战略
研发
市场

数据挖掘原理

主讲教师：李志勇

数据科学系
数字农业工程技术研究中心
移动：13882213811 电邮: lzy@sicau.edu.cn

团队
责任
合作

第六章：数据聚类

——物以类聚，人以群分

主讲教师：李志勇

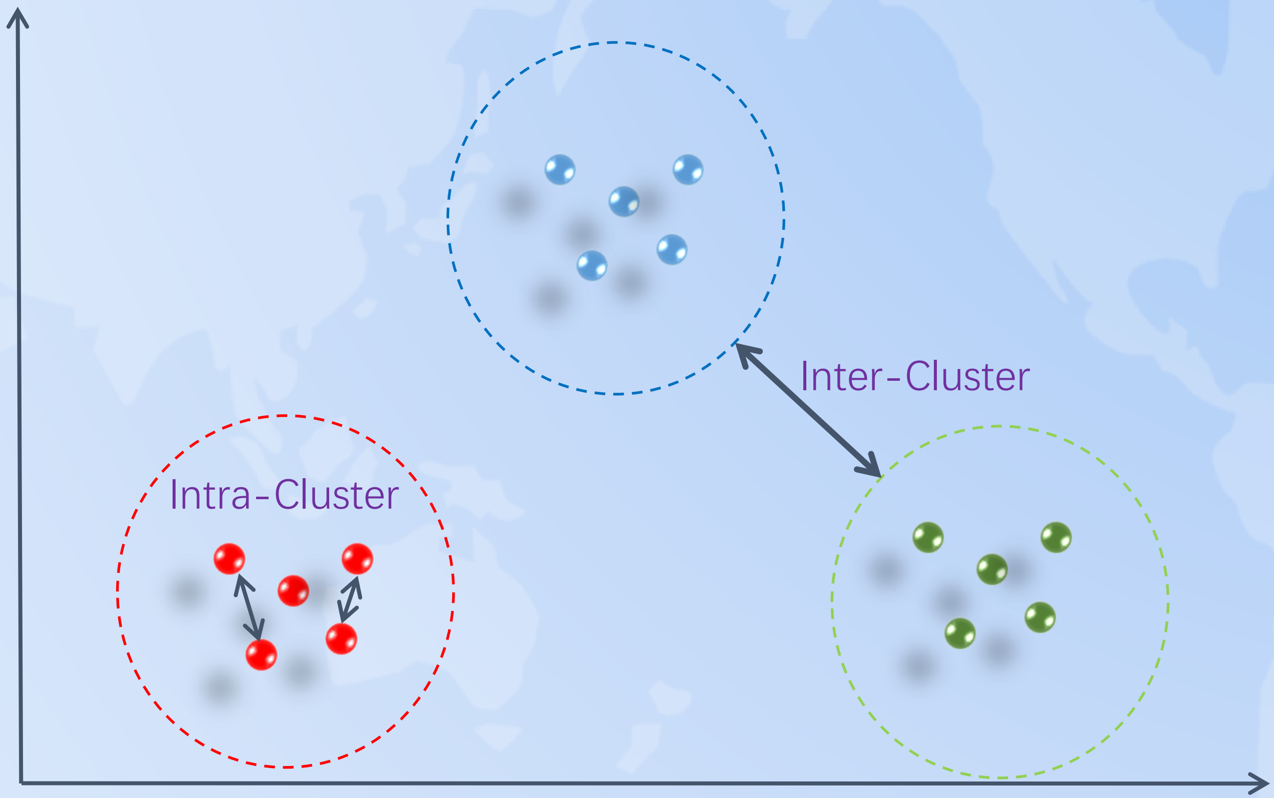
主要介绍内容

- 6.1 基于划分的聚类分析
- 6.2 层次聚类
- 6.3 基于密度的聚类
- 6.4 层次聚类方法
- 6.5 基于密度的聚类方法

6.1 聚类问题概述

聚类分析的定义

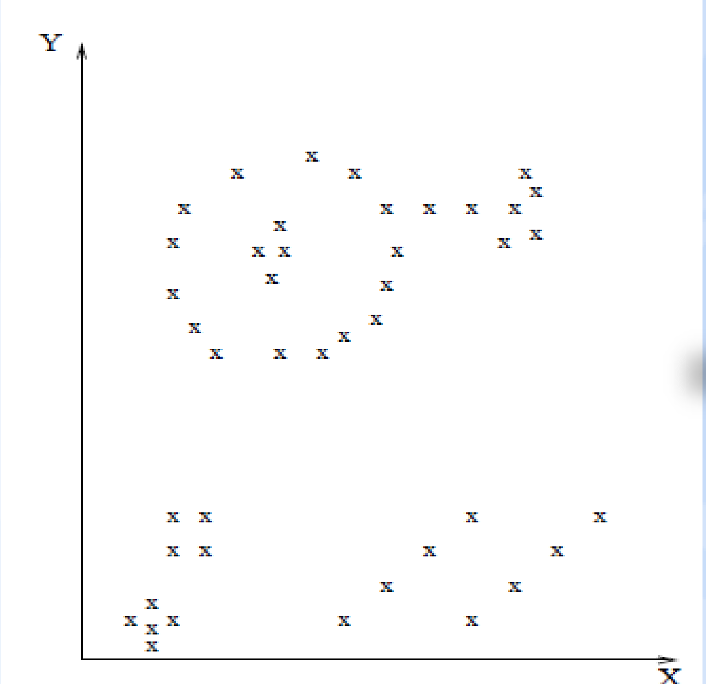
聚类分析是将数据集划分为多个类别的过程，聚类之后的每个类别中任意两个数据样本之间具有较高的相似度，而不同类别的数据样本之间具有较低的相似度。



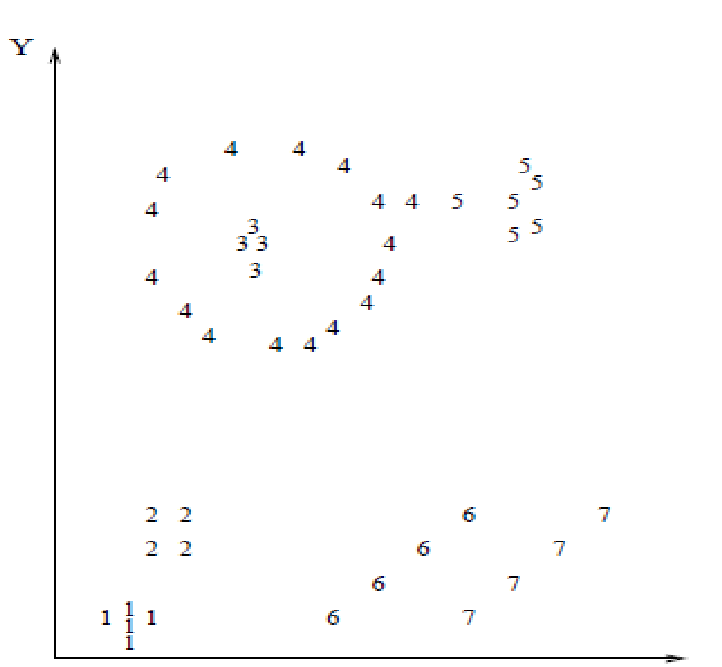
6.1 聚类问题概述

无监督算法

- 无标签
- 数据驱动



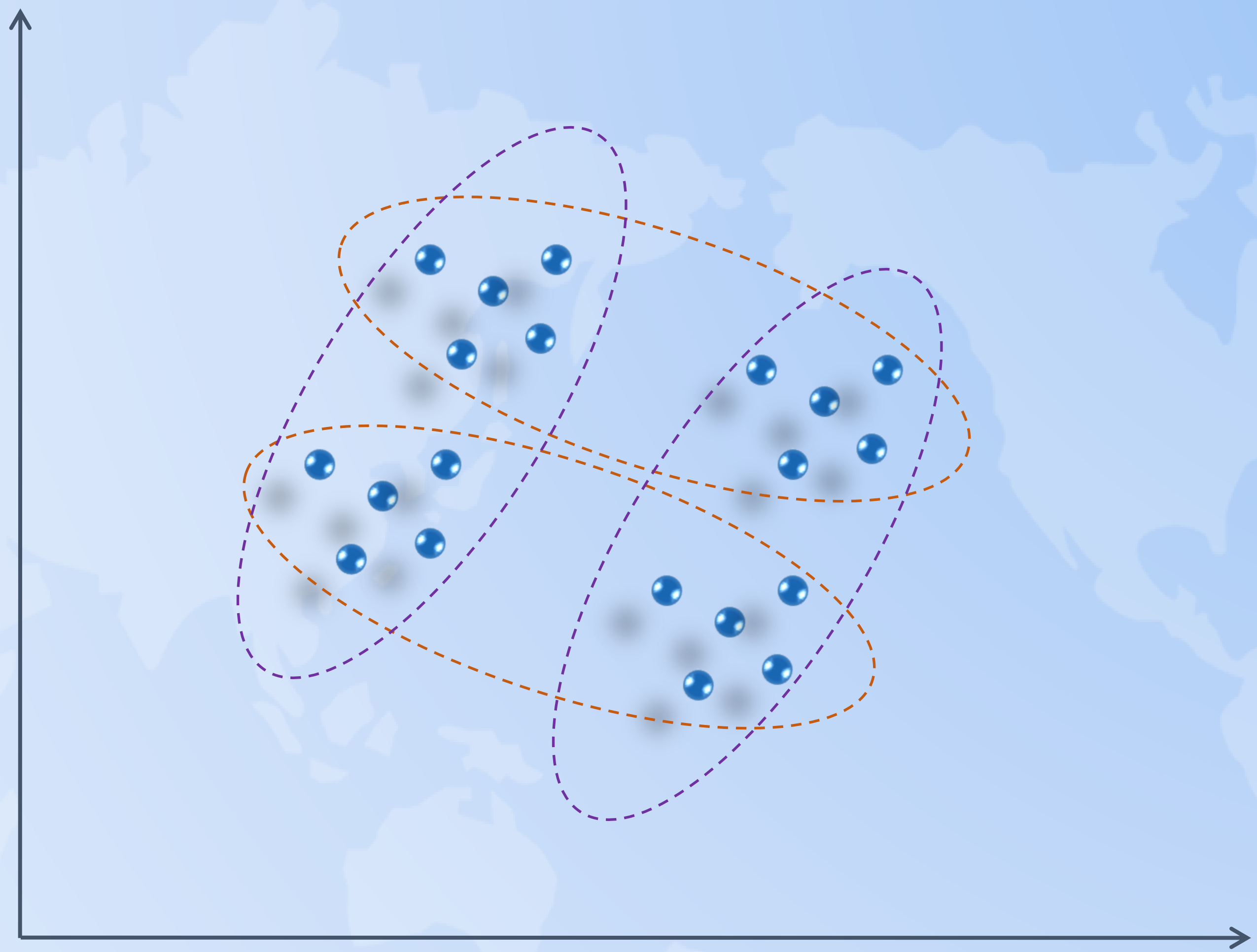
(a)



(b)

6.1 聚类问题概述

聚类是一种无监督的过程，聚类结果没有对错之分



应用场景

- 市场营销
  寻找行为相似的顾客群体
- 生物学
  发现具有相似特征的动物或植物群
- 生物信息学
  聚类基因和序列
- 地震研究
  对观测到的地震震中进行聚类，以确定危险区
- 万维网
  对博客数据进行聚类，以发现具有相似访问模式的组
- 社交网络
  发现有亲密友谊的个人群体

应用场景

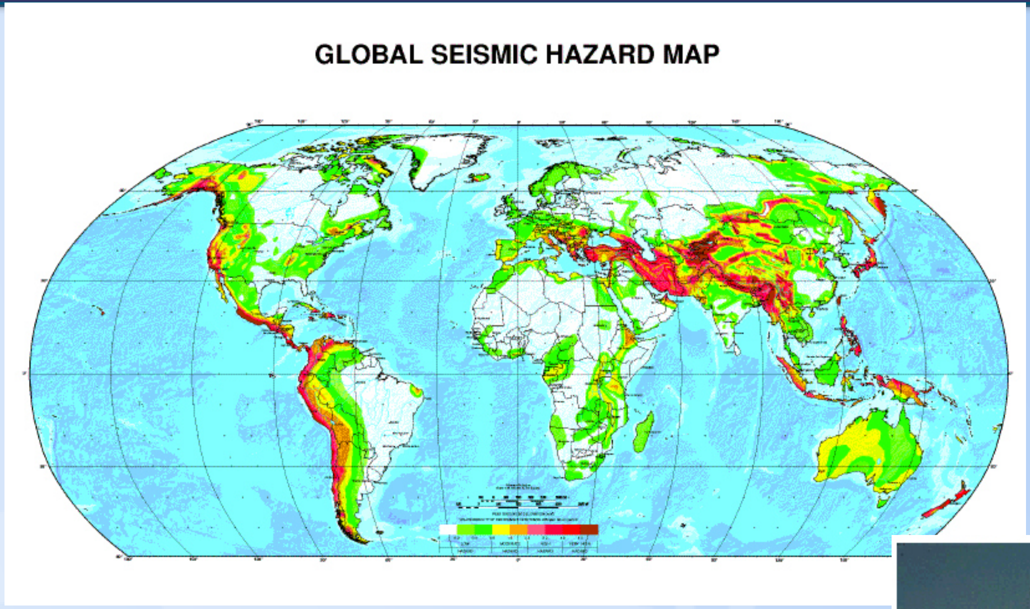
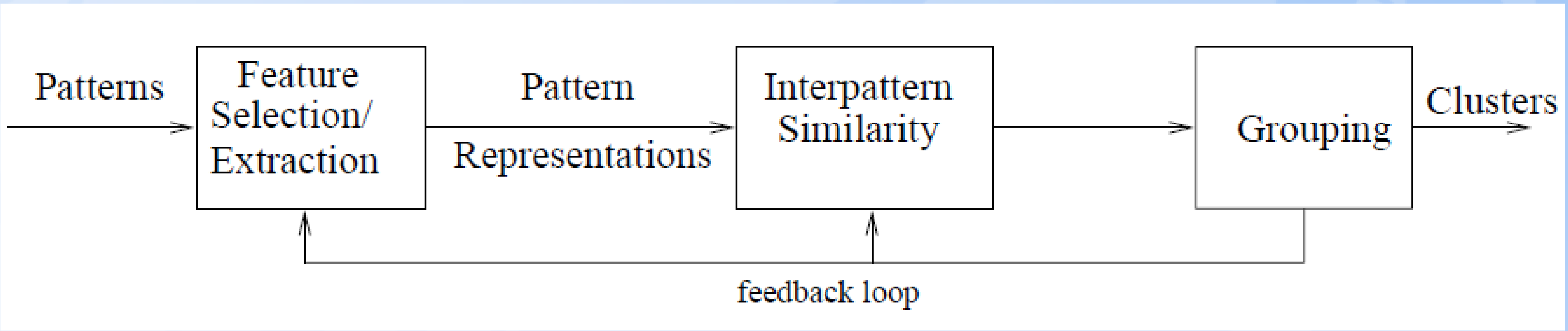


Image Segmentation

聚类流程

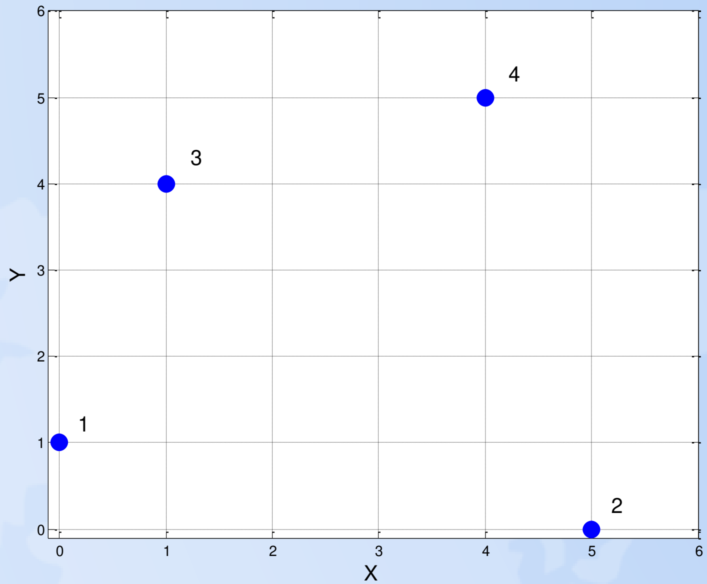


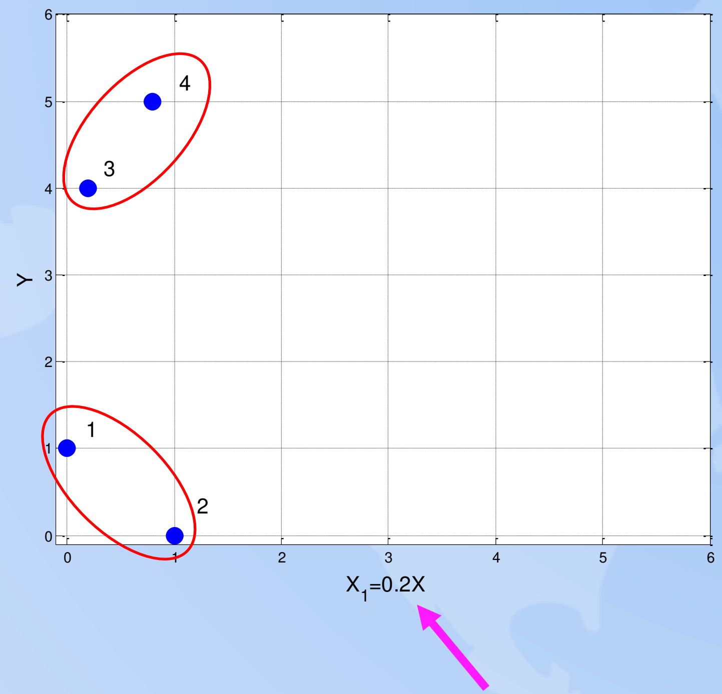
The Big Picture

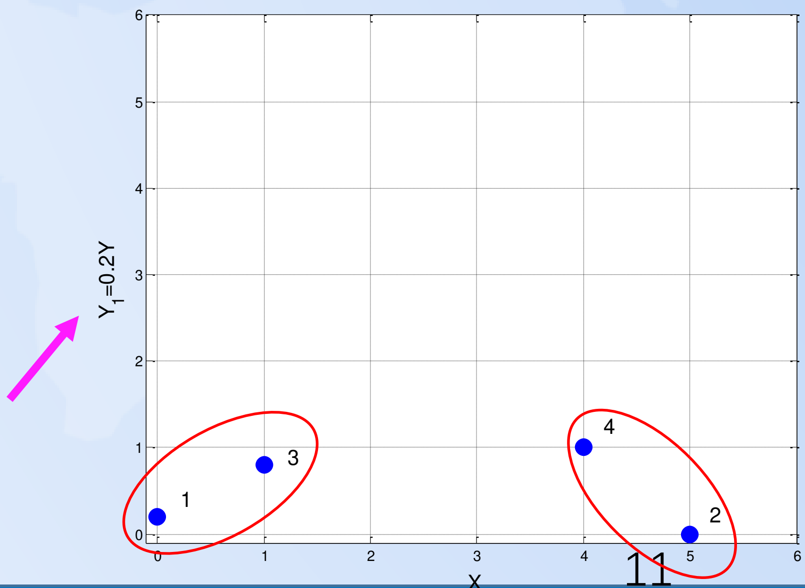
聚类算法的一般性要求

- 可伸缩性
- 处理不同类型属性的能力
- 发现任意形状聚类的能力
- 处理噪音数据的能力
- 对输入记录顺序不敏感
- 减少对先验知识和用户自定义参数的依耐性
- 可解释性和实用性

实际考虑

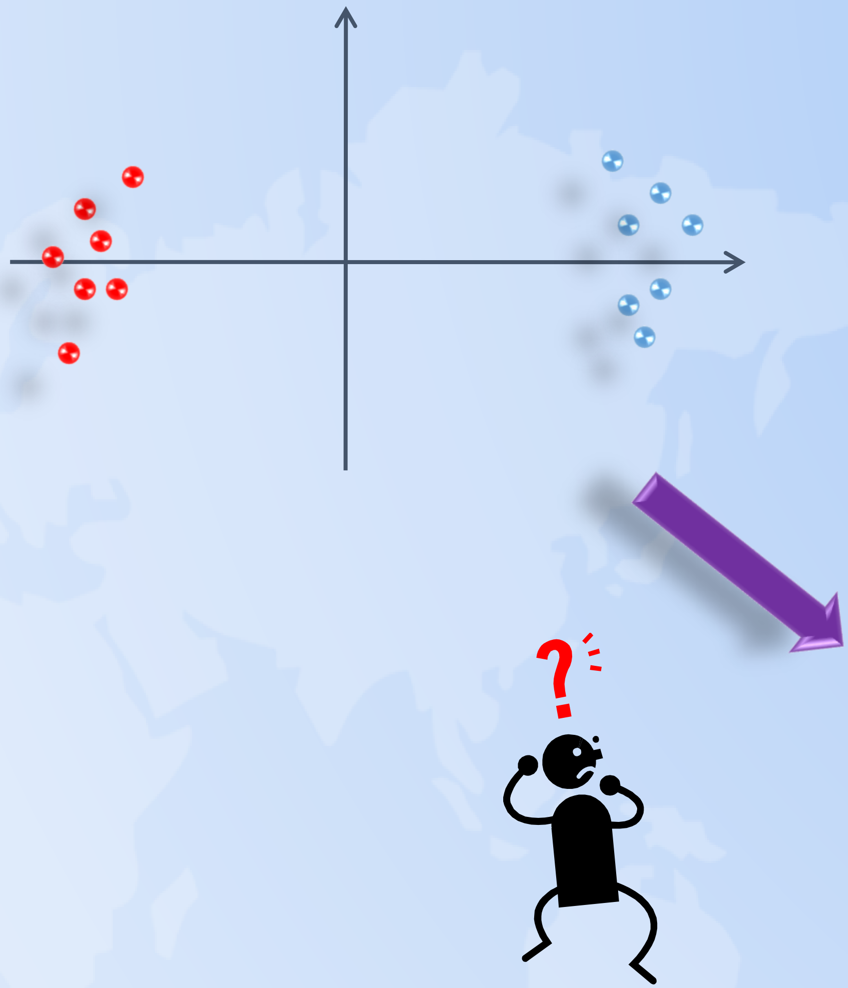




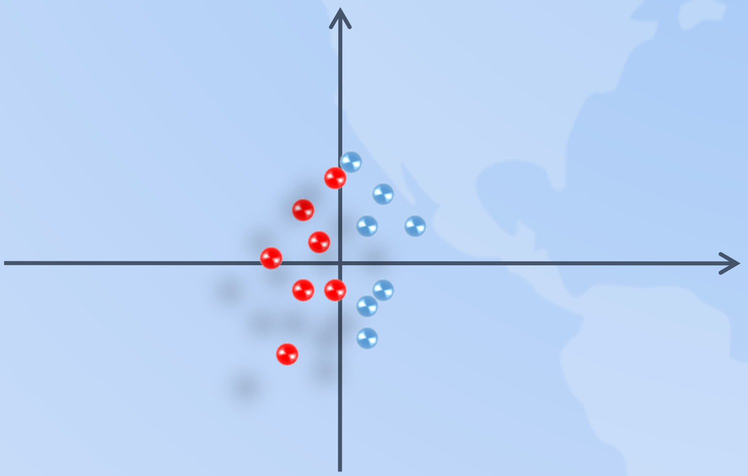


是否需要坐标缩放?

实际考虑



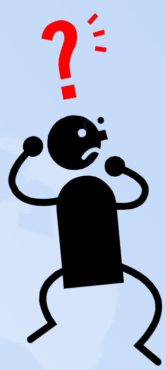
是否需要标准化?



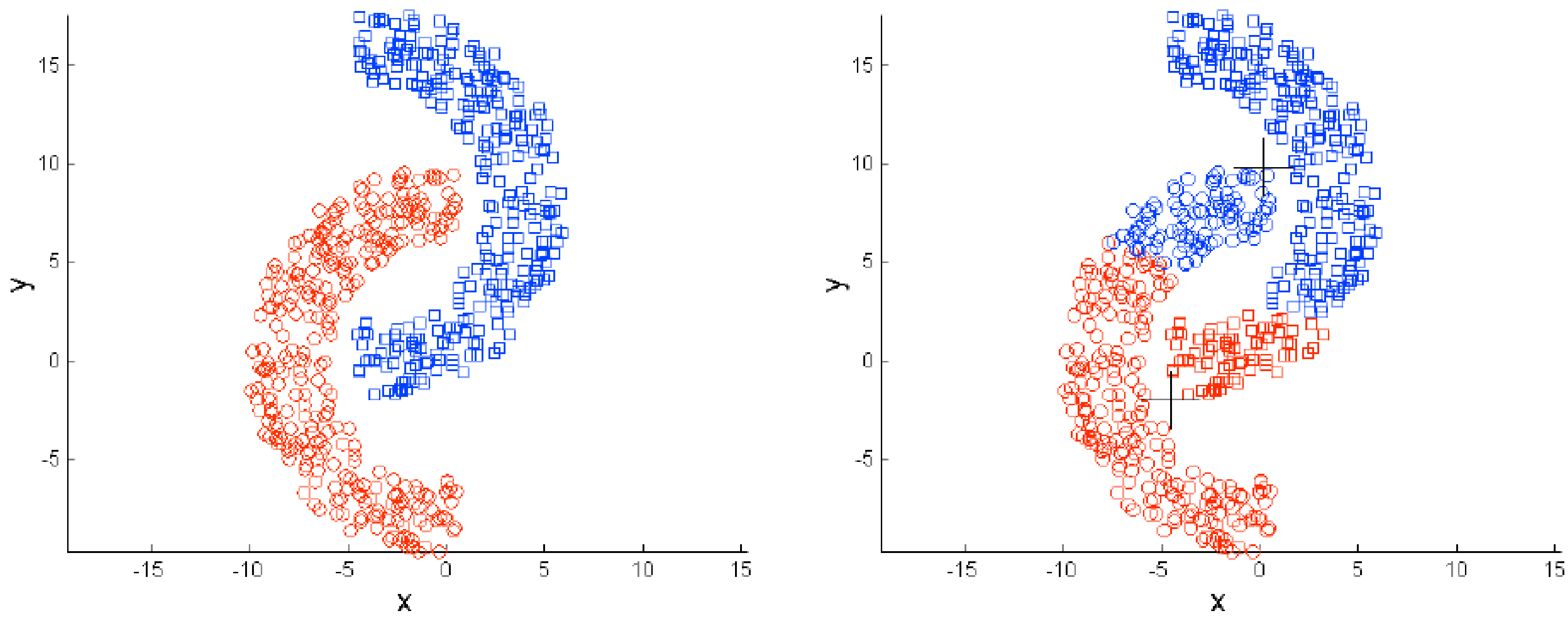
评价标准

$$J_e = \sum_{i=1}^{c} \sum_{x \in D_i} \left\| x - m_i \right\|^2, \qquad m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

VS.



评价标准



6.2 相似度计算方法

聚类分析示例数据集

聚类分析的数据集没有类别属性

样本序号	描述属性1	描述属性2
x_1	1	3
x_2	1	6.5
x_3	1.5	4
x_4	4.5	7.5
x_5	4	8.5
x_6	5.5	9
x_7	4.5	8

6.2.1 连续型属性的相似度计算方法

- 欧氏距离（Euclidean distance）

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d}(x_{ik} - x_{jk})^2}$$

- 曼哈顿距离（Manhattan distance）

$$d(x_i, x_j) = \sum_{k=1}^{d}\left|x_{ik} - x_{jk}\right|$$

- 明考斯基距离（Minkowski distance）

$$d(x_i, x_j) = \left(\sum_{k=1}^{d}\left|x_{ik} - x_{jk}\right|^q\right)^{1/q}$$

6.2.2 离散型属性的相似度计算方法

- 数据样本的二值离散型属性的取值情况

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">数据样本x_i</td><td></td></tr>
<tr><td>1</td><td>0</td><td>合计</td></tr>
<tr><td rowspan="2">数据样本x_j</td><td>1</td><td>a_{11}</td><td>a_{10}</td><td>$a_{11}+a_{10}$</td></tr>
<tr><td>0</td><td>a_{01}</td><td>a_{00}</td><td>$a_{01}+a_{00}$</td></tr>
<tr><td></td><td>合计</td><td>$a_{11}+a_{01}$</td><td>$a_{10}+a_{00}$</td><td>$a_{11}+a_{10}+a_{01}+a_{00}$</td></tr>
</table>

二值离散型属性是指只有两种取值的离散型属性

6.2.2 离散型属性的相似度计算方法

- 对称的二值离散型属性

$$d(x_i, x_j) = \frac{a_{10} + a_{01}}{a_{11} + a_{10} + a_{01} + a_{00}}$$

- 不对称的二值离散型属性

$$d(x_i, x_j) = \frac{a_{10} + a_{01}}{a_{11} + a_{10} + a_{01}}$$

6.2.3 多值离散型属性的相似度计算方法

- 多值离散型属性的相似度

$$d(x_i, x_j) = \frac{d - u}{d}$$

- d为数据集中的属性个数，u为样本x_i和x_j取值相同的属性个数

6.2.4 混合类型属性的相似度计算方法

对于包含混合类型属性的数据集的相似度通常有两种计算方法：

- 将属性按照类型分组，每个新的数据集中只包含一种类型的属性；之后对每个数据集进行单独的聚类分析
- 把混合类型的属性放在一起处理，进行一次聚类分析

6.3 k-means聚类算法

- k-means是典型的基于距离的聚类算法，最初来自于信号处理的一种矢量量化方法，现被广泛应用于数据挖掘。
- k-means聚类的目的是将n个观测值划分为k个类，使每个类中的观测值距离该类的中心（类均值）比距离其他类中心都近。
- Reference：

  J. MacQueen (1967): "Some Methods for Classification and Analysis of Multivariate Observations", *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol.1, pp. 281-297.

  詹姆斯·麦奎恩（James MacQueen）1967年第一次使用"k-means"这个术语

6.3 k-means聚类算法

k-means算法的思想介绍

输入： 数据集X，聚类个数k

输出： 使误差平方和准则函数最小的k个聚类。

1. 选定某种距离作为数据样本件的相似性度量

由于k-means算法不适合处理离散型数据，因此在计算个样本距离时，可以根据实际需要选择欧氏距离、曼哈顿距离或者明考斯基距离中的一种来作为算法的相似性度量。

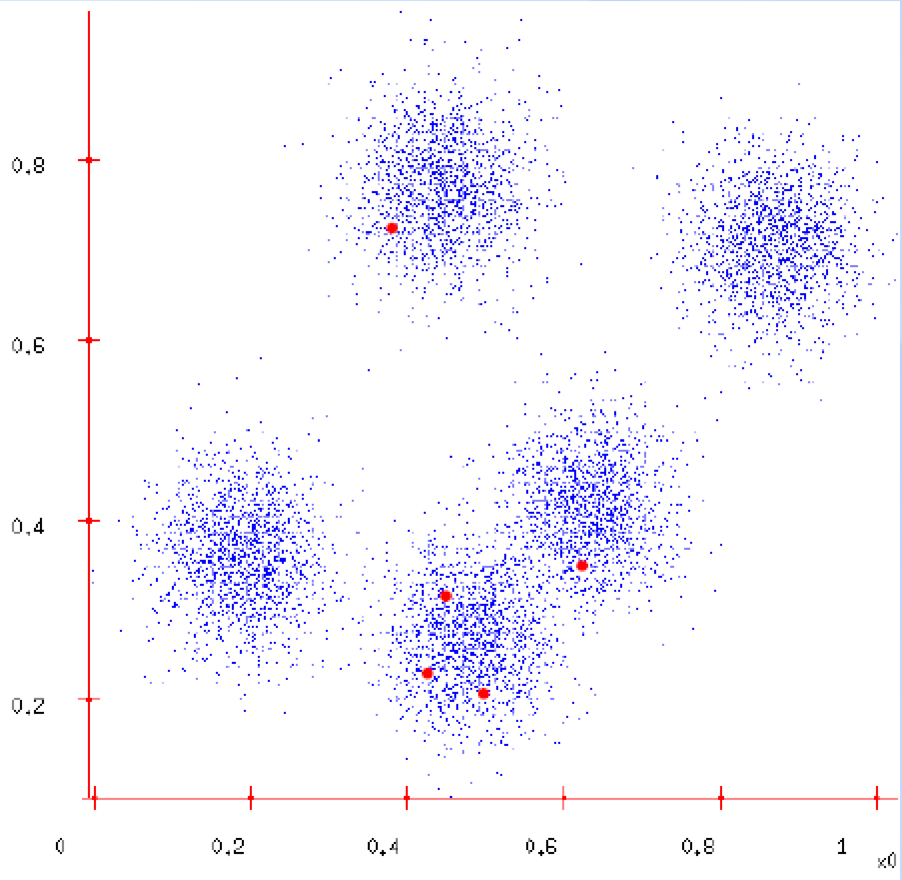
2. 选择评价聚类性能的准则函数

k-means算法使用误差平方和准则函数来评价聚类性能。

3. 根据一个簇中对象的平均值来计算相似度

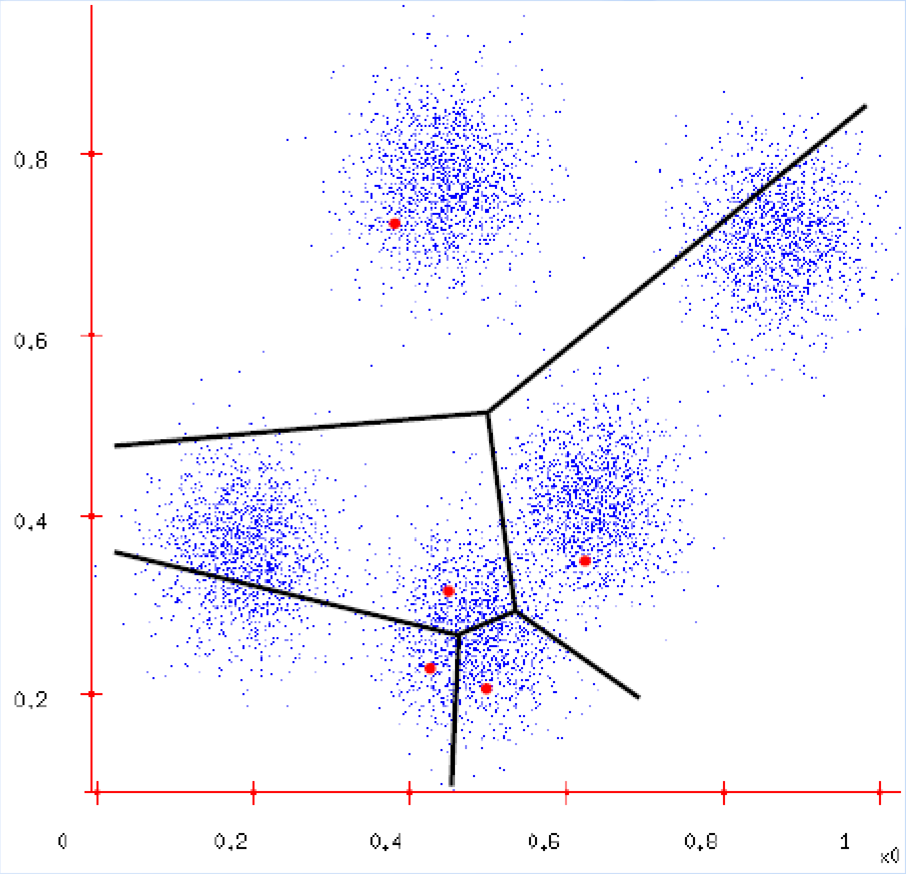
6.3 k-means聚类算法

（1）从数据集X中随机地选择k个数据样本作为聚类的初始代表点，每一个代表点表示一个类别。



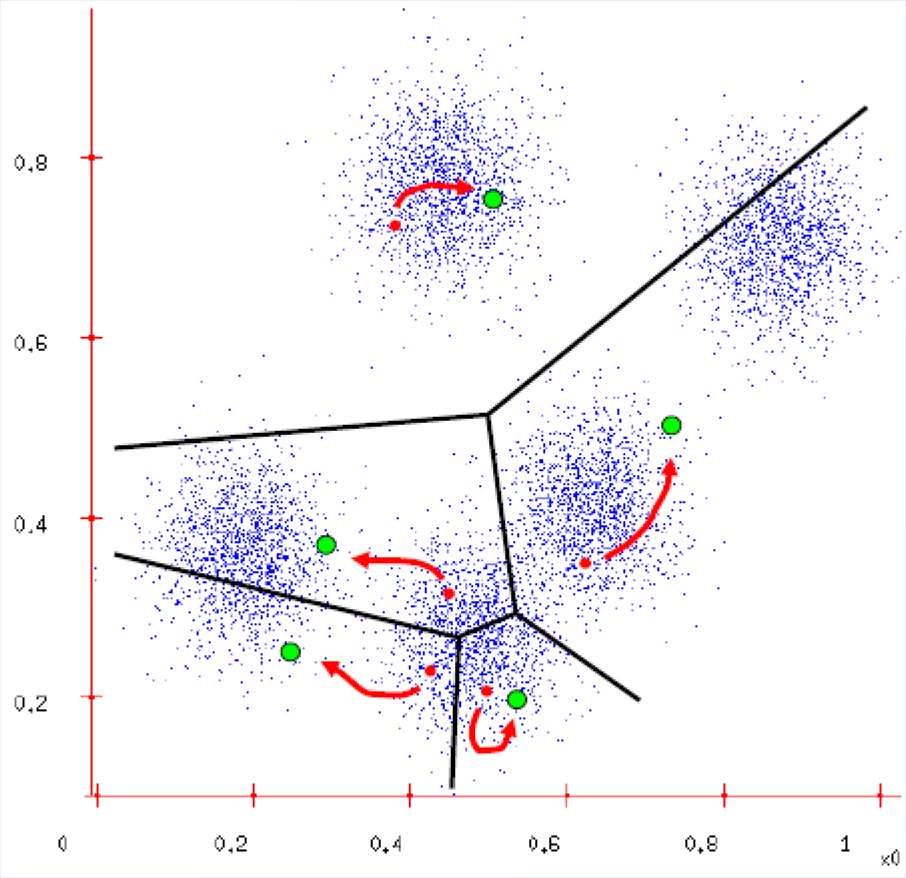
6.3 k-means聚类算法

（2）对于X中任一数据样本x，计算它与k个初始代表点的距离，并且将它划分到距离最近的初始代表点所表示的类别中。



6.3 k-means聚类算法

（3）完成数据样本的划分之后，对于每一个聚类，计算其中所有数据样本的均值，并且将其作为该聚类的新的代表点，由此得到k个均值代表点。



重复步骤(2)和(3)，直到各个聚类不再发生变化为止，即误差平方和准则函数的值达到最优。

6.3 k-means聚类算法

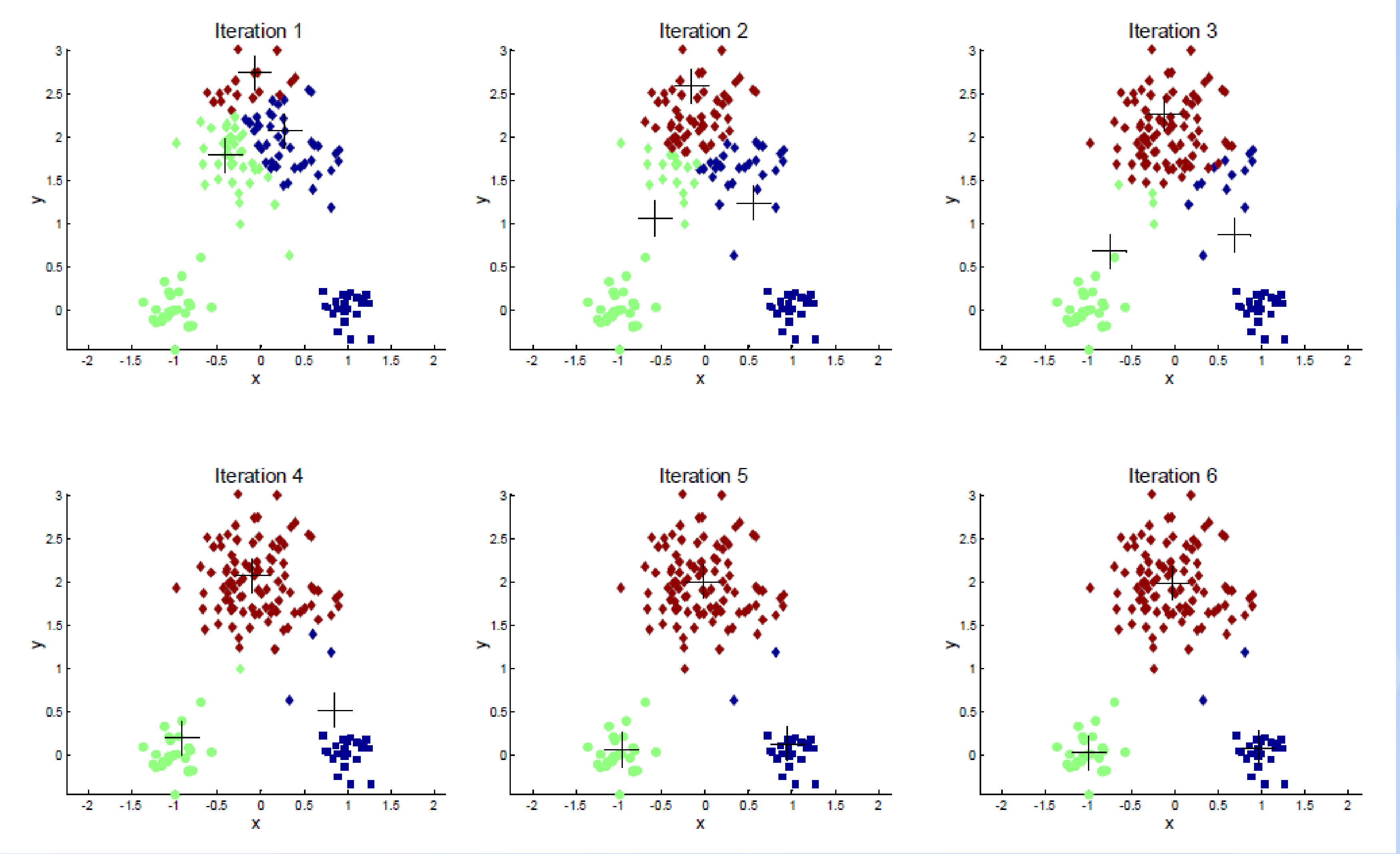
Pros：

k-means擅长处理球状分布的数据，当结果聚类是密集的，而且类和类之间的区别比较明显时，K均值的效果比较好。

Cons：

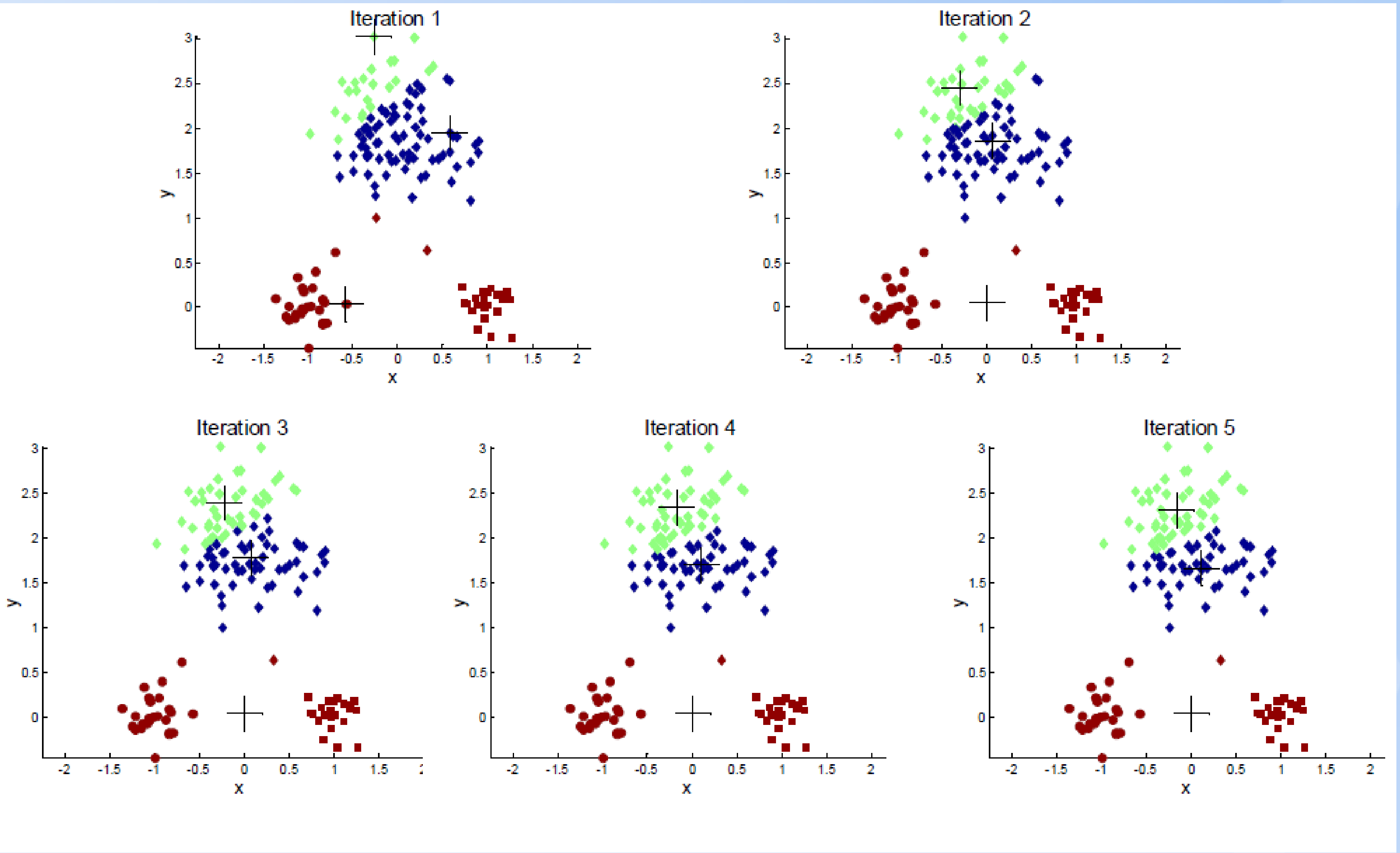
- 随机选取中心点有可能导致迭代次数很大或者限于某个局部最优状态
- 算法时间复杂度比较高 O(nkt)
- 不能发现非凸形状的簇
- 需要事先确定超参数K
- 对噪声和离群点敏感

随机选取中心点的影响



Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
x
y

随机选取中心点的影响

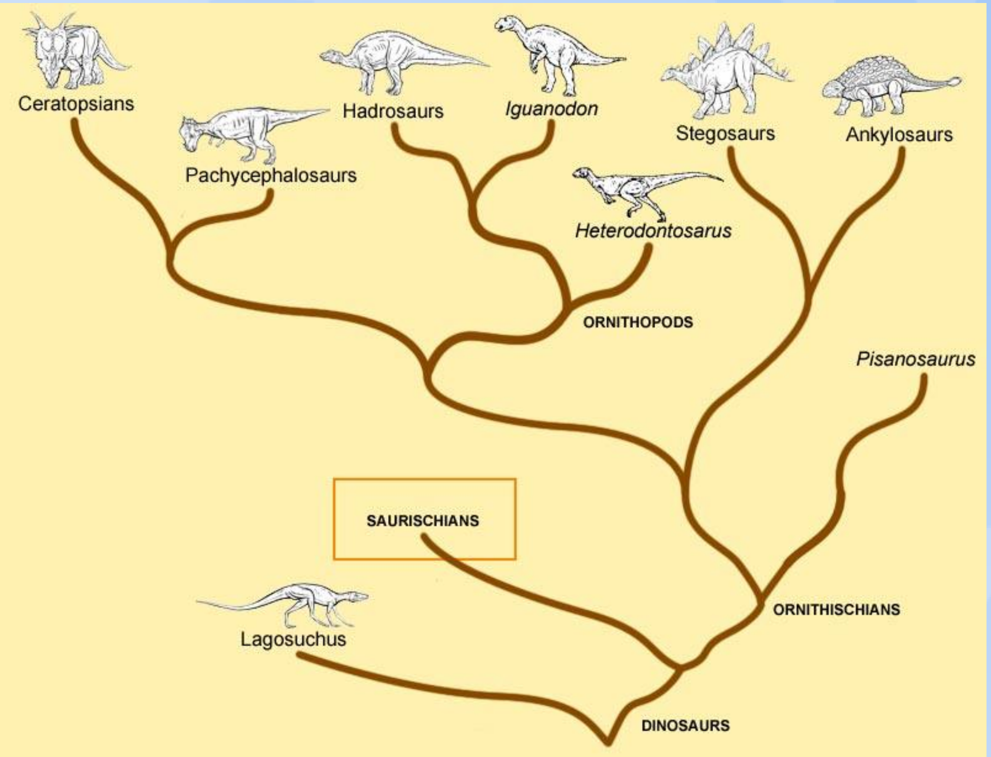

Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
x
y

6.4 层次聚类方法

- 层次聚类，顾名思义就是要一层一层地进行聚类，可以从下而上地把小的簇合并聚集，也可以从上而下地将大的簇进行分割。

- 绝大多数层次聚类属于凝聚型层次聚类，它们只是在簇间相似度的定义上有所不同。

- 如何判断两个簇之间的相似度呢？



6.4 层次聚类方法

层次聚类方法最常用的相似性度量有：

- 最小距离 $$d_{min}(X_i, X_j) = \min_{p \in X_i, p' \in X_j} d(p, p')$$
- 最大距离 $$d_{max}(X_i, X_j) = \max_{p \in X_i, p' \in X_j} d(p, p')$$
- 均值距离 $$d_{mean}(X_i, X_j) = d(m_i, m_j)$$
- 平均距离 $$d_{avg}(X_i, X_j) = \frac{1}{n_i n_j} \sum_{p \in X_i} \sum_{p' \in X_j} d(p, p')$$

6.4 层次聚类方法

层次聚类的流程

这里给出采用最小距离的凝聚层次聚类算法流程：

(1) 将每个对象看作一类，计算两两之间的最小距离；

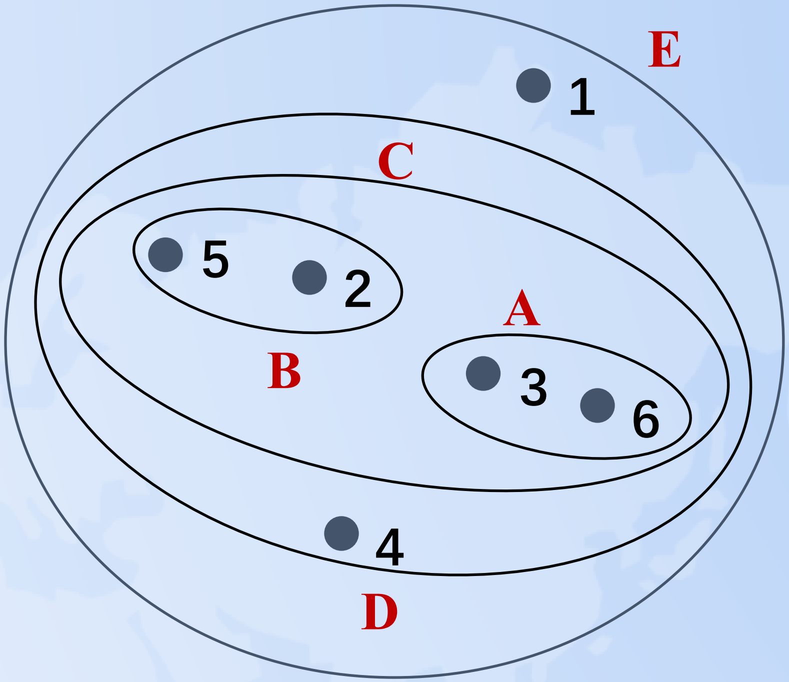
(2) 将距离最小的两个类合并成一个新类；

(3) 重新计算新类与所有类之间的距离；

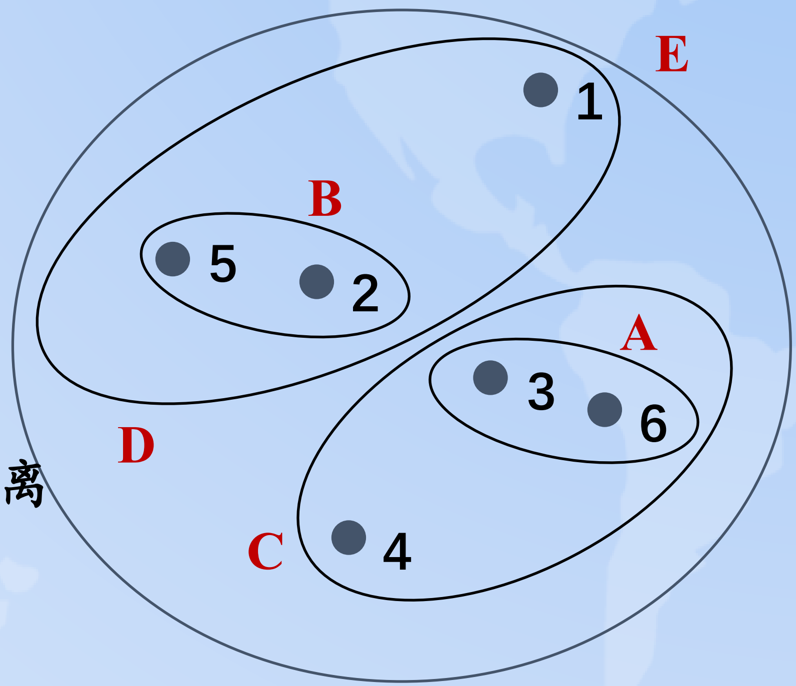
(4) 重复(2)、(3)，直到所有类最后合并成一类。

凝聚的层次聚类并没有类似基本K均值的全局目标函数，没有局部极小问题或是很难选择初始点的问题。合并的操作往往是最终的，一旦合并两个簇之后就不会撤销。当然其计算存储的代价是昂贵的。

6.4 层次聚类方法



最小距离



最大距离

6.4 层次聚类方法——案例

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

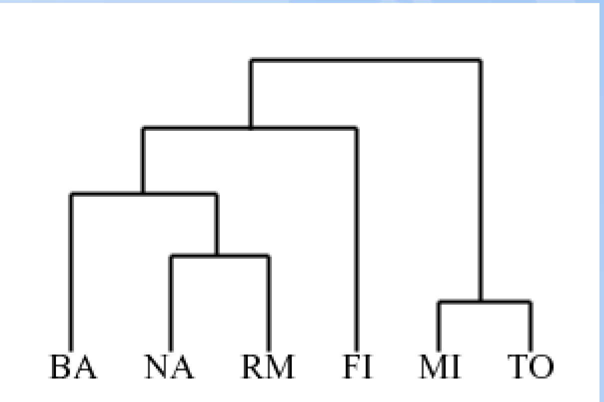
6.4 层次聚类方法——案例

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

6.4 层次聚类方法——案例

	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

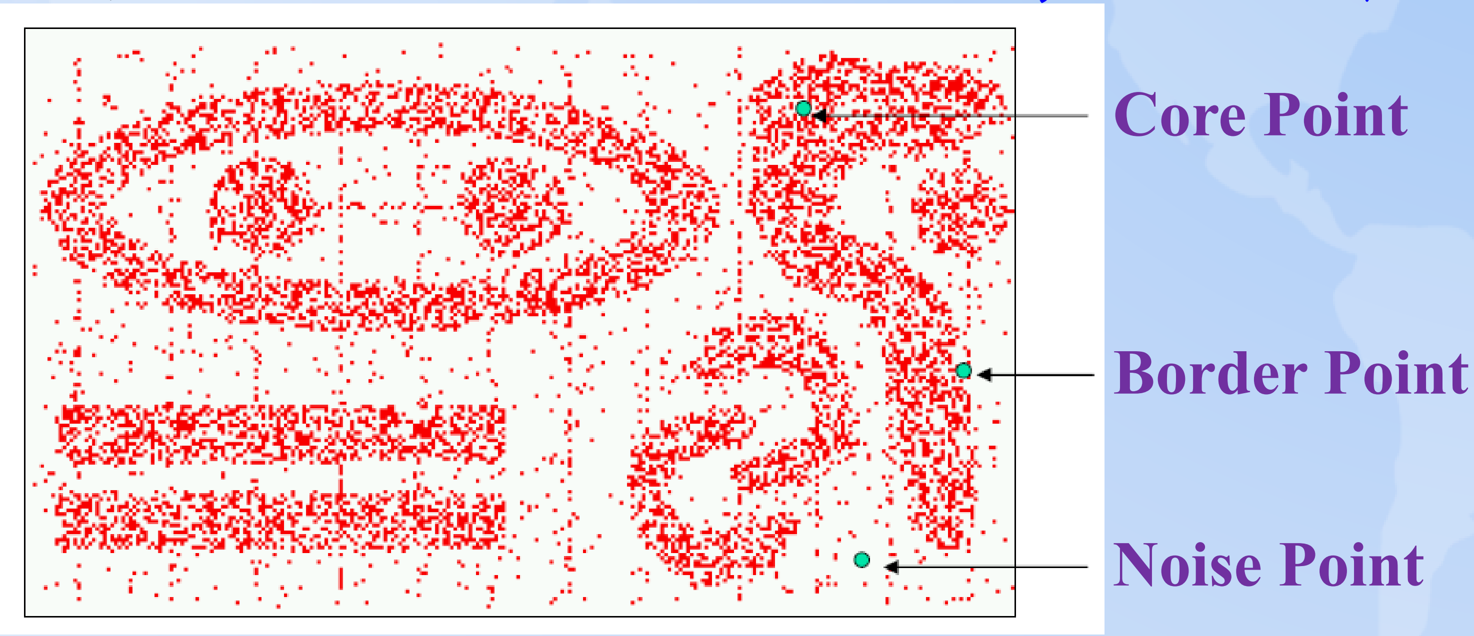


	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0

6.5 基于密度的聚类方法

由于划分式聚类和层次聚类算法算往往只能发现凸形的聚类簇。为了弥补这一缺陷，发现各种任意形状的聚类簇，开发出基于密度的聚类算法。

这类算法认为，在整个样本空间点中，各目标类簇是由一群的稠密样本点组成的，而这些稠密样本点被低密度区域(噪声)分割，而算法的目的就是要过滤低密度区域，发现稠密样本。



6.5 基于密度的聚类方法

DBSCAN（Ester, 1996）是基于密度的聚类方法中最典型的代表算法之一。其核心思想就是先发现密度较高的点，然后把相近的高密度点逐步都连成一片，进而生成各种簇。