数据分析技术

第5章 聚类方法

聚类分析

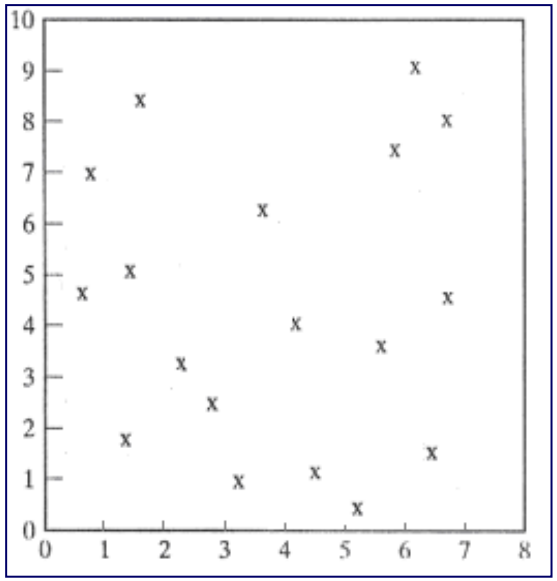
授课对象：计算机专业学生

- 物以类聚，人以群分
- 将对象进行自动分组的方法
- 聚类：clustering

编号	账户余额	年龄（岁）	收入（元）	性别	子女个数
100	很低	15	1967	男	0
200	高	25	8453	男	1
300	中	32	6125	女	2
400	低	20	2167	男	1
500	低	55	2439	女	4

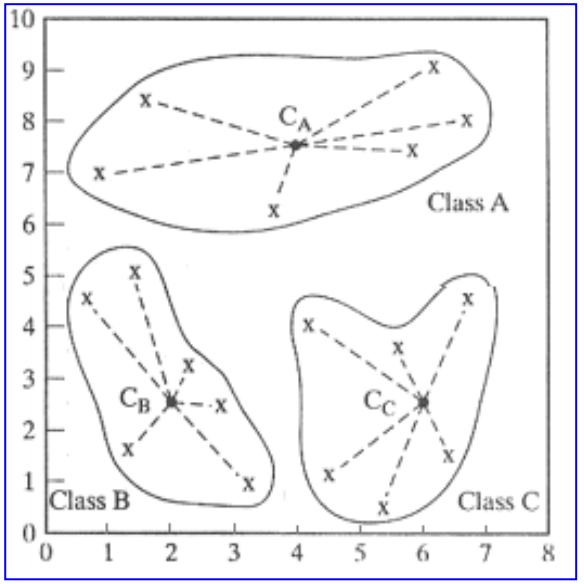
- 收集客户的个人资料、消费行为等多方面的数据；
- 利用聚类技术实现客户的自动分群；

分类与聚类



（a）待分类样例

（b）类定义



（c）分类结果

X ∈ （0，8）
Y ∈ （0，10）

欧式距离：待分点
中心点

K近邻算法

二. 聚类方法的应用

应用：聚类分析可以作为一个独立的工具来获得数据的分布况

聚类方法概述

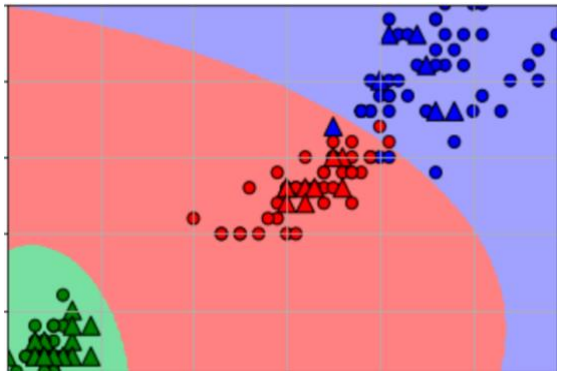
聚类的基本概念

- 在没有训练的条件下，把样本划分为若干类
- 无类别标签的样本
- 无监督学习

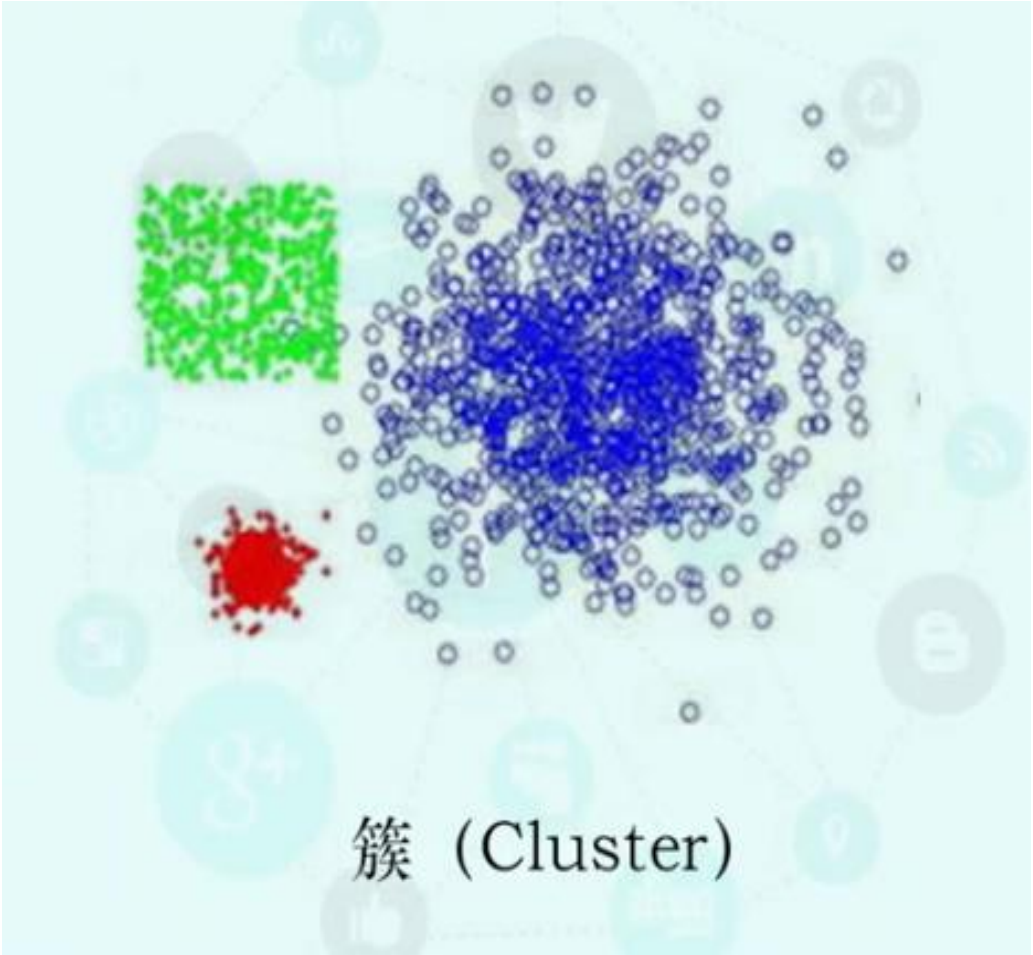
分类的基本概念

- 已知存在哪些类，将每一条记录分别属于哪一类标记出来。
- 有类别标签的样本
- 有监督学习

区别



- 聚类分析：是指根据给定一组对象的描述信息，发现由具有共同特性的对象构成簇的过程。



聚类的数学定义

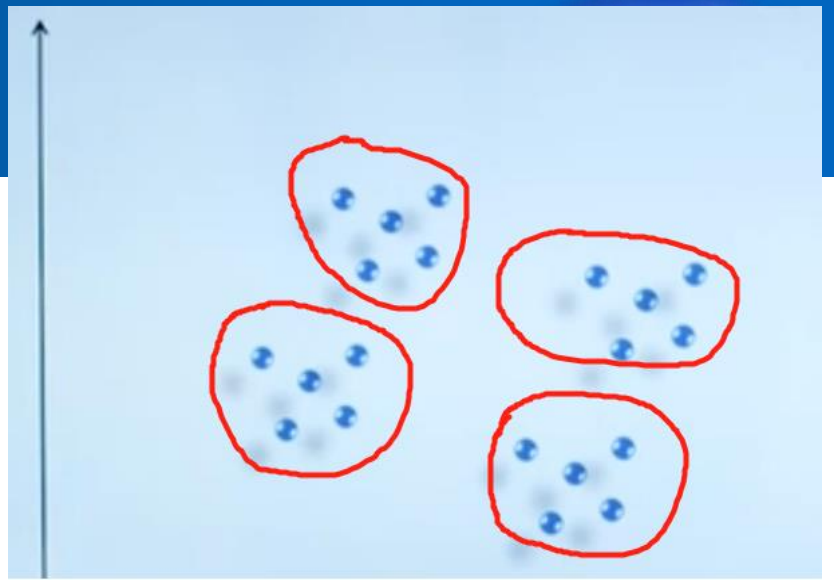
假设数据集 D 由 n 个对象的构成：即 $D=\{o_1, o_2, \cdots, o_n\}$，

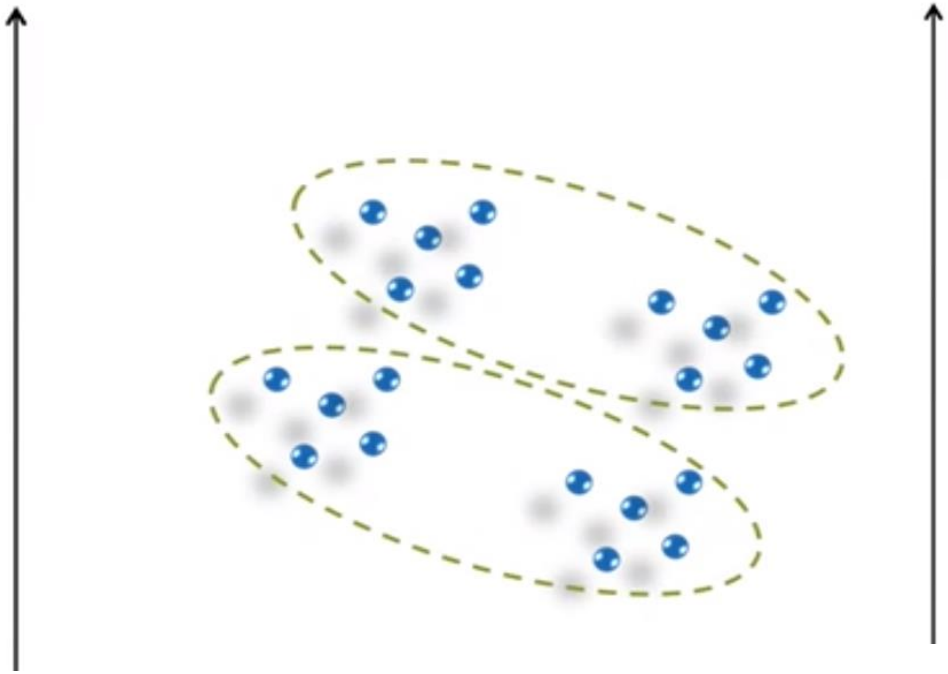
其中，每个对象o_i由 m 个属性描述，即$o_i=\{x_{i1}, x_{i2}, \cdots, x_{im}\}$，

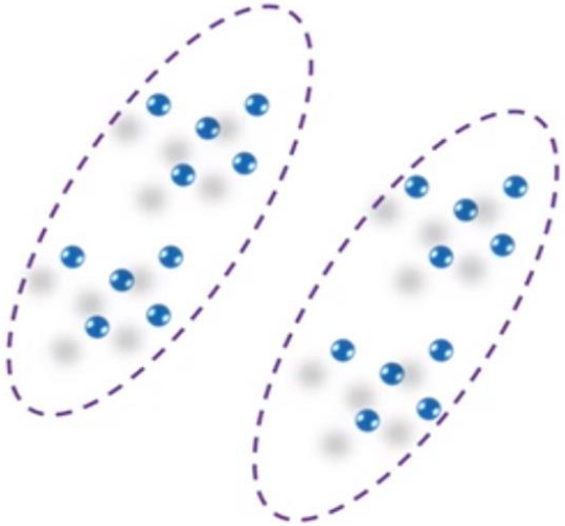
属性集合 $A=\{A_1, A_2, \cdots, A_m\}$，$x_{ij}$是第 i 个对象第 j 个属性的取值

聚类分析，就是给定数据集 D，对其中所有的对象进行自动的分组，从而得到出 k 个簇。设簇的集合为 $C=\{C_1, C_2, \cdots, C_k\}$，

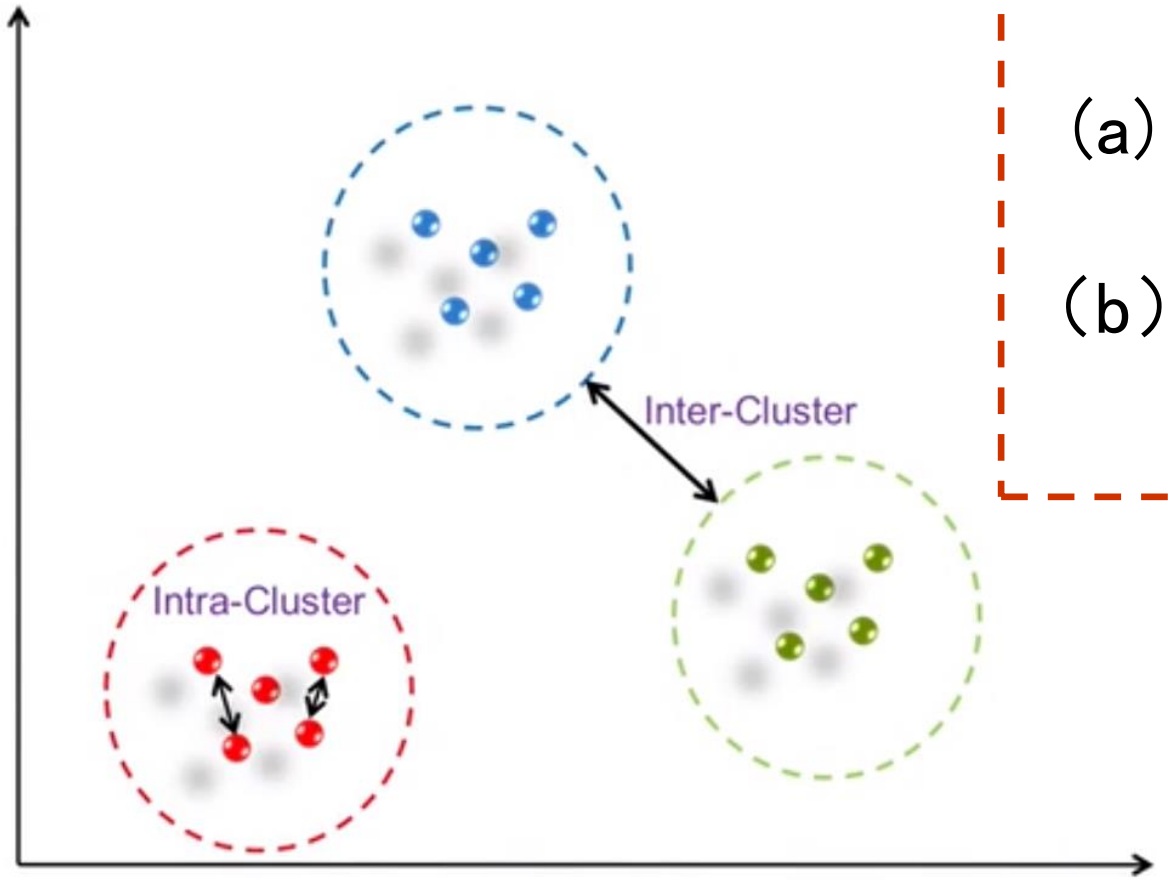
其中每一个$C_i=\{o_{i_1}, o_{i_2}, \cdots, o_{i_l}\} \subseteq D$







聚类方法概述



如何评判聚类的好坏：

（a） 类间距离：大

（b）类内距离：小

基于距离的相似度

两个对象o_i和o_j的欧式距离(Euclidean Distance)$d(o_i, o_j)$的方法如式(1)所示：

$$d(o_i, o_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{im} - x_{jm})^2}$$

曼哈顿距离(Manhattan Distance) $d(o_i, o_j)$计算方

$$d(o_i, o_j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{im} - x_{jm}|$$

基于距离的相似度

明可夫斯基距离(Minkowski Distance)是欧式距离和曼哈顿距离的概括，如式(3)所示：

$$d(o_i, o_j) = \sqrt[p]{|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \ldots + |x_{im} - x_{jm}|^p} \quad (3)$$

明可夫斯基距离又称为L_p范式。因此，p=1 时对应曼哈顿距离，又称L_1范式；P=2 时对应欧式距离，又称L_2范式

p=正无穷时称为切比雪夫距离，相应公式如式(4) 所示：

$$d(o_i, o_j) = \lim_{p \to \infty} \left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}} = \max_{1 \le k \le m} |x_{ik} - x_{jk}| \quad (4)$$

编号	账户余额	年龄（岁）	收入（元）	性别	子女个数
100	很低	15	1967	男	0
200	高	25	8453	男	1
300	中	32	6125	女	2
400	低	20	2167	男	1
500	低	55	2439	女	4

账号余额：序数属性

性别：对称二值属性

子女个数：数值属性

年龄：数值属性

……

标称属性的取值只有相同和不同的区别

基于标称属性的两个对象相似度，记为$s_k(o_i, o_j)$计算方法如公式所示：

$$s_k(o_i, o_j) = \begin{cases} 1, & \text{若} x_{ik} = x_{jk} \\ 0, & \text{其他情况} \end{cases}$$

序数属性的取值有顺序区别，不同简单的用相同或不同进行描述。

假设对象$o_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$和$o_j = \{x_{j1}, x_{j2}, \cdots, x_{jm}\}$的第 k 个属性是序数属性，有 p 个不同取值，首先将其取值排序，按照顺序映射为整数 0 到(p-1)，并用此序号代替原来的取值，则基于此属性的两对象相似度，$s_k(o_i, o_j)$的计算方法如公式 2 所示：

$$s_k(o_i, o_j) = 1 - \frac{|x_{ik} - x_{jk}|}{p-1} \quad (2)$$

编号	账户余额	年龄（岁）	收入（元）	性别	子女个数
100	很低 0	15	1967	男	0
200	高 3	25	8453	男	1
300	中 2	32	6125	女	2
400	低 1	20	2167	男	1
500	低 1	55	2439	女	4

$$s_k(o_i, o_j) = 1 - \frac{|x_{ik} - x_{jk}|}{p-1} \quad (2)$$

区间属性、比例属性可以通过取值的差来衡量相异度。

基于此属性的两对象相似度，$s_k(o_i, o_j)$，计算方法如公式 3 所示：

$$s_k(o_i, o_j) = \frac{1}{1+|x_{ik}-x_{jk}|} \quad (3)$$

综合上述内容，对象$o_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$和$o_j = \{x_{j1}, x_{j2}, \ldots, x_{jm}\}$的相似度计算过程如下：

第一步：令 $k = 1$，$c = 0$，$s(o_i, o_j) = 0$.

第二步：按照第 k 个属性的类型分别进行如下计算。

- 对于非对称二值属性，若$x_{ik} = x_{jk} = 0$，转至第(3)步；
  否则，按照标称属性处理
- 对于标称属性，利用公式（1）计算$s_k(o_i, o_j)$
- 对于序数属性，利用公式（2）计算$s_k(o_i, o_j)$
- 对于数值属性，利用公式（3）计算$s_k(o_i, o_j)$

  $c = c + 1$，$s(o_i, o_j) = s(o_i, o_j) + s_k(o_i, o_j)$

第三步：若 $k = m$ 则 $s(o_i, o_j) = s(o_i, o_j)/c$，停止；

否则 $k = k + 1$，转至第二步。

设两个n维向量$\vec{x}=(x_1,x_2,\cdots,x_n)^T$和$\vec{y}=(y_1,y_2,\cdots,y_n)^T$为两个观测，其所定义的距离一般需要满足三个条件：

1. 非负性：$d(\vec{x},\vec{y})\geq 0$，$d(\vec{x},\vec{y})=0$当且仅当$\vec{x}=\vec{y}$

2. 对称性：$d(\vec{x},\vec{y})=d(\vec{y},\vec{x})$

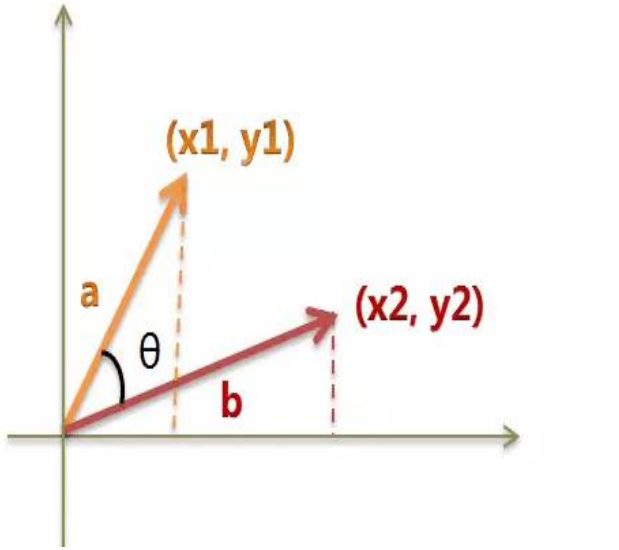
3. 三角不等式：假设存在另一个n维向量$\vec{z}$，$d(\vec{x},\vec{y})\leq d(\vec{x},\vec{z})+d(\vec{z},\vec{y})$

体重的取值起主导作用，因为其范围远远大于身高

余弦距离

- 用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小。

$$sim(X,Y) = cos\theta = \frac{\vec{x} \cdot \vec{y}}{\|x\| \cdot \|y\|}$$



余弦相似度

假设两个对象o_i和o_j对应的向量分别为$o_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$和$o_j = \{x_{j1}, x_{j2}, \ldots, x_{jm}\}$，则余弦相似度$\cos(o_i, o_j)$的计算如公式所示：

$$\cos(o_i, o_j) = \frac{\sum_{k=1}^{m} x_{ik} \times x_{jk}}{\sqrt{\sum_{k=1}^{m} x_{ik}^2} \times \sqrt{\sum_{k=1}^{m} x_{jk}^2}} = \frac{o_i}{\|o_i\|} \cdot \frac{o_j}{\|o_j\|}$$

- 余弦相似度特点：
  - 相似度忽略了各个属性值的绝对大小
  - 两个向量中，只要有一个对象在某维度的取值为0，则该维度相当于被忽略

余弦相似度是信息检索中用于衡量文档相似度的主要方法

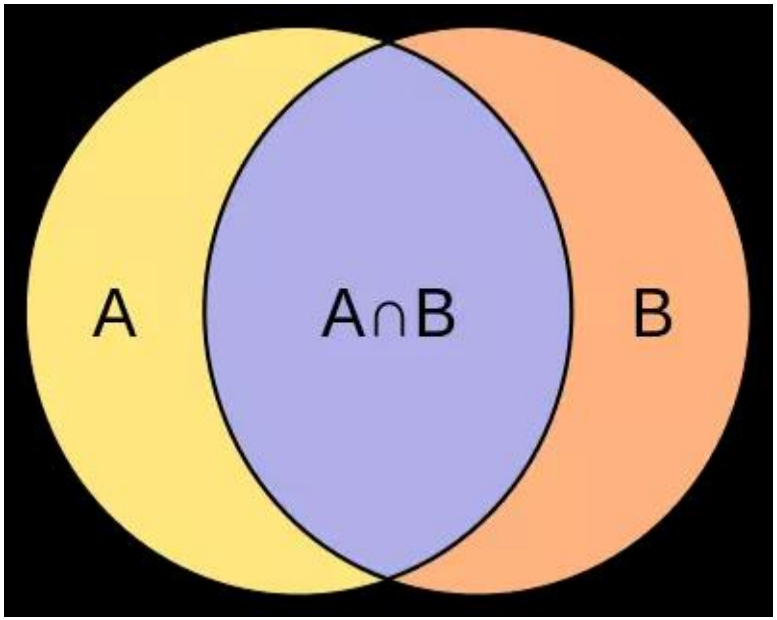
多种属性相似度的综合度量

- 什么时候用余弦距离什么时候用欧式距离呢？
- 欧氏距离体现数值上的绝对差异，而余弦距离体现方向上的相对差异。
- 例如，统计两部剧的用户观看行为，用户A的观看向量为(0,1)，用户B为(1,0)；此时二者的余弦距很大，而欧氏距离很小；分析两个用户对于不同视频的偏好，更关注相对差异，应当使用余弦距离。
- 而当我们分析用户活跃度，以登陆次数(单位：次)和平均观看时长(单：分钟)作为特征时，余弦距离会认为(1,10)、(10,100)两个用户距离很近；但显然这两个用户活跃度是有着极大差异的，此时我们更关注数值绝对差异，应当使用欧氏距离。

Jaccard距离

- Jaccard系数：判断两个集合的相似度， jaccard similarity coefficient。非对称二元属性的相似。性给定两个集合A,B jaccard 系数定义为A与B交集的大小与并集大小的比值，jaccard值越大说明相似度越高。

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{|A\cap B|}{|A|+|B|-|A\cap B|}$$



当A和B都为空时，jaccard(A,B)=1；

Jaccard Similarity

Set A = 

Set B = 

Jaccard Similarity J (A,B) = | Intersection (A,B) | / | Union (A,B) |

首先计算出A

= 2 / 7

= 0.286

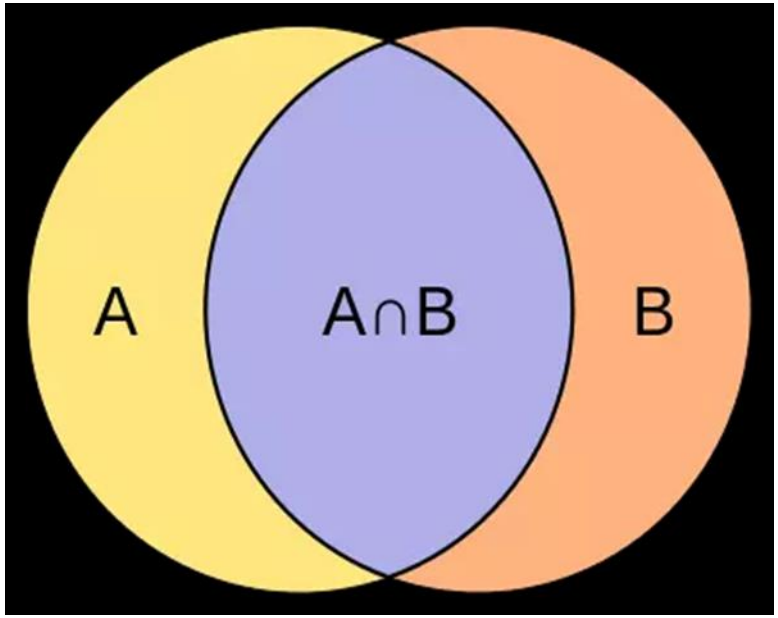
Union 

Intersection (A,B) = 

| Union (A,B) | = 7 | Intersection (A,B)| = 2

- jaccard距离：用于描述不相似度

$$d_j(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} = \frac{A \Delta B}{|A \cup B|}$$



类间距离

- 距离函数都是关于两个样本的距离刻画，然而在聚类应用中，最基本的方法是计算类间的距离。

- 设有两个类C_a和C_b，它们分别有m和h个元素，它们的中心分别为γ_a和γ_b。设元素$x \in C_a$，$y \in C_b$，这两个元素间的距离通常通过类间距离来刻画，记为$D(C_a, C_b)$。

- 类间距离的度量主要有：
  - 最短距离法：
  - 最长距离法：
  - 中心法：
  - 类平均法：
  - 离差平方和

- 类间距离的度量主要有：
  - 最短距离法：定义两个类中最靠近的两个元素间的距离为类间距离。

$$D_{kl} = \min_{i,j}\left\lfloor d_{ij} \right\rfloor$$

  - 最长距离法：定义两个类中最远的两个元素间的距离为类间距离。

$$D_{kl} = \max_{i,j}\left\lfloor d_{ij} \right\rfloor$$

  - 中心法：定义两类的两个中心间的距离为类间距离。

$$\overline{x_i} = \frac{1}{n_i}\sum_{x \in C_i} x \qquad D_C(C_a, C_b) = d(r_a, r_b)$$

  - 类平均法：它计算两个类中任意两个元素间的距离，并且综合他们为类间距离：
  - 离差平方和：

$$D_G(C_a, C_b) = \frac{1}{mh}\sum_{x \in C_a}\sum_{y \in C_b} d(x, y)$$

离差平方和

- 离差平方和用到了类直径的概念：
  - 类的直径反映了类中各元素间的差异，可定义为类中各元素至类中心的欧氏距离之和，其量纲为距离的平方：

$$r_a = \sum_{i=1}^{m} (x_i - \overline{x_a})^T (x_i - \overline{x_b})$$

- 根据上式得到两类C_a和C_b的直径分别为γ_a和γ_b，类$C_{a+b} = C_a \cup C_b$的直径为γ_{a+b}，则可定义类间距离的平方为：

$$D_W^2(C_a, C_b) = r_{a+b} - r_a - r_b$$

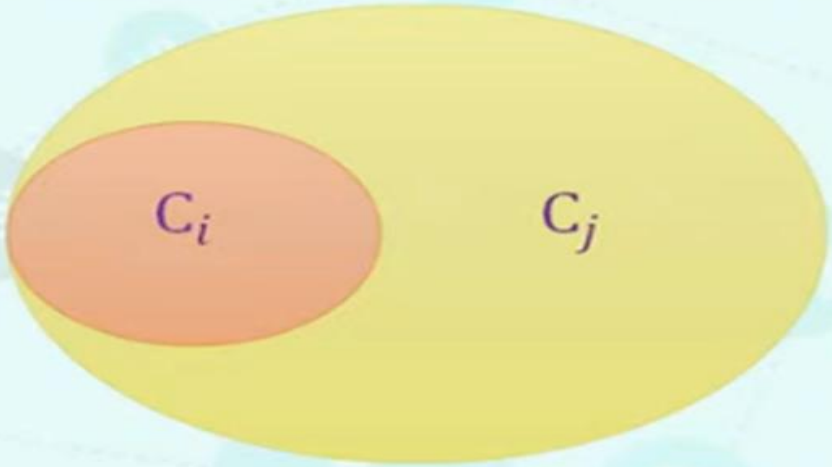
聚类的原则

- 基本原则：
  - 高内聚、低耦合
- 衡量聚类效果的标准
  - 簇内相似度越高、簇间相似度越低，聚类效果越好

轮廓系数法：

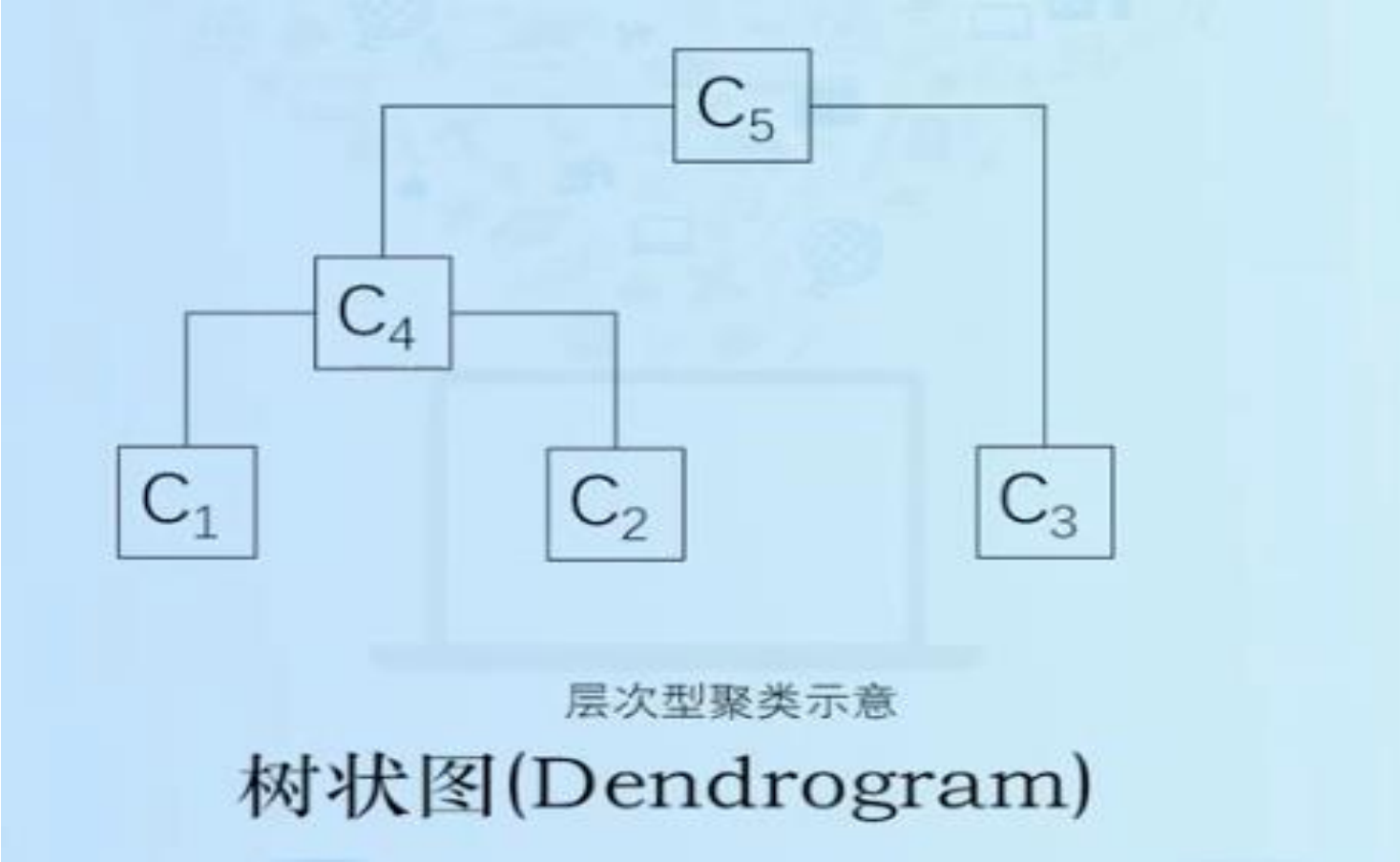
如果形成的各个簇之间没有交集，即任取不同的两个簇C_i和C_j，$C_i \cap C_j = \varnothing$，则这类聚类通常称为划分型聚类

轮廓系数法：

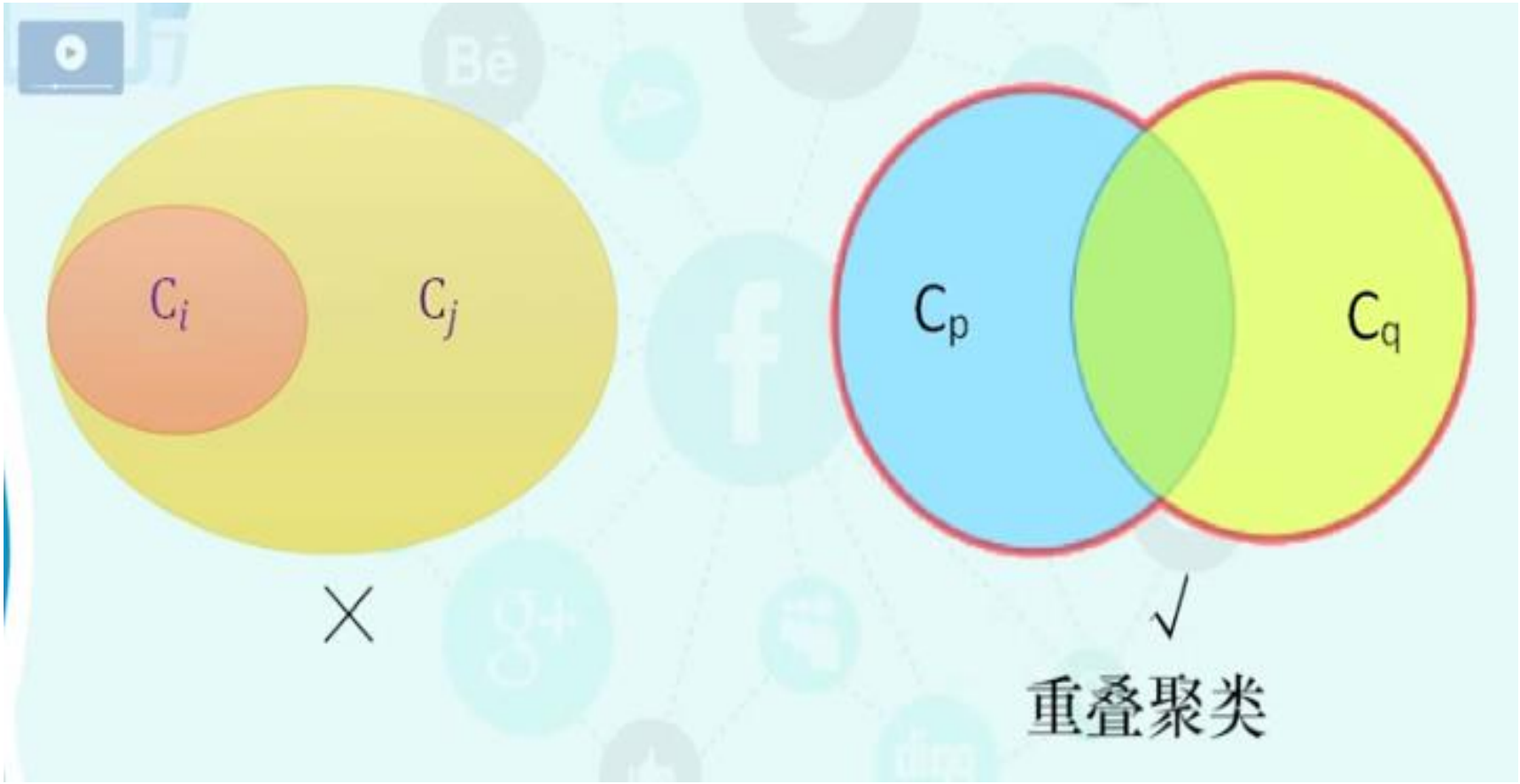


如果有交集的簇之间 必然存在包含关系，即任取两个不同的簇C_i和C_j，如果$C_i \cap C_j \neq \emptyset$，则$C_i \subset C_j$，或者$C_j \subset C_i$，那么我们称簇之间具有层次关系，这种聚类称为层次型聚类

- 轮廓系数法：



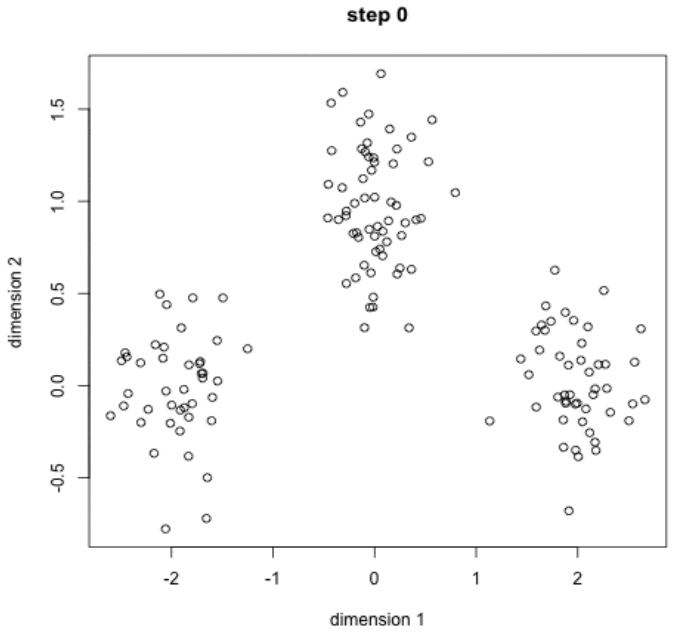
- 轮廓系数法：

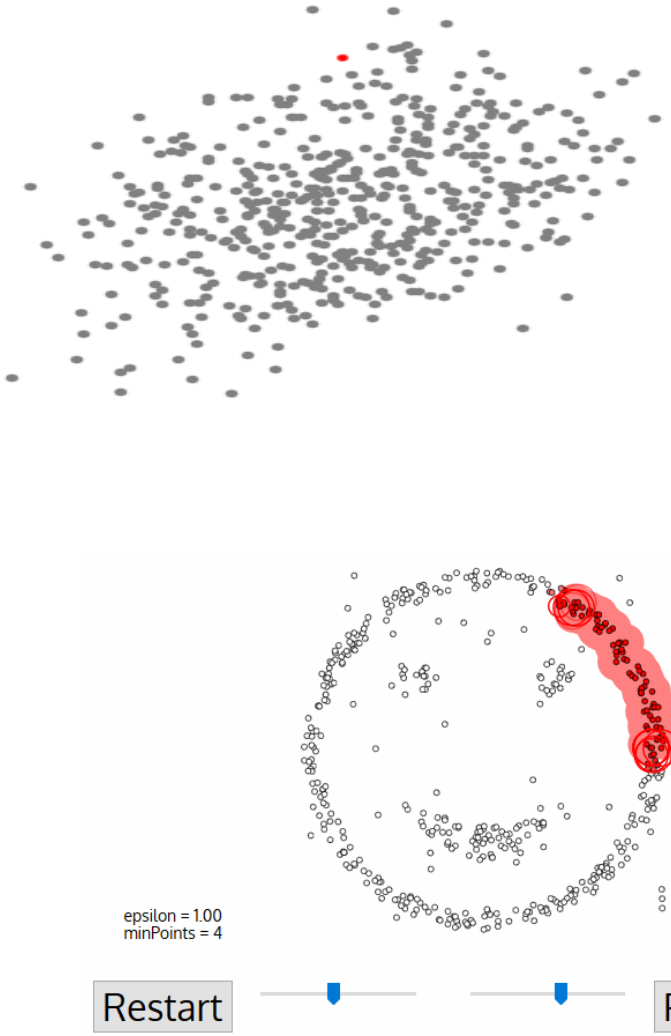


- 根据数据与簇之间的关系，可分为完全聚类和部分聚类：

如果 $D = C_1 \cup C_2 \cup \ldots \cup C_k$，即所有对象都被分配到簇中，则为完全聚类；否则，若 D 中存在对象 $o_i \notin C_1 \cup C_2 \cup \ldots \cup C_k$，则为部分聚类，通常我们将那些未被分到任意一个簇中的对象称为孤立点(Outlier)。

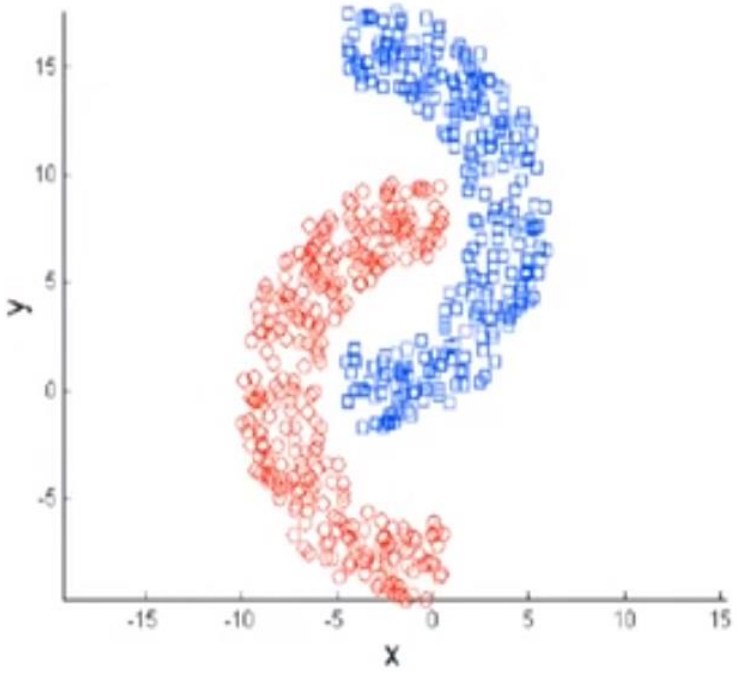
聚类方法概述

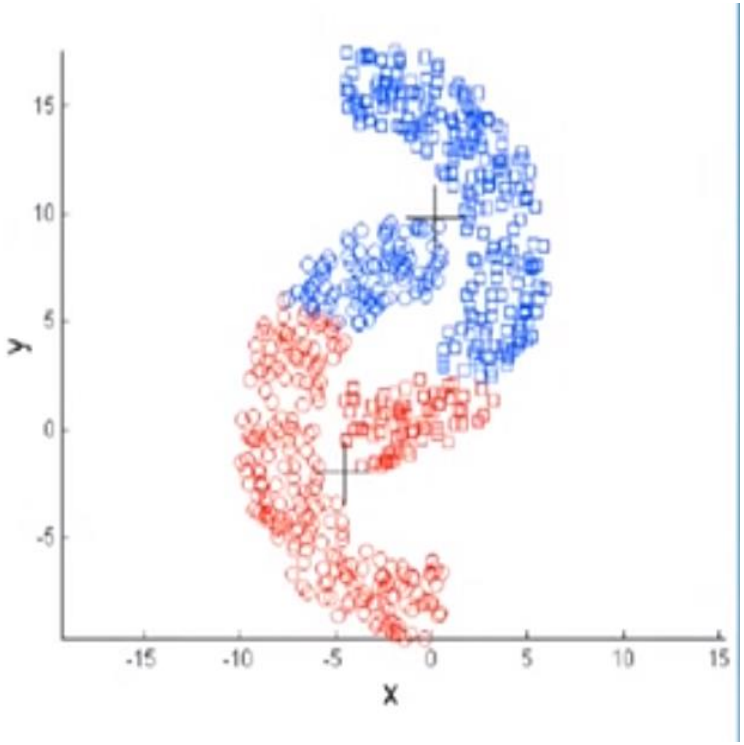

step 0
dimension 2
dimension 1
-0.5
0.0
0.5
1.0
1.5
-2
-1
0
1
2


epsilon = 1.00
minPoints = 4
Restart
Pause

一. 聚类方法概述

评判聚类好坏：$$J_e = \sum_{i=1}^{c} \sum_{x \in D_i} \|x - m_i\|^2, \qquad m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$





2.2 聚类分析方法的分类

按照聚类的标准，聚类方法可分为如下两种：

- 统计聚类方法：这种聚类方法主要基于对象之间的几何距离的。
- 概念聚类方法：概念聚类方法基于对象具有的概念进行聚类。

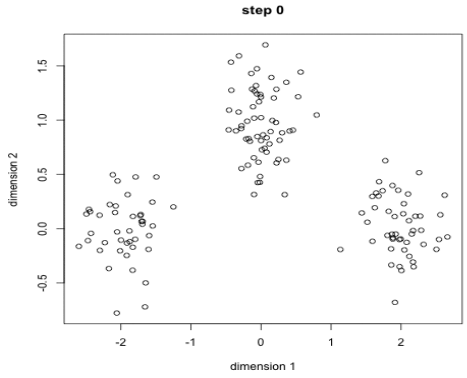
按照聚类算法所处理的数据类型，聚类方法可分为三种：

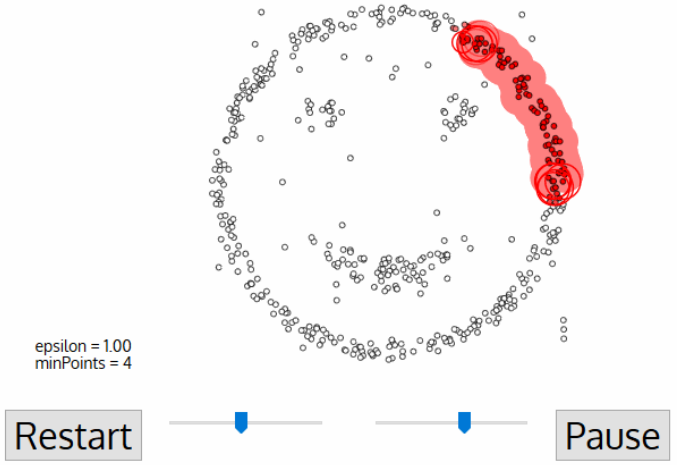
- 数值型数据聚类方法：所分析的数据的属性只限于数值数据。
- 离散型数据聚类方法：所分析的数据的属性只限于离散型数据。
- 混合型数据聚类方法：能同时处理数值和离散数据。

2.3 聚类分析的目标

按照聚类的尺度，聚类方法可被分为以下三种：

- 基于距离的聚类算法：用各式各样的距离来衡量数据对象之间的相似度，如k-means、k-medoids、BIRCH、CURE等算法。
- 基于密度的聚类算法：相对于基于距离的聚类算法，基于密度的聚类方法主要是依据合适的密度函数等。
- 基于互连性(Linkage-Based)的聚类算法：通常基于图或超图模型。高度连通的数据聚为一类。





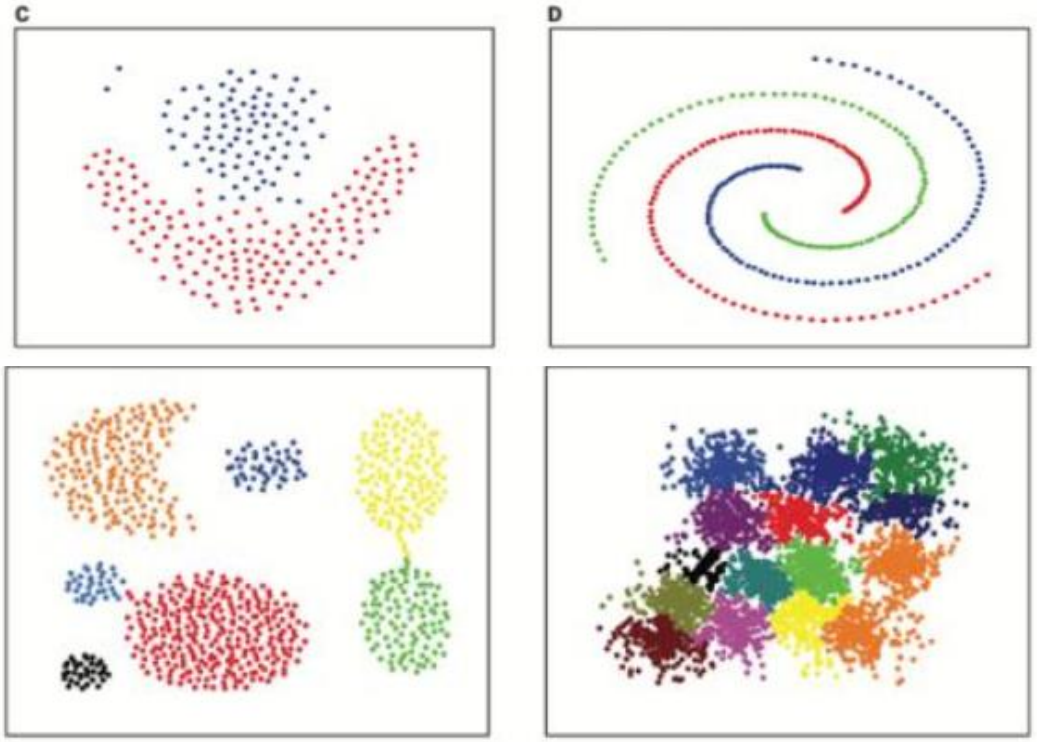
2.3 聚类分析的目标

聚类在数据挖掘中的典型应用有：

1、聚类分析可以作为其它算法的预处理步骤

2、聚类分析可以作为一个独立的工具来获得数据的分布情况

3、聚类分析可以完成孤立点挖掘



- 聚类技术：
  - 划分法：k均值、k中心点
  - 层次法：凝聚层次聚类、分裂层次聚类
  - 基于密度的方法：Density-based approach
  - 基于模型的方法：Model-based approach

二. 聚类方法-kmeans算法

k-means（K均值算法）

(a) 给定集合D，有n个样本点

(b) 随机指定k个点，作为k个子集的质心

(c) 根据样本点与k个质心的距离远近，将每个样本点划归最近质心所在的子集

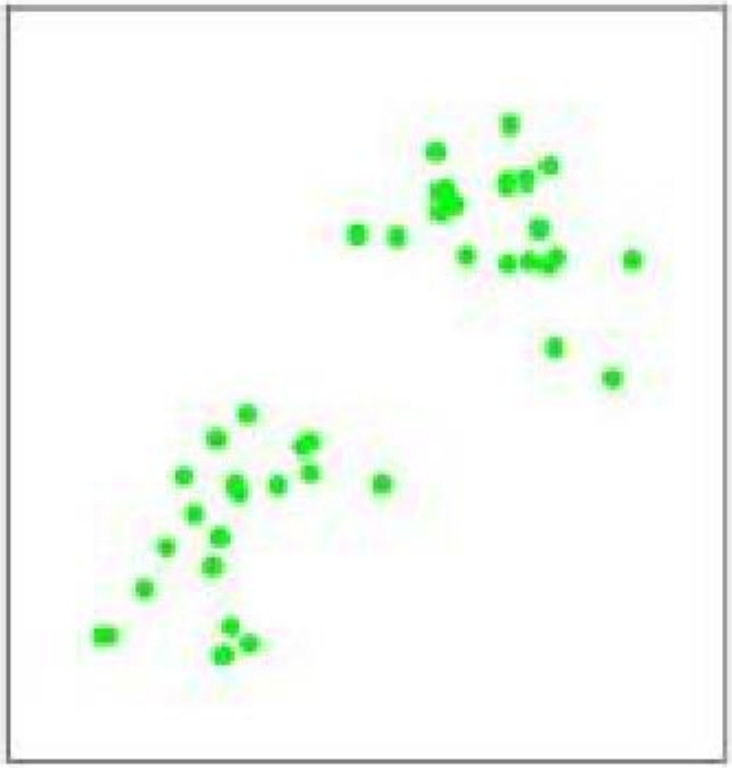
(d) 对k个子集重新计算质心

(e) 根据新的质心，重复操作(c)

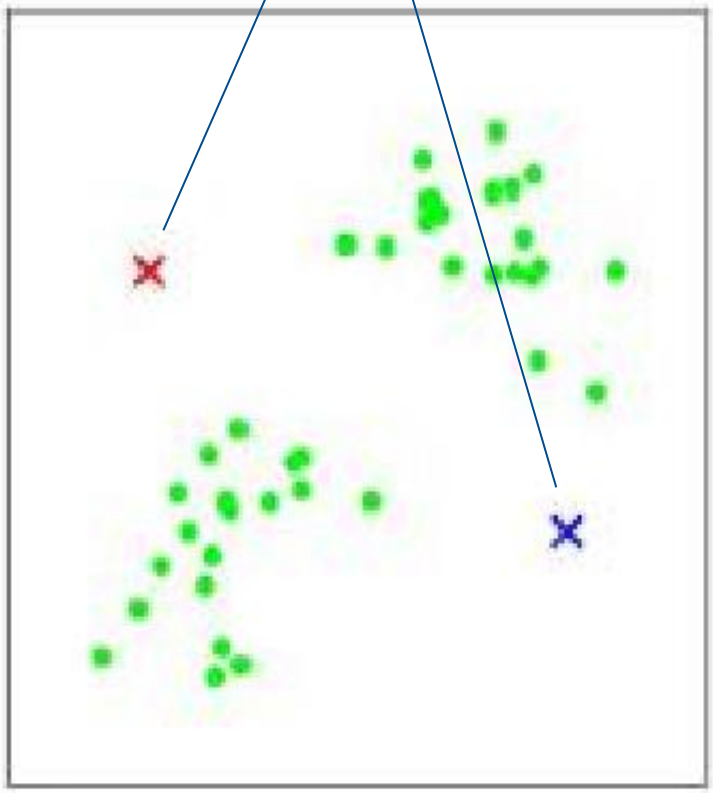
(f) 重复操作(d)和(e)，直至结果足够收敛或者不再变化

二. 聚类方法-kmeans算法

a、随机指定两个点，作为两个子集的质心



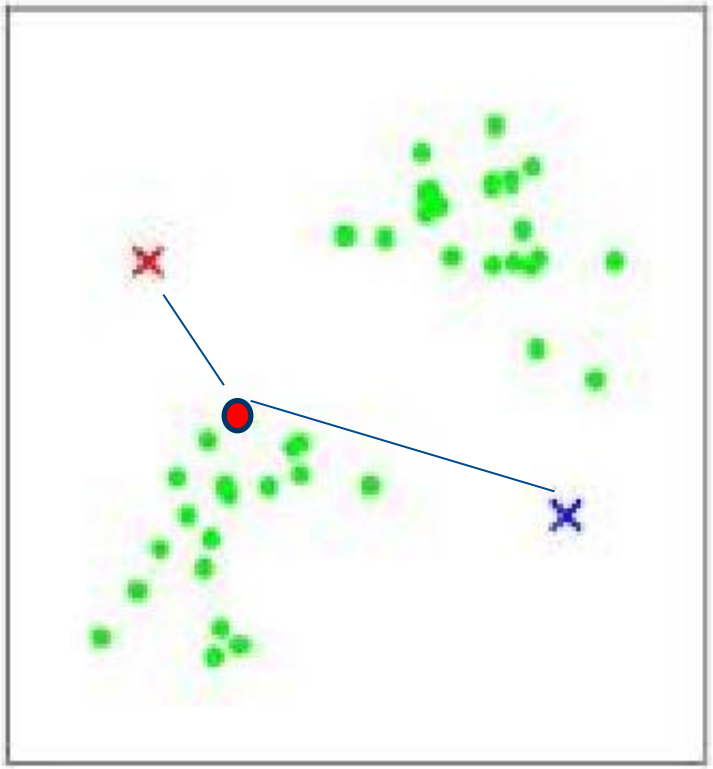
(a)



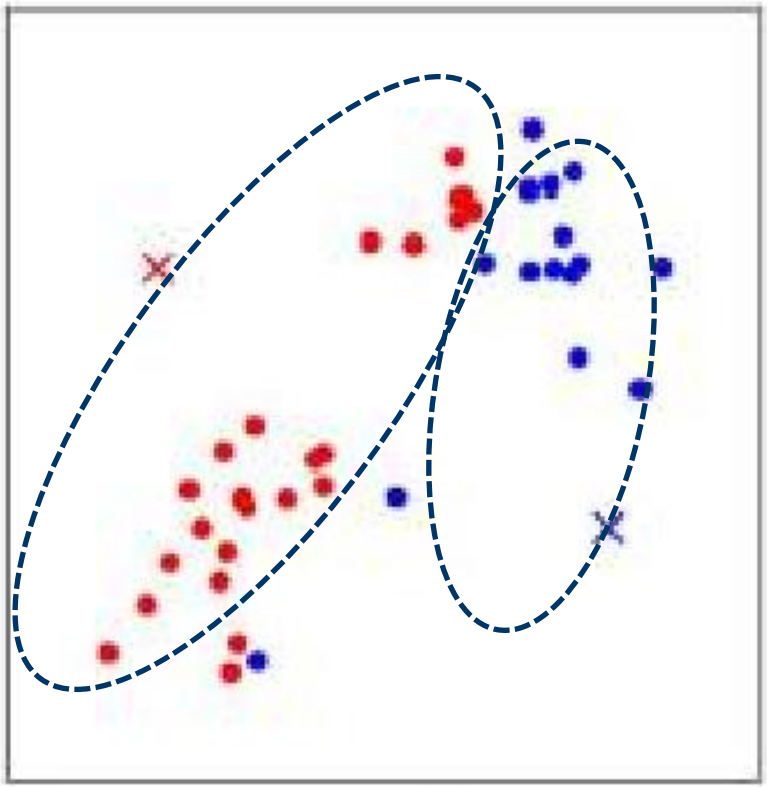
(b)

二. 聚类方法-kmeans算法

b、第一次迭代



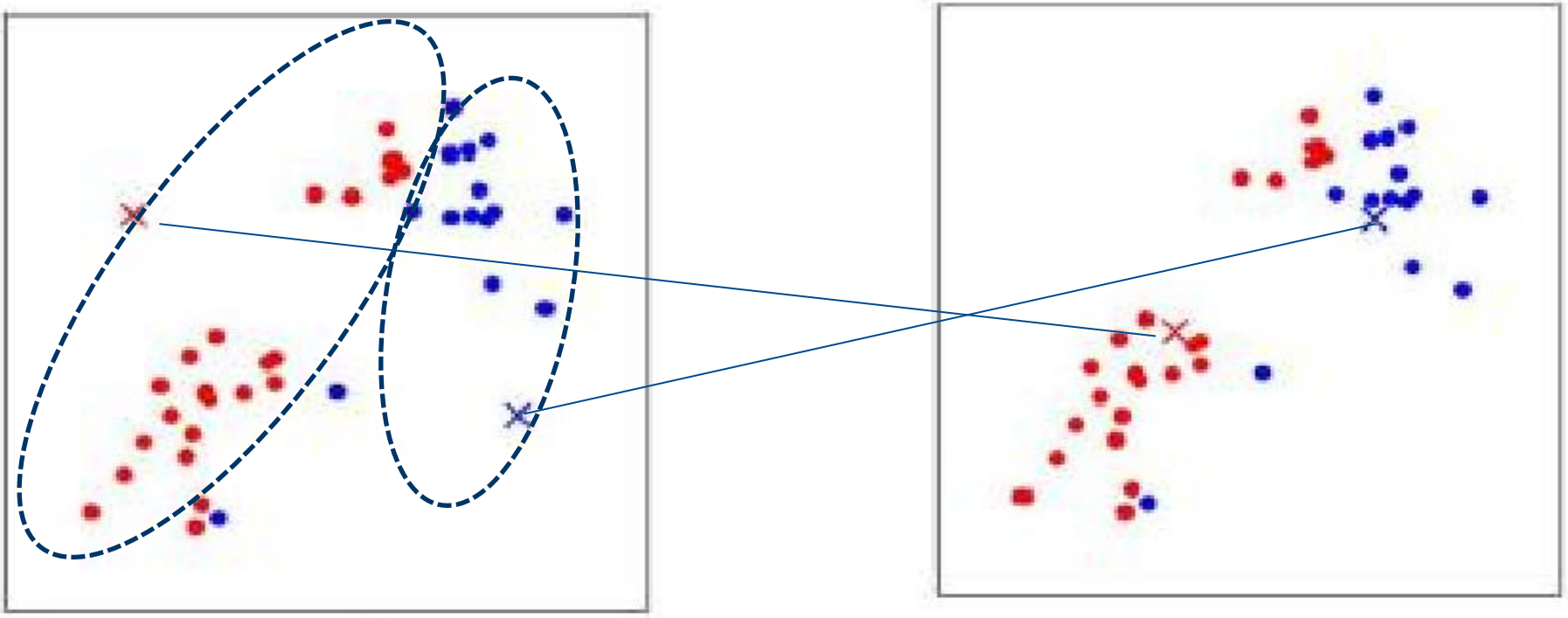
(b)



(c)

二. 聚类方法-kmeans算法

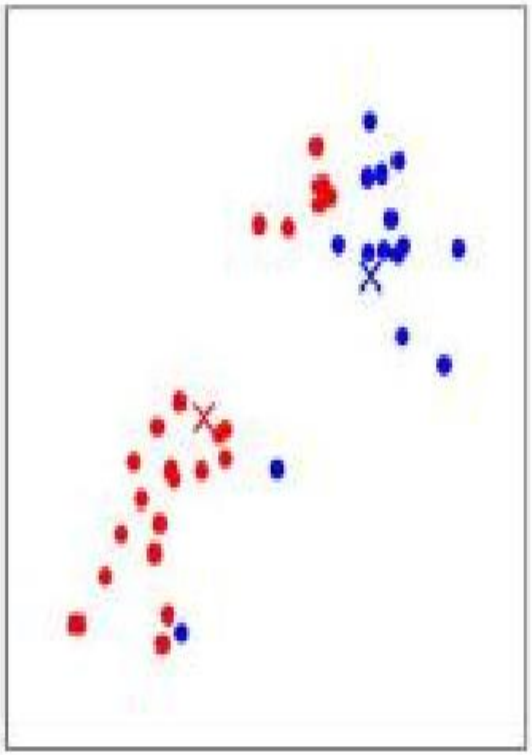
c、重新计算质心：$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n}$



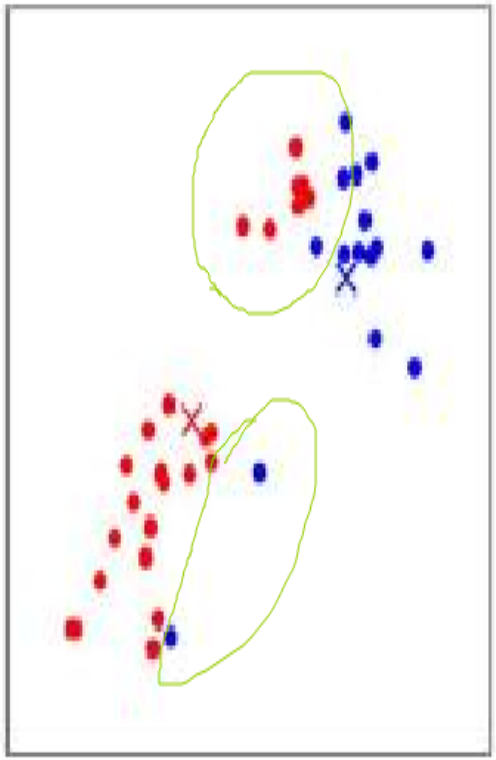
(c)　　(d)

二. 聚类方法-kmeans算法

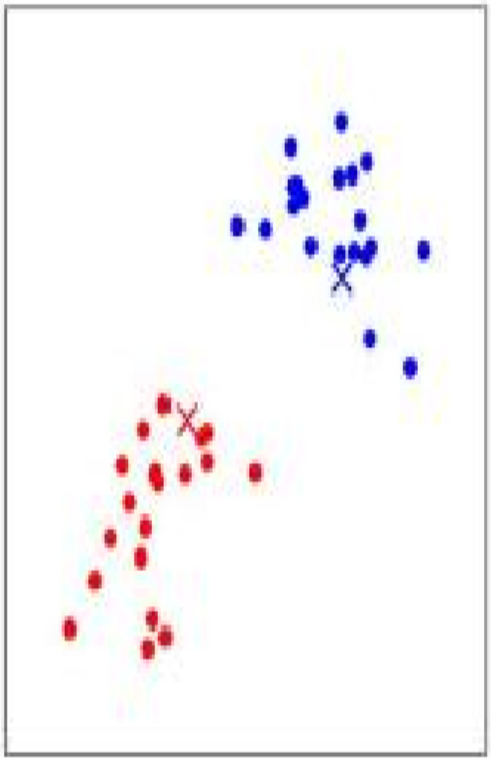
d、第二次迭代，重新计算质心



(d)



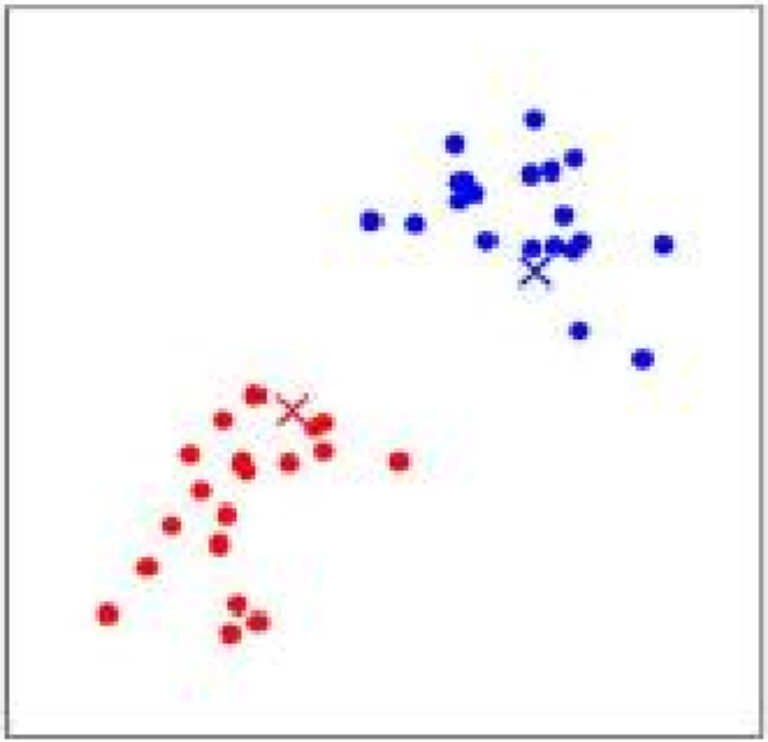
(d)



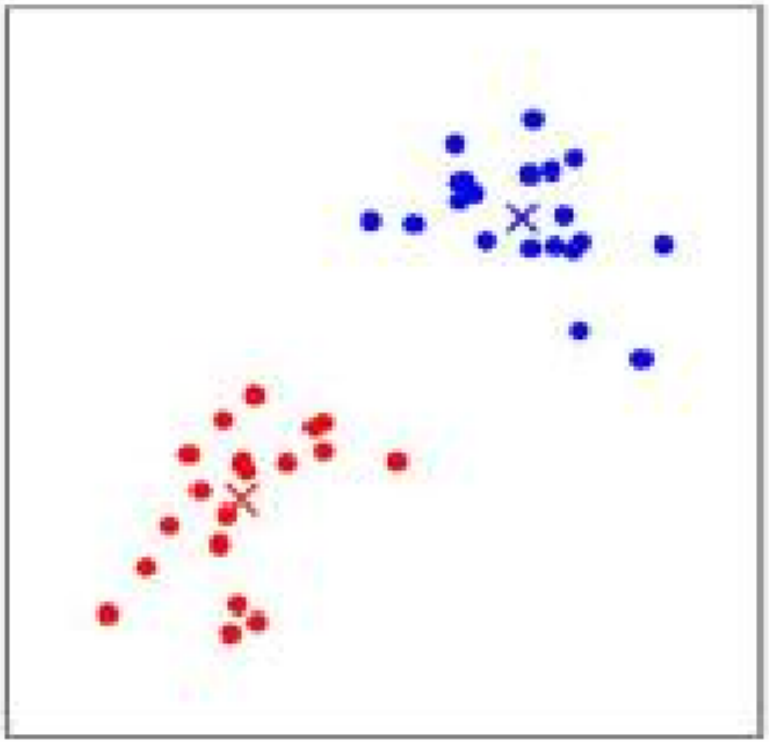
(e)

二. 聚类方法-kmeans算法

e、重复迭代，计算质心，直到质心足够收敛或不再变化

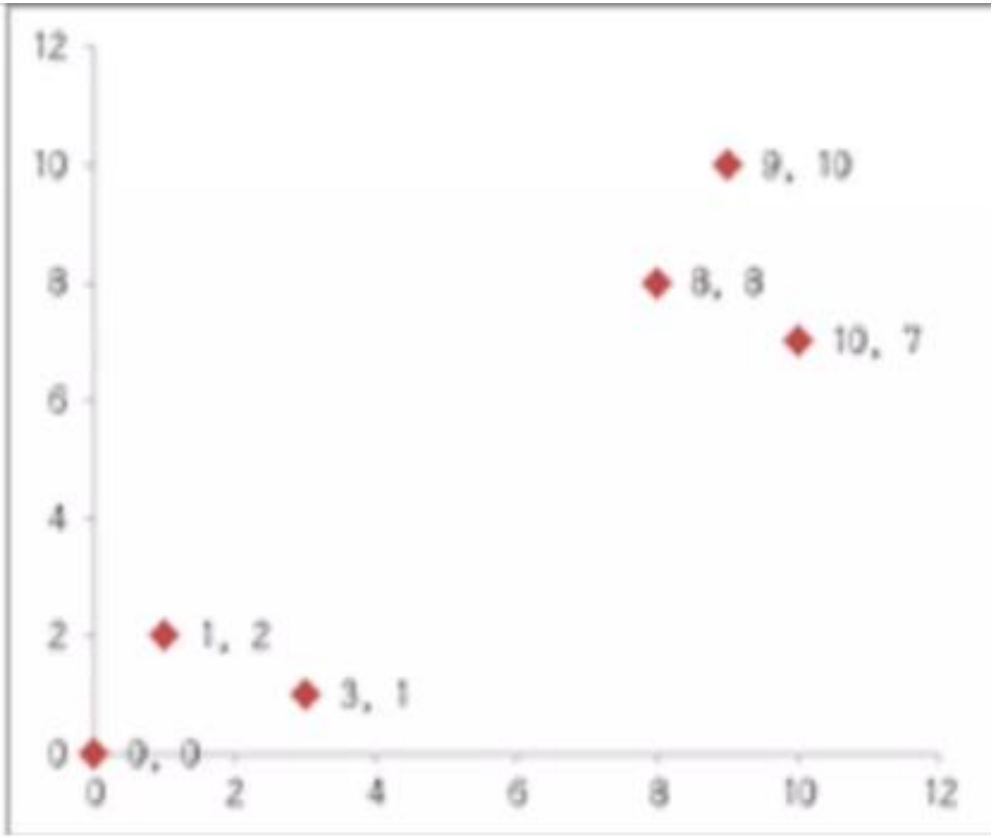


(e)



(f)

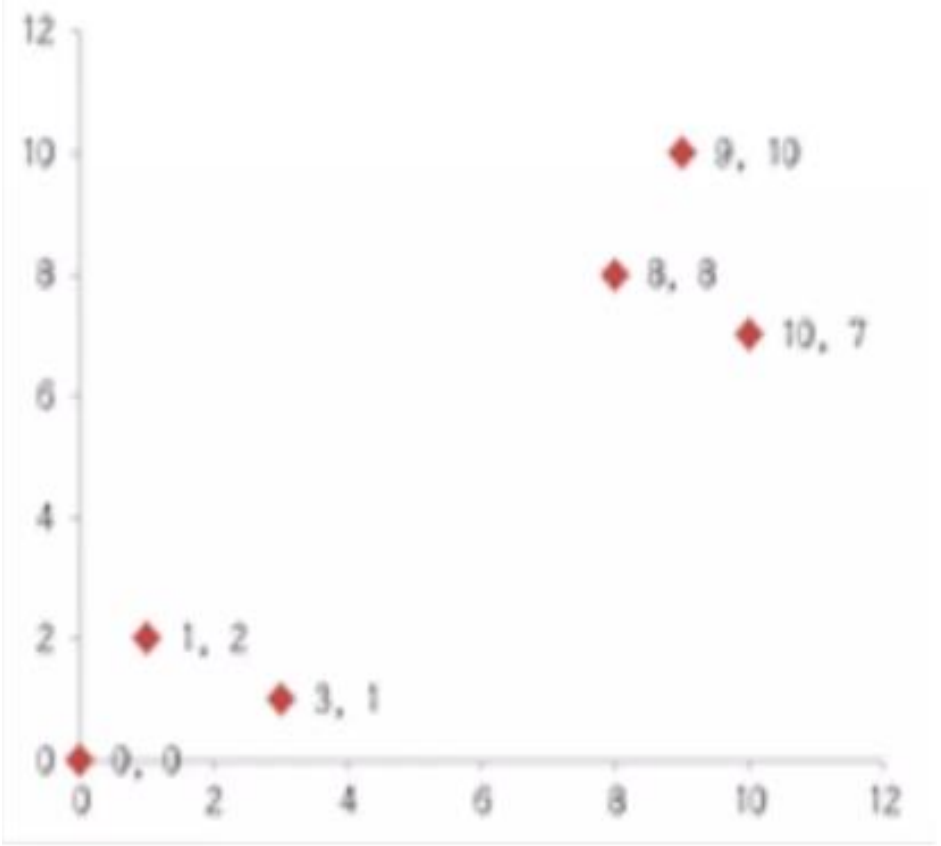
	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7



- 令K等于2，我们随机选择两个点：P1和P2

- 通过欧氏距离计算剩余点分别到这两个点的距离：

	P1	P2
P3	3.16	2.24
P4	11.3	9.22
P5	13.5	11.3
P6	12.2	10.3

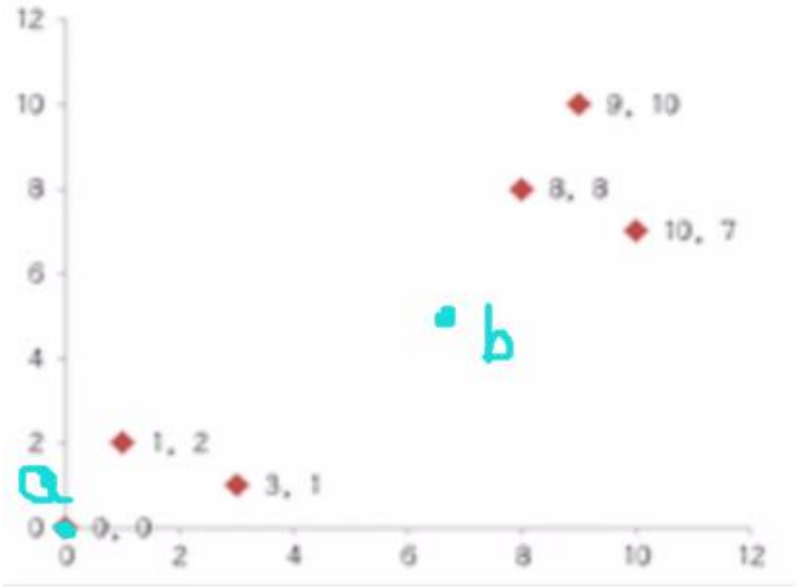


- 第一次分组后结果：
- 组A：P1
- 组B：P2、P3、P4、P5、P6

- 分别计算A组和B组的质心：
- A组质心还是Pa=（0，0）
- B组新的质心坐标为：Pb=（1+3+8+9+10）/5
- （2+1+8+10+7）/5
- =（6.2，5.6）

	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7

- 再次计算每个点到质心的距离：



	P1	P哥
P2	2.24	6.3246
P3	3.16	5.6036
P4	11.3	3
P5	13.5	5.2154
P6	12.2	4.0497

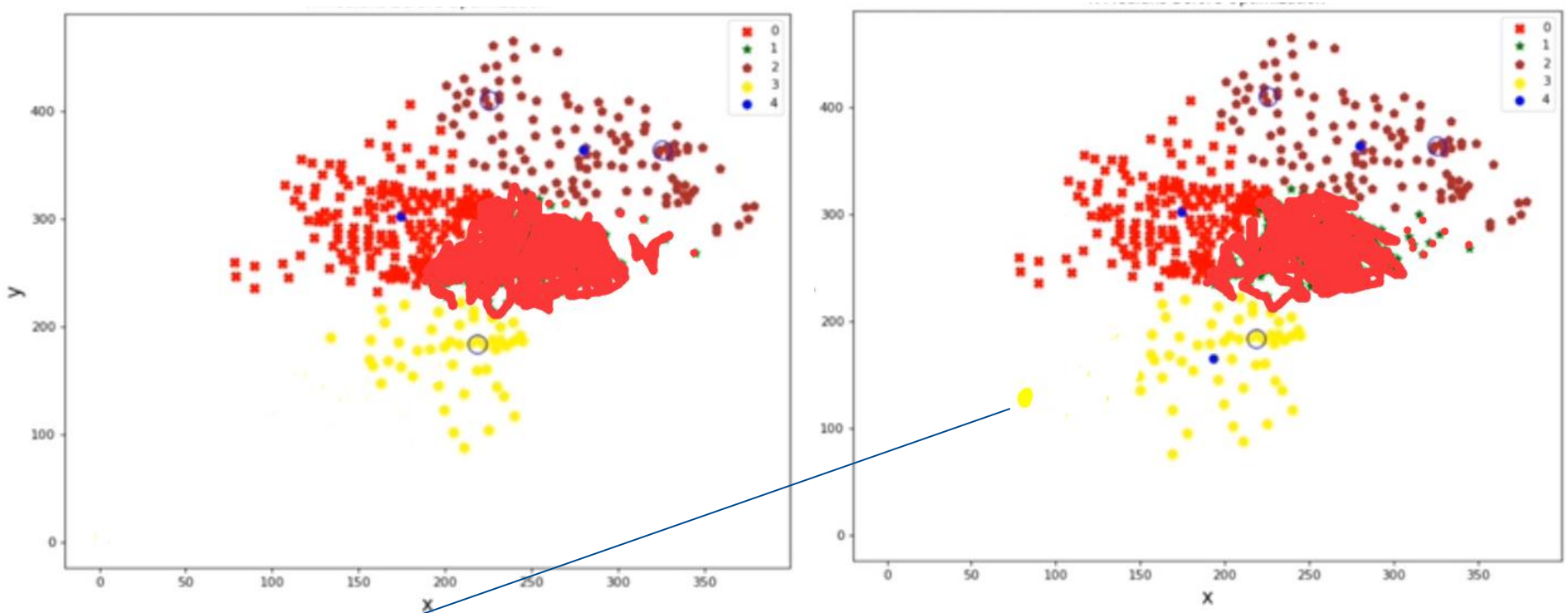
- 第二次分组结果：
- 组A：P1、P2、P3
- 组B：P4、P5、P6

- 再次计算每个点到质心的距离：

	P哥1	P哥2
P1	1.4	12
P2	0.6	10
P3	1.4	9.5
P4	47	1.1
P5	70	1.7
P6	56	1.7

- 第三次分组结果：
- 组A：P1、P2、P3
- 组B：P4、P5、P6
- 可以发现，第三次分组结果和第二次分组结果一致，说明已经收敛，聚类结束。

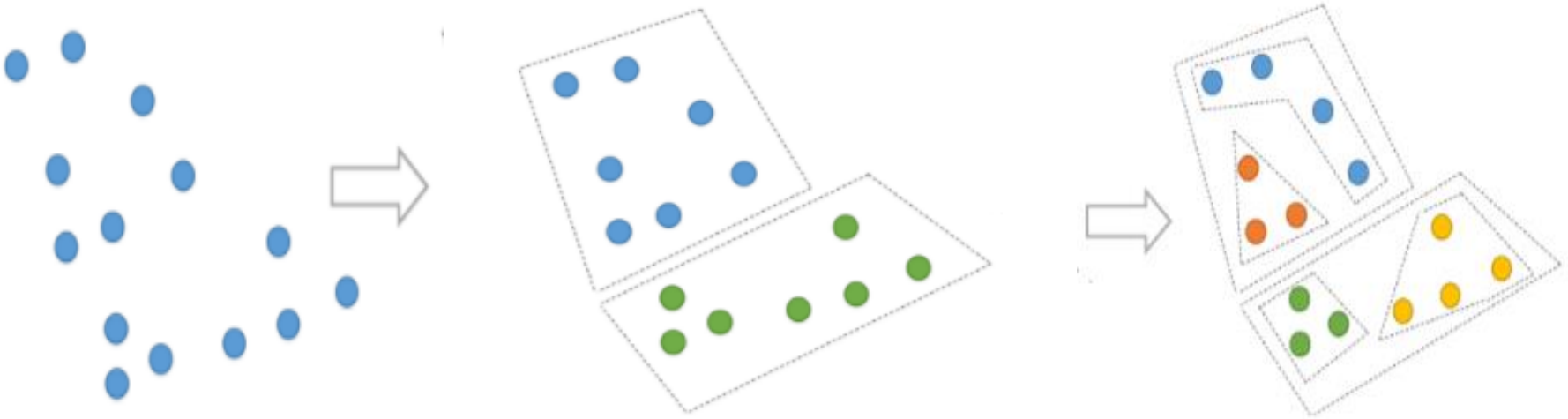
三、k-means 存在的问题

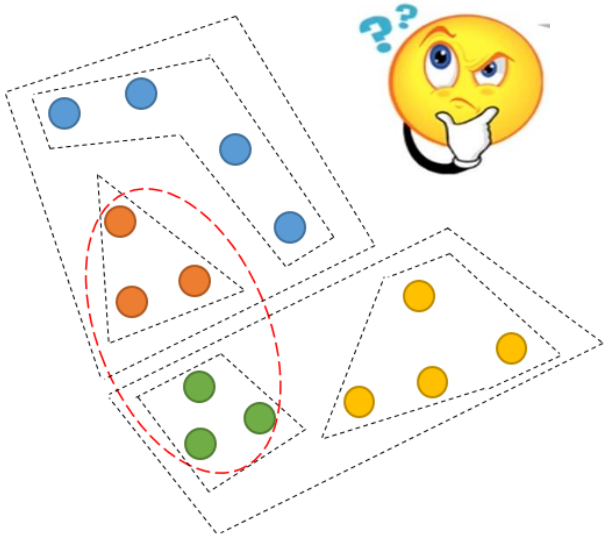


孤立点：与数据的其他部分差异较大

孤立点的处理方法：标准分数优化方法，超出偏移值，进行去除

三、k-means 存在的问题





一旦两个点在最开始被划分到了不同的簇，即使这两个点距离很近，在后面的过程中也不会被聚类到一起。

k-means算法的性能分析

- **主要优点：**
  - 是解决聚类问题的一种经典算法，简单、快速。
  - 对处理大数据集，该算法是相对可伸缩和高效率的。
  - 当结果簇是密集的，它的效果较好。
- **主要缺点**
  - 在簇的平均值被定义的情况下才能使用，可能不适用于某些应用。
  - 必须事先给出k（要生成的簇的数目），而且对初值敏感，对于不同的初始值，可能会导致不同结果。
  - 不适合于发现非凸面形状的簇或者大小差别很大的簇。而且，它对于“躁声”和孤立点数据是敏感的。

K-medoids 方法

- k-中心点（K-medoids）：算法k-means算法对于孤立点是敏感的。为了解决这个问题，不采用簇中的平均值作为参照点，可以选用簇中位置最中心的对象，即中心点作为参照点。这样划分方法仍然是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执行的。

- 在K中心点算法中，每次迭代后的质点都是从聚类的样本点中选取，k中心点算法不采用簇中对象的平均值作为簇中心，而选用簇中离平均值最近的对象作为簇中心。

- Partitioning Around Medoids (PAM)算法，是一种常见的k中心点聚类方法，利用贪婪搜索，不一定可以找到最优解，但是比穷尽搜索更快。
- 对下列表中的10个数据聚类，每个数据的维度都为2，k=2。

X_1	2	6
X_2	3	4
X_3	3	8
X_4	4	7
X_5	6	2
X_6	6	4
X_7	7	3
X_8	7	4
X_9	8	5
X_{10}	7	6

$$\underbrace{3+0+4+4}_{\text{objects in cluster 1}} + \underbrace{3+1+1+0+2+2}_{\text{objects in cluster 2}} = 20$$

- 随机挑
，4）.

Data object		Distance to	
i	X_i	$c_1 = (3,4)$	$c_2 = (7,4)$
1	(2, 6)	3	7
2	(3, 4)	0	4
3	(3, 8)	4	8
4	(4, 7)	4	6
5	(6, 2)	5	3
6	(6, 4)	3	1
7	(7, 3)	5	1
8	(7, 4)	4	0
9	(8, 5)	6	2
10	(7, 6)	6	2
Cost		11	9

- 随机挑选k=2个中心点：c1=（3，4），c2=（7，4）.那么将所有点到这两点的距离计算出来

Data object		Distance to	
i	X_i	$c_1 = (3,4)$	$c_2 = (7,4)$
1	(2, 6)	3	7
2	(3, 4)	0	4
3	(3, 8)	4	8
4	(4, 7)	4	6
5	(6, 2)	5	3
6	(6, 4)	3	1
7	(7, 3)	5	1
8	(7, 4)	4	0
9	(8, 5)	6	2
10	(7, 6)	6	2
Cost		11	9

- 挑选一个非中心点O’，比如X7，O‘=（7，3）。那么此时这两个中心点暂时变成了c1(3,4) and O’(7,3)；total cost = 3+4+4+2+2+1+3+3 = 22

i	O′		Data objects (X_i)		Cost (distance)
1	7	3	2	6	8
3	7	3	3	8	9
4	7	3	4	7	7
5	7	3	6	2	**2**
6	7	3	6	4	**2**
8	7	3	7	4	**1**
9	7	3	8	5	**3**
10	7	3	7	6	**3**

2.4 聚类分析方法：k-means算法

- K值：要得到的簇的个数
- 质心：每个簇的均值向量，即向量各维取平均
- 距离量度：

欧式距离： $d_{12}=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

曼哈顿距离： $d_{12}=|x_1-x_2|+|y_1-y_2|$

切比雪夫距离： $d_{12}=\max(|x_1-x_2|,|y_1-y_2|)$

余弦距离： $\cos\theta=\frac{x_1x_2+y_1y_2}{\sqrt{x_1^2+y_1^2}\sqrt{x_2^2+y_2^2}}$

Jaccard相似系数： $J(A,B)=\frac{|A\cap B|}{|A\cup B|}$

相关系数： $\rho_{XY}=\frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}=\frac{E((X-EX)(Y-EY))}{\sqrt{D(X)}\sqrt{D(Y)}}$

2.4 聚类分析方法-k-means算法

K-means算法主要思想

算法5-1 *k*-means算法

输入：簇的数目*k*和包含*n*个对象的数据库。

输出：*k*个簇，使平方误差准则最小。

(1)assign initial value for means; /*任意选择*k*个对象作为初始的簇中心；*/

(2) REPEAT

(3) FOR *j*=1 to *n* DO assign each X_j to the closest clusters;

(4) FOR *i*=1 to *k* DO / *更新簇平均值*/

(5) Compute /*计算准则函数*E*/

(6) UNTIL *E*不再明显地发生变化。

3.1 k-means算法改进

- K-modes算法可以看做是k-means算法在非数值属性集合上的版本。
- 具体算法步骤如下：
- 1.随机确定k个聚类中心
- 2.对于样本xj，分别比较其与k个中心之间的距离（这里的距离为不同属性值的个数）
- 3.将xj划分到距离最小的簇，在全部的样本都被划分完毕之后，重新确定簇中心，向量Ci中的每一个分量都更新为簇i中的众数
- 重复步骤2和3，直到总距离（各个簇中样本与各自簇中心距离之和）不再降低，返回最后的聚类结果

3.1 k-means算法改进

$$X = \begin{Bmatrix} 1 & 5 & 0 & 3 \\ 1 & 6 & 1 & 3 \\ 3 & 6 & 0 & 3 \\ 2 & 7 & 0 & 4 \\ 1 & 5 & 1 & 4 \\ 2 & 5 & 1 & 2 \end{Bmatrix} \qquad Y = \begin{Bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{Bmatrix}$$

- 即第1、2、3、5个样本被划分到C1，
- 即第4、6个样本被划分到C2
- 接下来更新C1和C2 C1=[1,6,1,3],C2=[2,7,0,4] 后面的步骤就是不断重复步骤二和三了

- 假设有N个样本，M个属性且全是离散的，簇的个数为k。
- 假设有7个样本，每个样本有4个属性，表示为矩阵X

$$X = \begin{Bmatrix} 1 & 5 & 0 & 3 \\ 1 & 6 & 1 & 3 \\ 3 & 6 & 0 & 3 \\ 2 & 7 & 0 & 4 \\ 1 & 5 & 1 & 4 \\ 2 & 5 & 1 & 2 \end{Bmatrix}$$

随机确定2个聚类中心

C1=[1,5,0,3],C2=[2,5,1,2]

划分结果用Y表示

$$Y = \begin{Bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{Bmatrix}$$