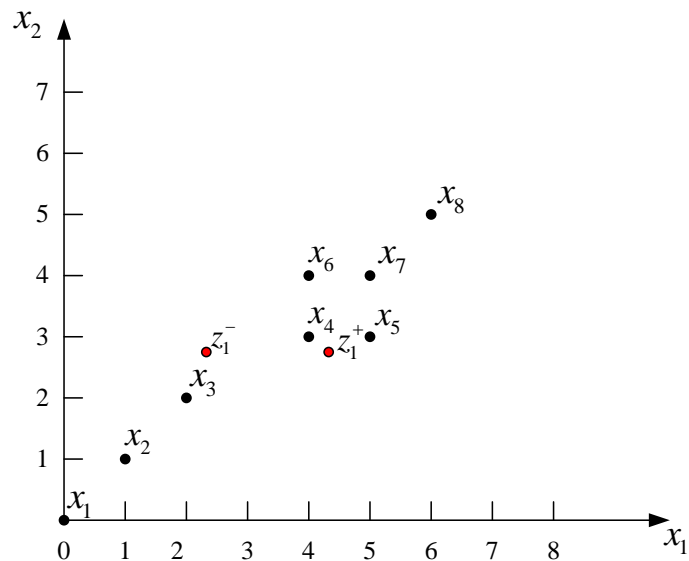
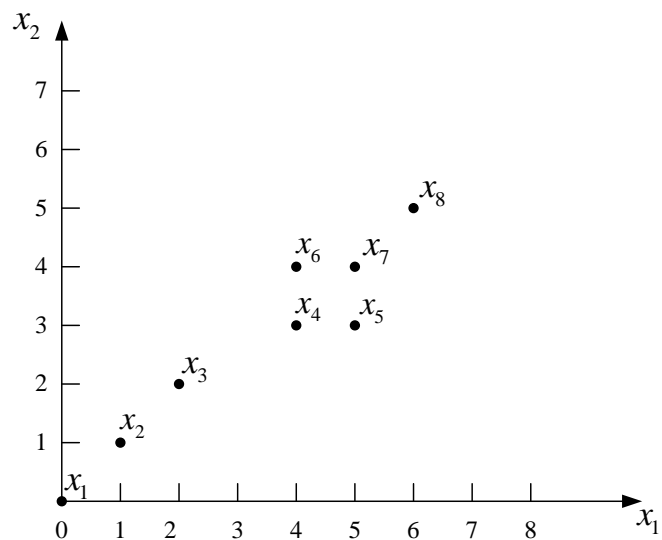
○ 步骤14(结束步骤)

如是最后一次迭代则终止, 否则可根据需要转步骤1或步骤2, 转步骤1是为了更改控制数。迭代计数要加1。

分裂步骤中样本中心的确定

- 由于 \mathbf{z}_j^+ 与 \mathbf{z}_j^- 值设置不当将会导致影响到其它类别，因此 \mathbf{z}_j^+ 与 \mathbf{z}_j^- 可按以下步骤计算：
 - (a) 给定一 k 值， $0 < k < 1$ ；
 - (b) 令 $r_j = k\sigma_{j\max}$ ；
 - (c) $\mathbf{z}_j^+ = \mathbf{z}_j + r_j$ ， $\mathbf{z}_j^- = \mathbf{z}_j - r_j$ ；其中 k 值应使 \mathbf{S}_j 中的样本到 \mathbf{z}_j^+ 与 \mathbf{z}_j^- 的距离不同，但又应使 \mathbf{S}_j 中的样本仍然在分裂后的新样本类中。

ISODATA算法举例

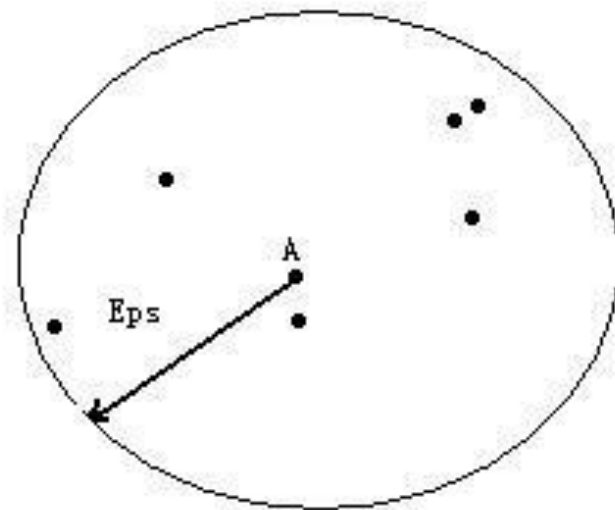
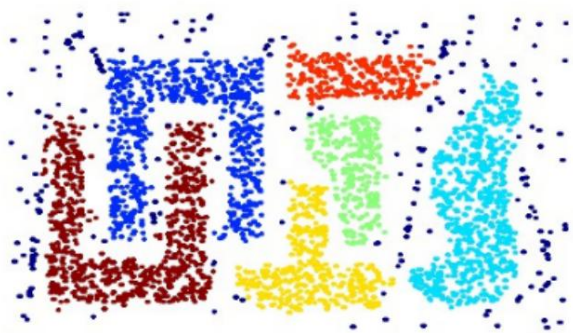


第一次迭代中分裂后得到的聚类中心

基于密度的聚类算法（DBSCAN算法）

传统基于中心的密度定义为：

- 数据集中特定点的密度通过该点Eps半径之内的点计数(包括本身)来估计。
- 显然，密度依赖于半径。

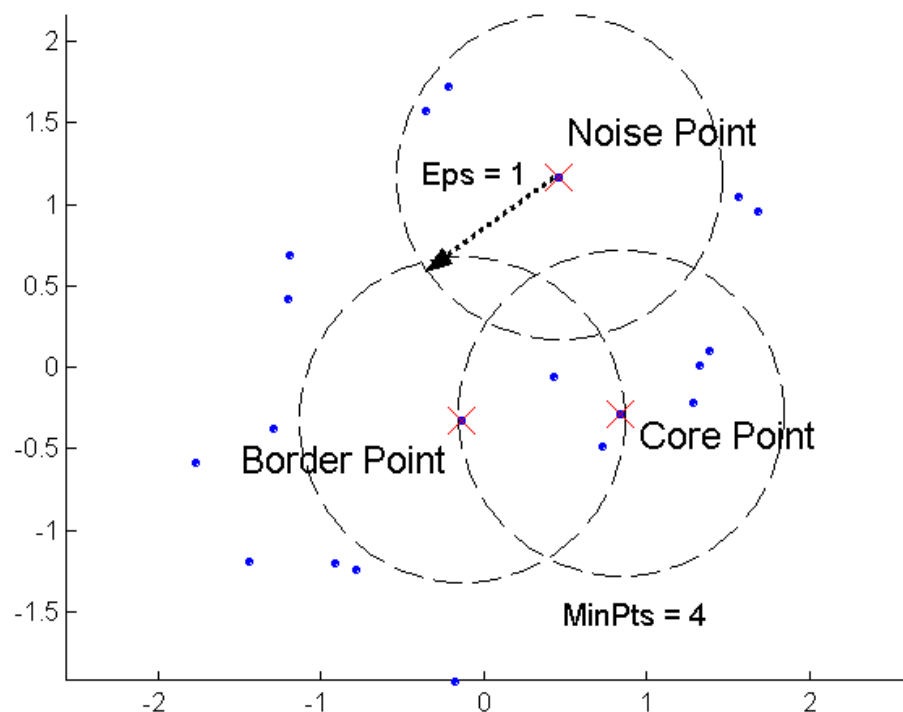




DBSCAN

- 基于密度定义，我们将点分为：
 - 稠密区域内部的点(核心点)
 - 稠密区域边缘上的点(边界点)
 - 稀疏区域中的点(噪声或背景点).
- 核心点(core point) :在半径Eps内含有超过MinPts数目的点，则该点为核心点
 - 这些点都是在簇内的
- 边界点(border point):在半径Eps内点的数量小于MinPts，但是在核心点的邻居
- 噪音点(noise point):任何不是核心点或边界点的点.

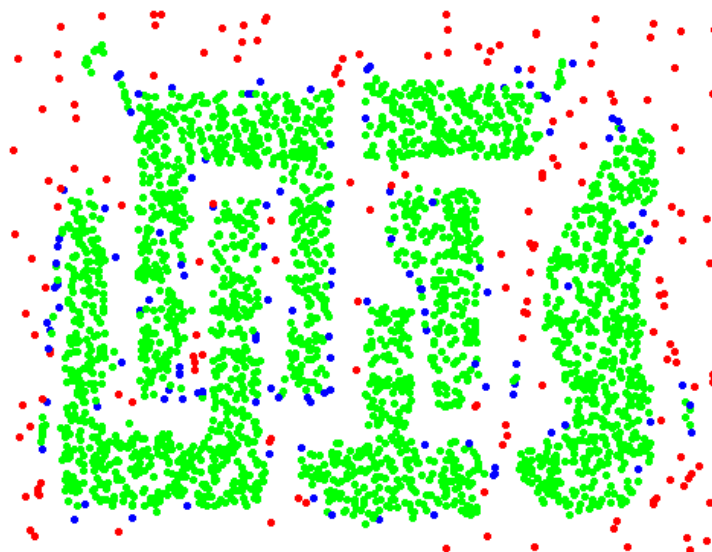
DBSCAN: 核心点、边界点和噪音点



DBSCAN: 核心点、边界点和噪音点



Original Points



Point types: **core**,
border and **noise**

Eps = 10, MinPts = 4

DBSCAN算法概念

Eps邻域: 给定对象半径**Eps**内的邻域称为该对象的**Eps**邻域, 我们用 $N_{Eps}(p)$ 表示点**p**的**Eps**-半径内的点的集合, 即:

$$N_{Eps}(p) = \{q \mid q \text{ 在数据集 } D \text{ 中, } \text{distance}(p, q) \leq Eps\}$$

核心对象: 如果对象的**Eps**邻域至少包含最小数目**MinPts**的对象, 则称该对象为核心对象。

边界点: 边界点不是核心点, 但落在某个核心点的邻域内。

噪音点: 既不是核心点, 也不是边界点的任何点

DBSCAN算法概念

直接密度可达： 给定一个对象集合D，如果p在q的Eps邻域内，而q是一个核心对象，则称对象p从对象q出发时是直接密度可达的(directly density-reachable)。

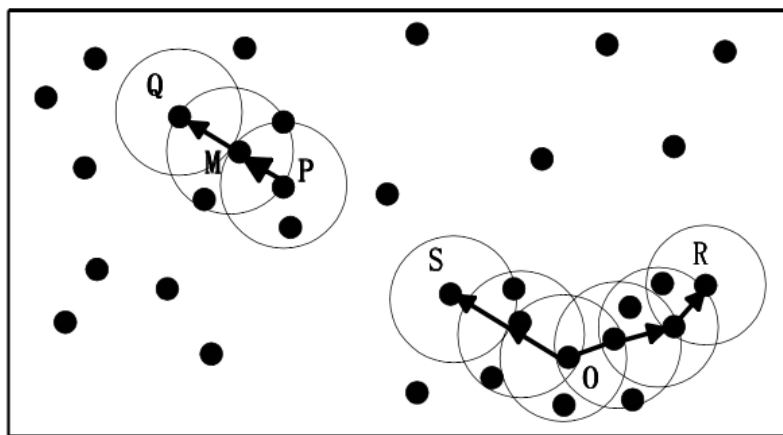
密度可达： 如果存在一个对象链 $p_1, p_2, \dots, p_n, p_1 = q, p_n = p$ 对于 $p_i \in D (1 \leq i \leq n)$

， p_{i+1} 是从 p_i 关于Eps和MinPts直接密度可达的，则对象p是从对象q关于Eps和MinPts密度可达的(density-reachable)。

密度相连： 如果存在对象 $O \in D$ ，使对象p和q都是从O关于Eps和MinPts密度可达的，那么对象p到q是关于Eps和MinPts密度相连的(density-connected)。

DBSCAN算法概念示例

- 如图所示， Eps 用一个相应的半径表示，设 $MinPts=3$ ，请分析Q,M,P,S,O,R这5个样本点之间的关系。



“直接密度可达”和“密度可达”概念示意描述

解答：根据以上概念知道：由于有标记的各点M、P、O和R的 Eps 近邻均包含3个以上的点，因此它们都是核对象；M是从P“直接密度可达”；而Q则是从M“直接密度可达”；基于上述结果，Q是从P“密度可达”；但P从Q无法“密度可达”（非对称）。类似地，S和R从O是“密度可达”的；O、R和S均是“密度相连”的。



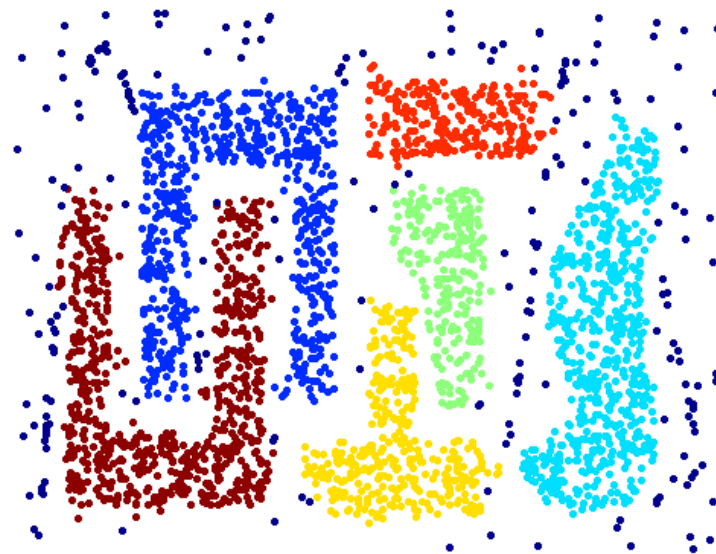
DBSCAN算法原理

- **DBSCAN**通过检查数据集中每点的**Eps**邻域来搜索簇，如果点**p**的**Eps**邻域包含的点多于**MinPts**个，则创建一个以**p**为核心对象的簇。
- 然后，**DBSCAN**迭代地聚集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。
- 当没有新的点添加到任何簇时，该过程结束。
- 优点
 - 基于密度定义，相对抗噪音，能处理任意形状和大小的簇
- 缺点
 - 当簇的密度变化太大时，会有麻烦
 - 对于高维问题，密度定义是个比较麻烦的问题

DBSCAN运行效果



Original Points



Clusters

- 对噪音不敏感
- 可以处理不同形状和大小的数据

聚类结果的评价

- 迅速评价聚类结果，在上述迭代运算中是很重要的，特别是具有高维特征向量的模式，不能直接看清聚类效果，因此，可考虑用以下几个指标来评价聚类效果：
 - 聚类中心之间的距离
 - 距离值大，通常可考虑分为不同类
 - 聚类域中的样本数目
 - 样本数目少且聚类中心距离远，可考虑是否为噪声
 - 聚类域内样本的距离方差
 - 方差过大的样本可考虑是否属于这一类
- 讨论：模式聚类目前还没有一种通用的放之四海而皆准的准则，往往需要根据实际应用来选择合适的方法。



课后作业

- 第四章习题 3, 6, 7, 9, 12
- 编程题：自己实现一种简单改进聚类算法，编程语言不限