机器学习

Machine learning

第六章 聚类分析

Clustering

授课人：周晓飞
zhouxiaofei@iie.ac.cn
2022-10-28

第六章 聚类分析

6.1 概述

6.2 序贯方法

6.3 层次聚类

6.4 K 均值聚类

第六章 聚类分析

6.1 概述

6.2 序贯方法

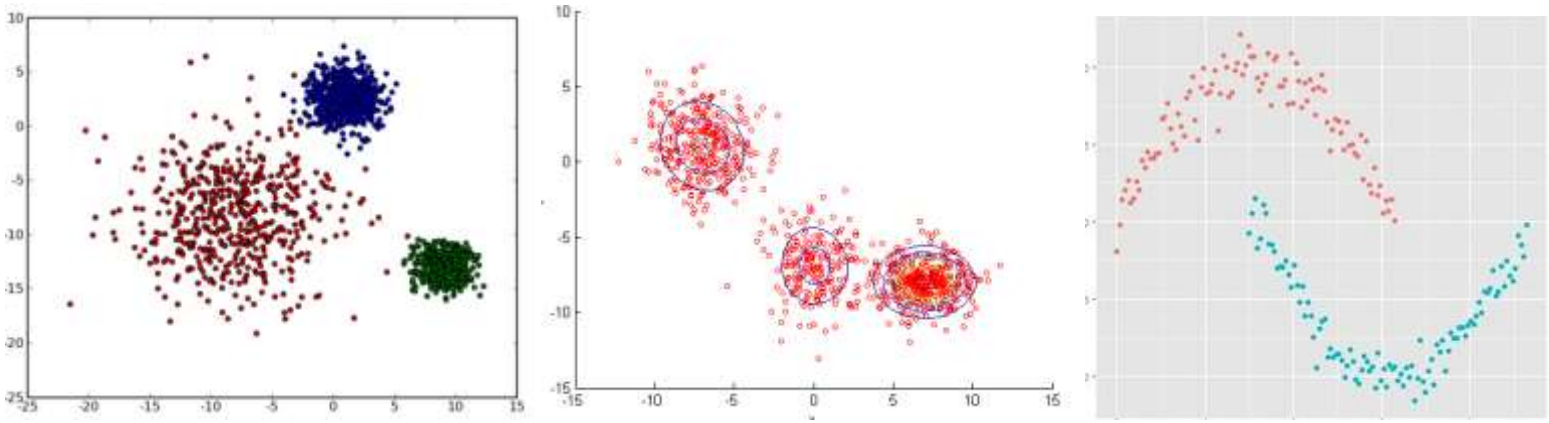
6.3 层次聚类

6.4 K 均值聚类

概述

聚类问题

- 聚类是无监督机器学习问题；
- 目标：感知样本间的相似度，进行类别归纳；
- 聚类研究的重要应用：(1)潜在类别预测，(2)数据压缩
- 既可以作为一个单独过程，用于寻找数据内在的分布结构，
- 也可以作为分类、稀疏表示等其他学习任务的前驱过程。



概述

聚类问题

聚类分析，在不同的应用学科有不同的称呼

Unsupervised learning (machine learning, pattern recognition)

numerical taxonomy (in biology, ecology)

typology (in social sciences)

partition (in graph theory)

聚类问题

聚类算法法的种类

- Sequential algorithms
- Hierarchical clustering algorithms
- based on cost function optimization

  --K-means

  --Probabilistic clustering algorithms

  --Fuzzy clustering algorithms
- Density-based clustering
- Other:

  --Genetic clustering algorithms --Branch and bound clustering algorithms

  --Subspace clustering algorithms --Kernel-based methods

概述

聚类问题

聚类划分：

样本集 $X=\{x_1,x_2,...,x_N\}$ 的 m-clustering 划分 $C_1,C_2,...,C_m$ 满足以下三个条件：

(1) $C_i \neq \phi, i=1,...,m$

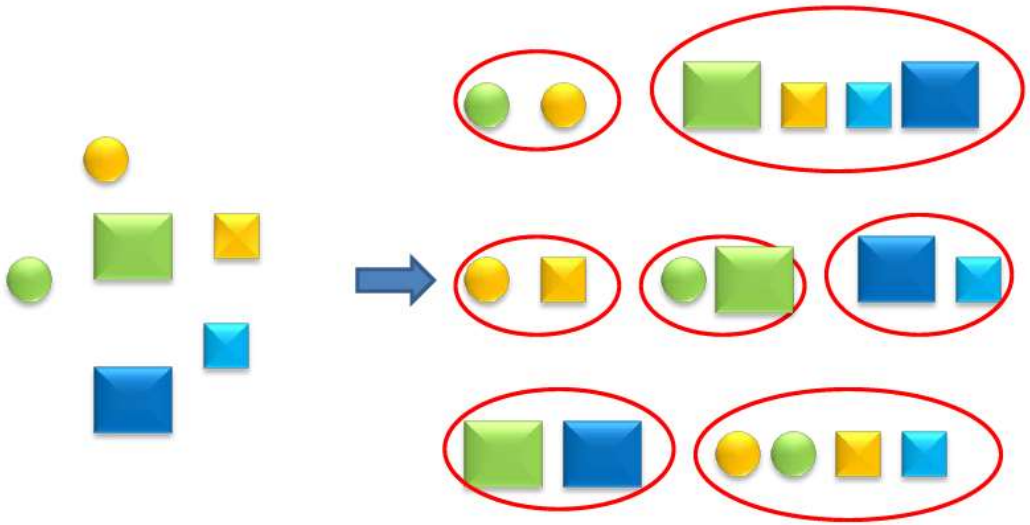
(2) $\bigcup_{i=1,..m} C_i = X$

(3) $C_i \bigcap C_j = \phi, \ i \neq j, \ i,j=1,...,m$

概述

聚类问题

影响聚类结果的因素



（1）属性选择导致不同结果；

（2）相似性度量是判断样本间、类别间的相似的标准；

（3）聚类规则是样本聚集条件，例如，近邻、损失函数。

相似性度量

常用到的相似性度量

（1） 样本---样本；
（2） 样本---集合；
（3） 集合---集合（类间距离）；
（4） 集合内样本间距离（类内距离）；

相似性度量

样本--样本

3.3 的向量相似性

- $d_p(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{l} w_i \left| x_i - y_i \right|^p \right)^{1/p}$
- $s_{\cos ine}(\boldsymbol{x},\boldsymbol{y}) = \dfrac{\boldsymbol{x}^T \boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$
- $r_{Pearson}(\boldsymbol{x},\boldsymbol{y}) = \dfrac{\boldsymbol{x}_d{}^T \boldsymbol{y}_d}{\|\boldsymbol{x}_d\|\|\boldsymbol{y}_d\|} \quad \boldsymbol{x}_d = [x_1 - \bar{x},\ldots,x_l - \bar{x}]^T \quad \boldsymbol{y}_d = [y_1 - \bar{y},\ldots,y_l - \bar{y}]^T$
- $s_T(\boldsymbol{x},\boldsymbol{y}) = \dfrac{1}{1 + \dfrac{(\boldsymbol{x}-\boldsymbol{y})^T(\boldsymbol{x}-\boldsymbol{y})}{\boldsymbol{x}^T\boldsymbol{y}}}$

相似性度量

样本---集合

（1）集合为离散点集：

- 到集合最远点距离

$$d(x,C)=\max_{y\in C} d(x,y)$$

- 到集合最近点距离

$$d(x,C)=\min_{y\in C} d(x,y)$$

- 到集合平均点距离

$$d(x,C)=\frac{1}{|C|}\sum_{y\in C} d(x,y)$$

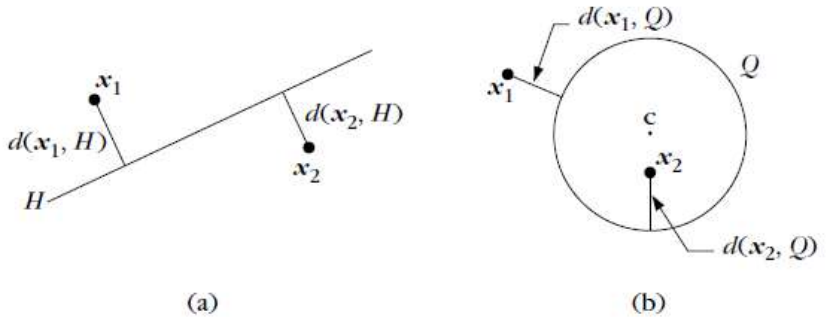
（2）集合为连续区域

- 集合为平面

$$d(x,H)=\min_{z\in H} d(x,z)$$

- 集合为圆

$$d(x,Q)=\min_{z\in Q} d(x,z)$$



相似性度量

集合---集合（类间距离）

- 集合间最远点距离

$$d_{\max}(C_i,C_j)=\max_{x\in C_i,y\in C_j} d(\boldsymbol{x},\boldsymbol{y})$$

- 集合间最近点距离

$$d_{\min}(C_i,C_j)=\min_{x\in C_i,y\in C_j} d(\boldsymbol{x},\boldsymbol{y})$$

- 集合间所有点平均距离

$$d_{avg}^{ss}(C_i,C_j)=\frac{1}{|C_i||C_j|}\sum_{x\in C_i,y\in C_j} d(\boldsymbol{x},\boldsymbol{y})$$

- 集合表征点间距离（如平均值）

$$d_{cen}(C_i,C_j)=d(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j)$$

$\boldsymbol{\mu}$ 代表簇 C 的中心点 $\boldsymbol{\mu}=\frac{1}{|C|}\sum_{1\leqslant i\leqslant|C|}\boldsymbol{x}_i$

相似性度量

集合内样本间距离（类内距离）

$$\mathrm{avg}(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \leqslant i < j \leqslant |C|} \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\mathrm{diam}(C) = \max_{1 \leqslant i < j \leqslant |C|} \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

性能度量

聚类性能的外部指标

指通过已知类簇划分，对聚类结果进行评价；判别同类别样本对标签一致与否，避免相同类簇划分，不同标签名称导致的不一致。

对数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_m\}$，假定通过聚类给出的簇划分为 $C=\{C_1,C_2,...,C_k\}$，参考模型给出的簇划分为 $C^*=\{C_1^*,C_2^*,...,C_s^*\}$。相应地，令 $\boldsymbol{\lambda}$ 与 $\boldsymbol{\lambda}^*$ 分别表示与 C 和 C^* 对应的簇标记向量。我们将样本两两配对考虑，定义

$$a=|SS|, SS=\{(\boldsymbol{x}_i,\boldsymbol{x}_j)\mid \lambda_i=\lambda_j, \lambda_i^*=\lambda_j^*, i<j\},$$
$$b=|SD|, SD=\{(\boldsymbol{x}_i,\boldsymbol{x}_j)\mid \lambda_i=\lambda_j, \lambda_i^*\neq\lambda_j^*, i<j\},$$
$$c=|DS|, DS=\{(\boldsymbol{x}_i,\boldsymbol{x}_j)\mid \lambda_i\neq\lambda_j, \lambda_i^*=\lambda_j^*, i<j\},$$
$$d=|DD|, DD=\{(\boldsymbol{x}_i,\boldsymbol{x}_j)\mid \lambda_i\neq\lambda_j, \lambda_i^*\neq\lambda_j^*, i<j\},$$

每个样本对 $(\boldsymbol{x}_i,\boldsymbol{x}_j)(i<j)$ 仅能出现在一个集合中，因此有 $a+b+c+d=m(m-1)/2$ 成立。

性能度量

- Jaccard 系数(Jaccard Coefficient, 简称 JC)

$$\mathrm{JC} = \frac{a}{a+b+c}.$$

- FM 指数(Fowlkes and Mallows Index, 简称 FMI)

$$\mathrm{FMI} = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}.$$

- Rand 指数(Rand Index, 简称 RI)

$$\mathrm{RI} = \frac{2(a+d)}{m(m-1)}.$$

上述性能度量的结果值均在 $[0,1]$ 区间, 值越大越好.

性能度量

聚类性能的内部指标

没有已知的类簇划分进行参考，通过聚类具有的类内相似和类间相异的特点进行评价。

- DB 指数(Davies-Bouldin Index, 简称 DBI)

$$\mathrm{DBI} = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\mathrm{avg}(C_i)+\mathrm{avg}(C_j)}{d_{\mathrm{cen}}(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j)}\right) .$$

- Dunn 指数(Dunn Index, 简称 DI)

$$\mathrm{DI} = \min_{1\leqslant i\leqslant k}\left\{\min_{j\neq i}\left(\frac{d_{\min}(C_i,C_j)}{\max_{1\leqslant l\leqslant k}\mathrm{diam}(C_l)}\right)\right\} .$$

DBI 的值越小越好, 而 DI 则相反, 值越大越好.

概述

本章内容

本章学习聚类分析方法：

- 序贯方法
- 层次聚类
- K-均值聚类

第六章 聚类分析

6.1 概述

6.2 序贯方法

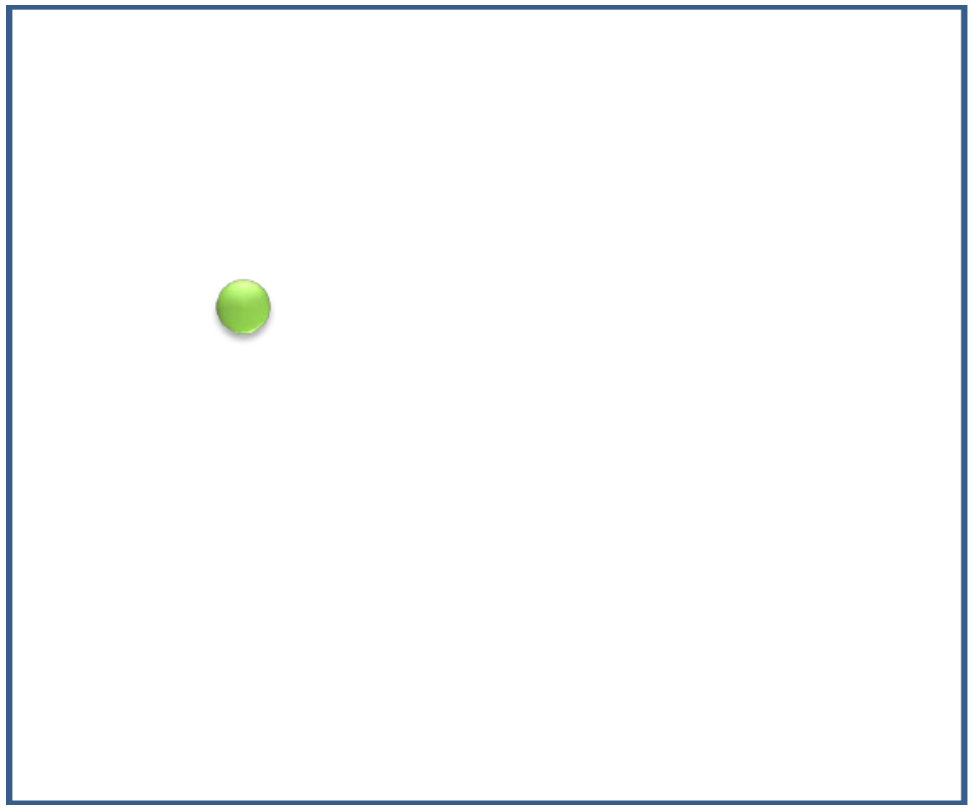
6.3 层次聚类

6.4 K 均值聚类

序贯方法

基本思想

逐一比较单个样本与类簇的相似性，有相似类则归类，无相似类则建立新类。



- **优点：**

一种简单的，快速算法。

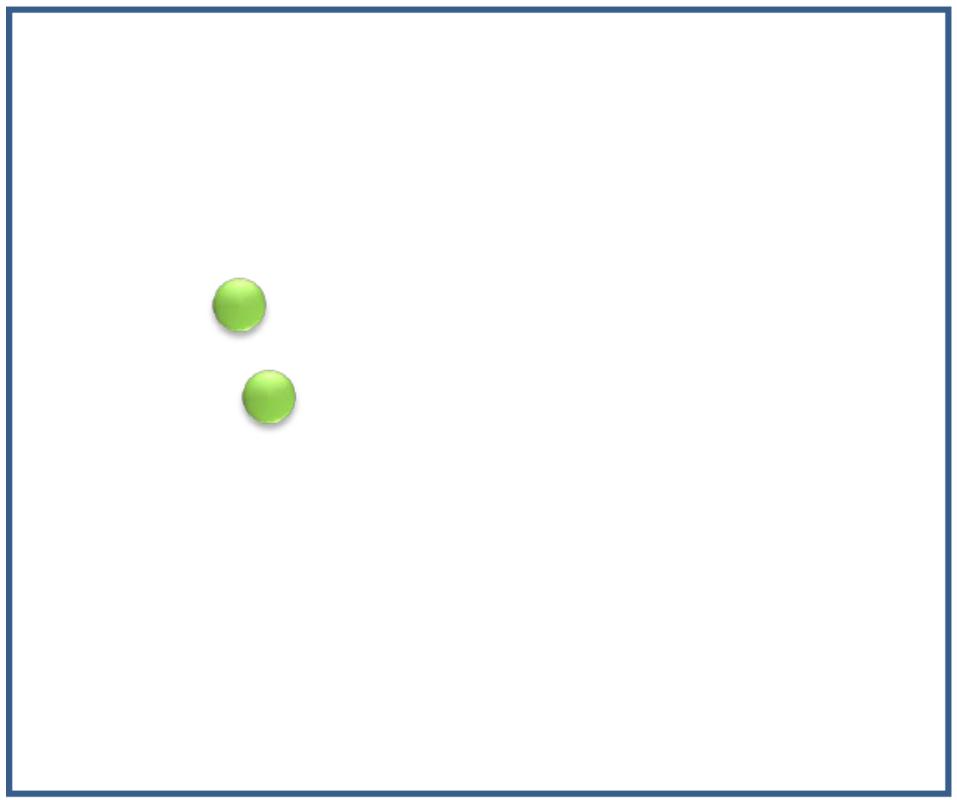
- **相似性的关键度量：**

类别相似性：样本---类簇（样本---集合）。

序贯方法

基本思想

逐一比较单个样本与类簇的相似性，有相似类则归类，无相似类则建立新类。



- **优点：**

一种简单的，快速算法。

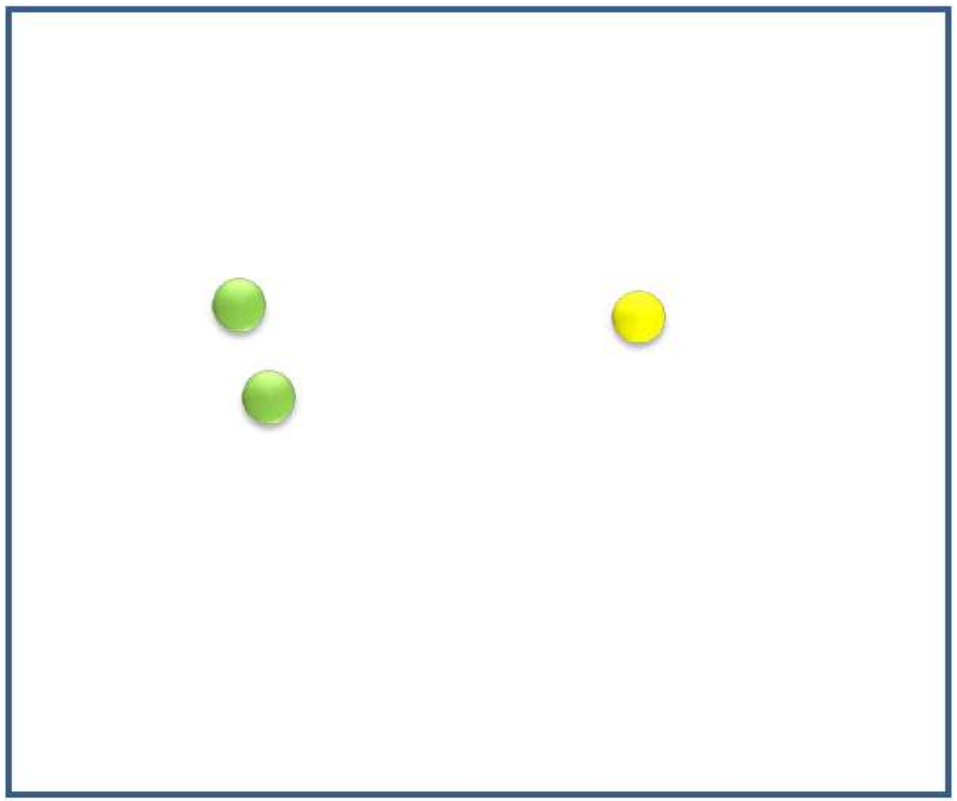
- **相似性的关键度量：**

类别相似性：样本---类簇（样本---集合）。

序贯方法

基本思想

逐一比较单个样本与类簇的相似性，有相似类则归类，无相似类则建立新类。



- **优点：**

一种简单的，快速算法。

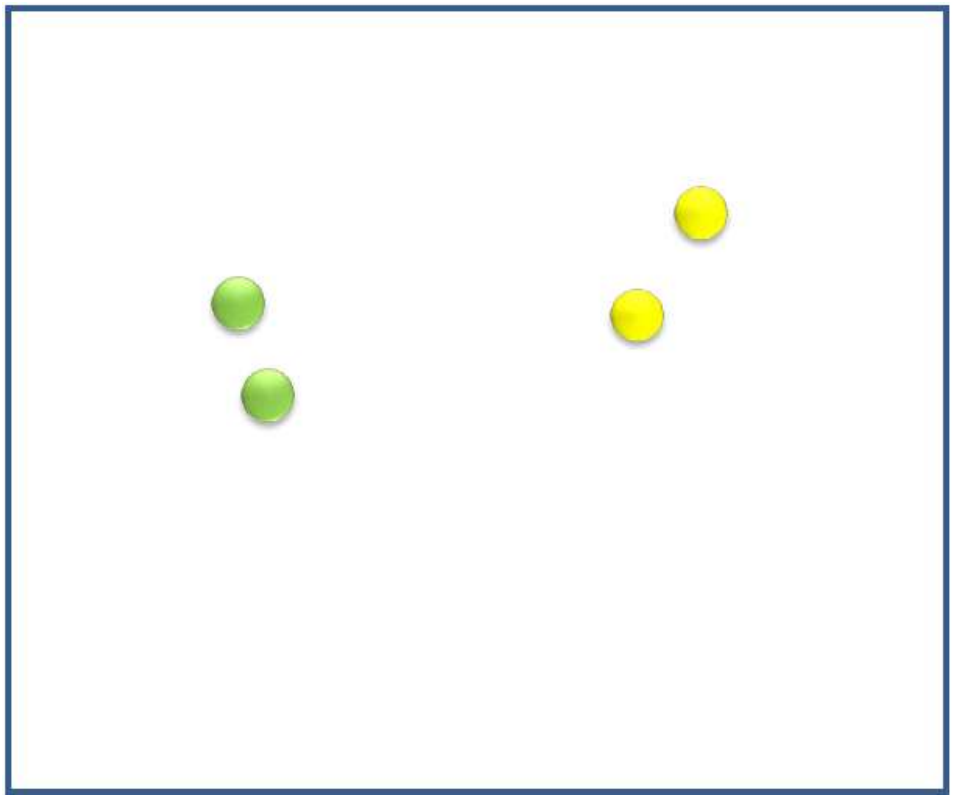
- **相似性的关键度量：**

类别相似性：样本---类簇（样本---集合）。

序贯方法

基本思想

逐一比较单个样本与类簇的相似性，有相似类则归类，无相似类则建立新类。



- **优点：**

一种简单的，快速算法。

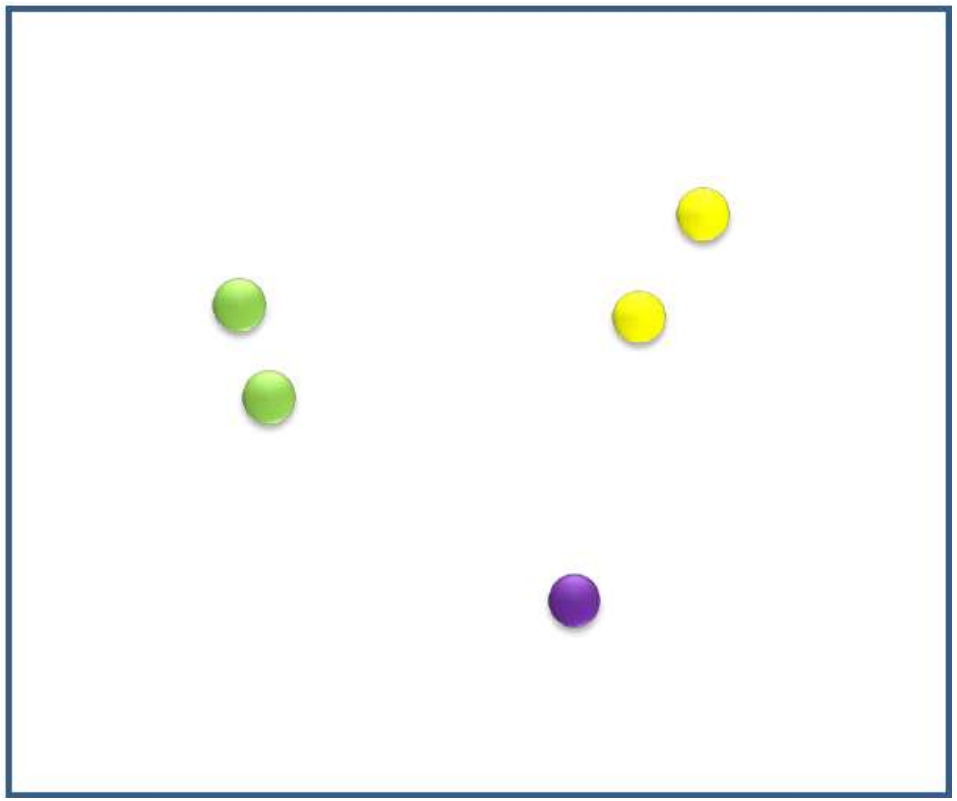
- **相似性的关键度量：**

类别相似性：样本---类簇（样本---集合）。

序贯方法

基本思想

逐一比较单个样本与类簇的相似性，有相似类则归类，无相似类则建立新类。



- **优点：**

一种简单的，快速算法。

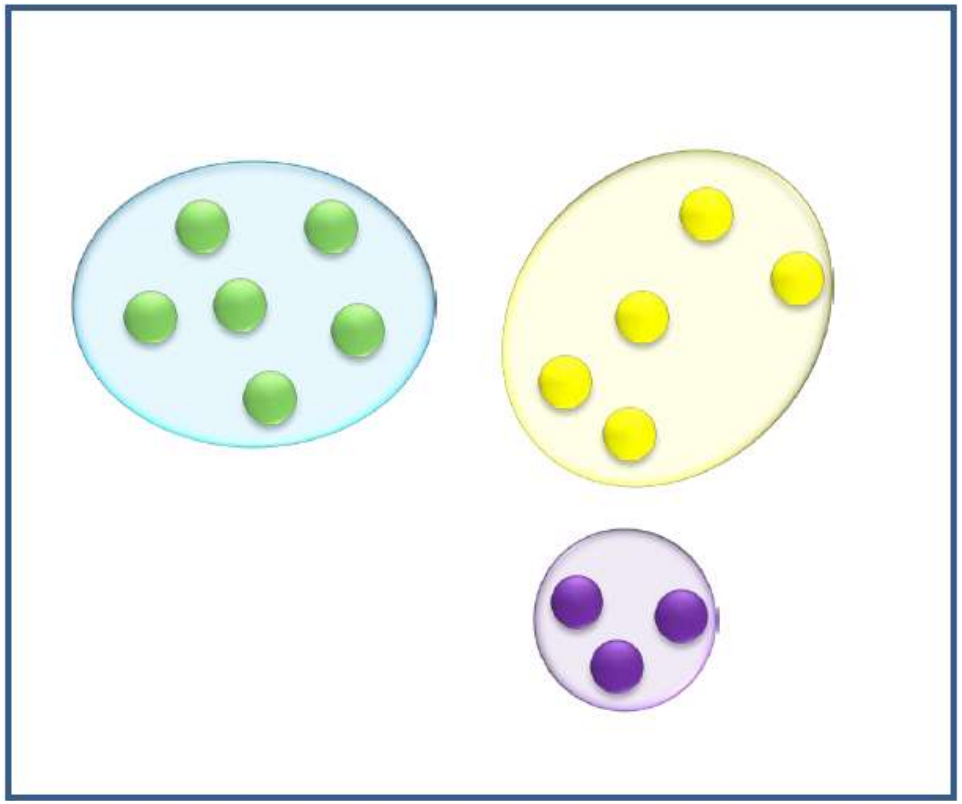
- **相似性的关键度量：**

类别相似性：样本---类簇（样本---集合）。

序贯方法

基本思想

逐一比较单个样本与类簇的相似性，有相似类则归类，无相似类则建立新类。



- **优点：**

一种简单的，快速算法。

- **相似性的关键度量：**

类别相似性：样本---类簇（样本---集合）。

序贯方法

基础的序贯方法

- $m = 1$
- $C_m = \{x_1\}$
- For $i = 2$ to N
  - Find C_k: $d(x_i, C_k) = \min_{1 \le j \le m} d(x_i, C_j)$.
  - If $(d(x_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m + 1$
    - $C_m = \{x_i\}$

    新建一个类
  - Else
    - $C_k = C_k \cup \{x_i\}$

    样本分给最相似的类
    - Where necessary, update representatives
  - End {if}
- End {For}

缺点：所有样本过滤一遍后才知道类别总数，而先出现的样本不能找到（后出现的）合适类别；

改进算法：采用两个阶段，类别确定、分类。

序贯方法

两阶段序贯方法

1：检测类别个数

- *Cluster Determination*
- $m = 1$
- $C_m = \{x_1\}$
  - For $i = 2$ to N
  - Find C_k: $d(x_i, C_k) = \min_{1 \le j \le m} d(x_i, C_j)$.
  - If $(d(x_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m + 1$
    - $C_m = \{x_i\}$
  - End {if}
- End {For}

只新建类别

2：类别划分

Pattern Classification

- For $i = 1$ to N
  - If x_i has not been assigned to a cluster, then
    - Find C_k: $d(x_i, C_k) = \min_{1 \le j \le m} d(x_i, C_j)$
    - $C_k = C_k \cup \{x_i\}$
    - Where necessary, update representatives
  - End {if}
- End {For}

只进行分类

缺点：以上两种方法依赖于阈值 Θ;

改进方法：弱化阈值作用，采用两个阈值，形成灰色带。

序贯方法

双阈值序贯方法

$d(x,C)<\Theta_1,\ x\in C$；

$d(x,C)>\Theta_2,\ x\in$ a new C

$\Theta_1<d(x,C)<\Theta_2,$ take place at later stage.

$m = 0$

$clas(\boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in X$

$prev\_change = 0$

$cur\_change = 0$

$exists\_change = 0$

序贯方法

While (there exists at least one feature vector $\boldsymbol{x}$ with $clas(\boldsymbol{x}) = 0$) do

- For $i = 1$ to N
  - if $clas(\boldsymbol{x}_i) = 0$ AND it is the first in the new while loop AND $exists\_change = 0$ then
    - $m = m + 1$
    - $C_m = \{\boldsymbol{x}_i\}$
    - $clas(\boldsymbol{x}_i) = 1$
    - $cur\_change = cur\_change + 1$
  - Else if $clas(\boldsymbol{x}_i) = 0$ then
    - Find $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
    - if $d(\boldsymbol{x}_i, C_k) < \Theta_1$ then
      - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
      - $clas(\boldsymbol{x}_i) = 1$
      - $cur\_change = cur\_change + 1$

最初的类别的建立，最后没有归类的，尝试自成一类

$d(x,C)<\Theta_1,\ x\in C$

$d(x,C)<\Theta_1,\ x\in C$

序贯方法

- else if $d(x_i, C_k) > \Theta_2$ then
  - $m = m + 1$
  - $C_m = \{x_i\}$
  - $clas(x_i) = 1$
  - $cur\_change = cur\_change + 1$

$d(x,C)>\Theta_2$, $x \in$ a new C

- End {If}
- Else if $clas(x_i) = 1$ then
  - $cur\_change = cur\_change + 1$
- End {If}

- End {For}
- $exists\_change = |cur\_change - prev\_change|$
- $prev\_change = cur\_change$
- $cur\_change = 0$

End {While}

- **前面的三种算法缺点：**（1）当类别一旦产生，不可变，尽管后来类簇增加，类别很相近也无法合并。（2）敏感于样本顺序，样本类别未必是最合适的。

序贯方法

增强算法

增强处理 1：对类别集合进行合并操作

Merging procedure

- (A) Find C_i, C_j $(i < j)$ such that $d(C_i, C_j) = \min_{k,r=1,\ldots,m,\, k \neq r} d(C_k, C_r)$
- If $d(C_i, C_j) \leq M_1$ then
  - Merge C_i, C_j to C_i and eliminate C_j.
  - Update the cluster representative of C_i (if cluster representatives are used).
  - Rename the clusters $C_{j+1}, \ldots, C_m$ to $C_j, \ldots, C_{m-1}$, respectively

增强处理 2：对样本类别重置

Reassignment procedure

- For $i = 1$ to N
  - Find C_j such that $d(\boldsymbol{x}_i, C_j) = \min_{k=1,\ldots,m} d(\boldsymbol{x}_i, C_k)$.
  - Set $b(i) = j$.
- End {For}
- For $j = 1$ to m
  - Set $C_j = \{\boldsymbol{x}_i \in X : b(i) = j\}$.
  - Update the representatives (if used).
- End {For}

第六章 聚类分析

6.1 概述

6.2 序贯方法

6.3 层次聚类

6.4 K 均值聚类

层次聚类

基本思想

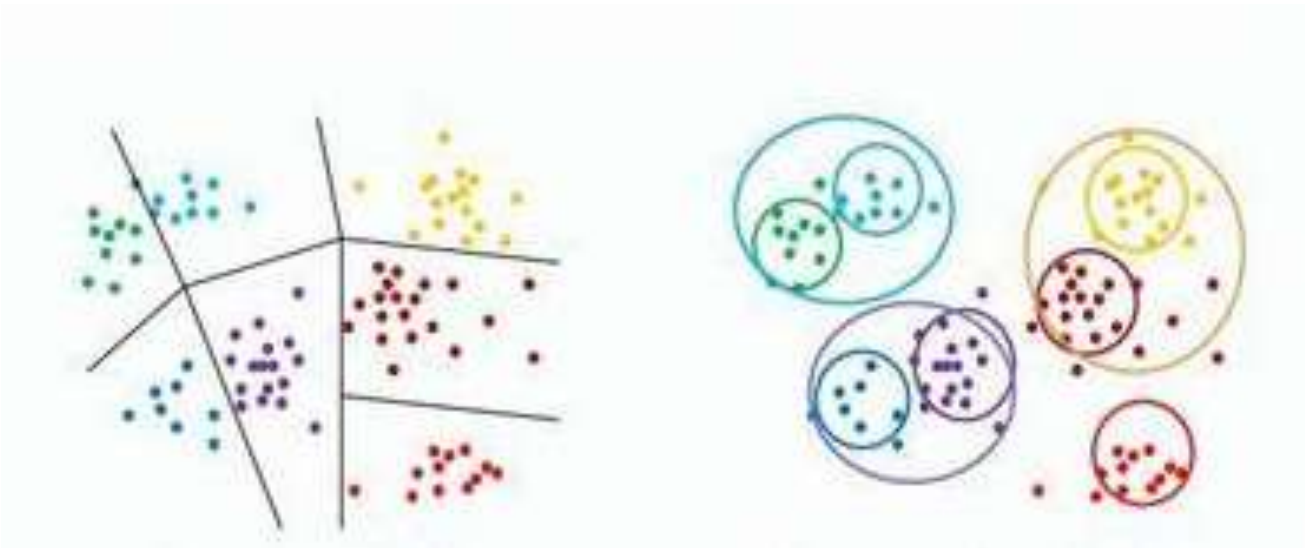
聚类嵌套定义：

R_1 和 R_2 是样本集 X 上的两种聚类划分，如果 R_1 中所有的类簇都是 R_2 中类簇的子集，则称 R_1 嵌套在 R_2 内，记作 $R_1 \subset R_2$。

例子：

$$R_1 = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\}$$
$$R_2 = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$$
$$R_3 = \{\{x_1, x_4\}, \{x_3\}, \{x_2, x_5\}\}$$
$$R_4 = \{\{x_1, x_2, x_4\}, \{x_3, x_5\}\}$$

$$R_1 \subset R_2$$
$$R_1 \not\subset R_3$$
$$R_1 \not\subset R_4$$



层次聚类

基本思想

层次聚类策略:

类簇之间（依据相似性）不断合并、或不断的分化，直到满足聚类停止条件。

自底向上/归并算法(agglomerative)

$$R_0 \subset R_1 \subset \ldots \subset R_{N-1}$$

自顶向下/分化算法(divisive)

$$R_{N-1} \subset \ldots \subset R_1 \subset R_0$$

层次聚类

归并算法

第 i 次迭代：计算所有两个类簇的相似性；

归并最相似的两个类簇，更新类别划分 R_i

缺点：没有归并的类簇间相似性，被重复计算

层次聚类

归并算法

Generalized Agglomerative Scheme (GAS)

- Initialization:
  - Choose $\mathfrak{R}_0 = \{C_i = \{\boldsymbol{x}_i\},\ i = 1, \ldots, N\}$ as the initial clustering.
  - $t = 0$.
- Repeat:
  - $t = t + 1$
  - Among all possible pairs of clusters (C_r, C_s) in $\mathfrak{R}_{t-1}$ find the one, say (C_i, C_j), such that

$$g(C_i, C_j) = \begin{cases} \min_{r,s} g(C_r, C_s), & \text{if } g \text{ is a dissimilarity function} \\ \max_{r,s} g(C_r, C_s), & \text{if } g \text{ is a similarity function} \end{cases}$$

  - Define $C_q = C_i \cup C_j$ and produce the new clustering $\mathfrak{R}_t = (\mathfrak{R}_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$.
- Until all vectors lie in a single cluster.

层次聚类

归并算法

例子 1：

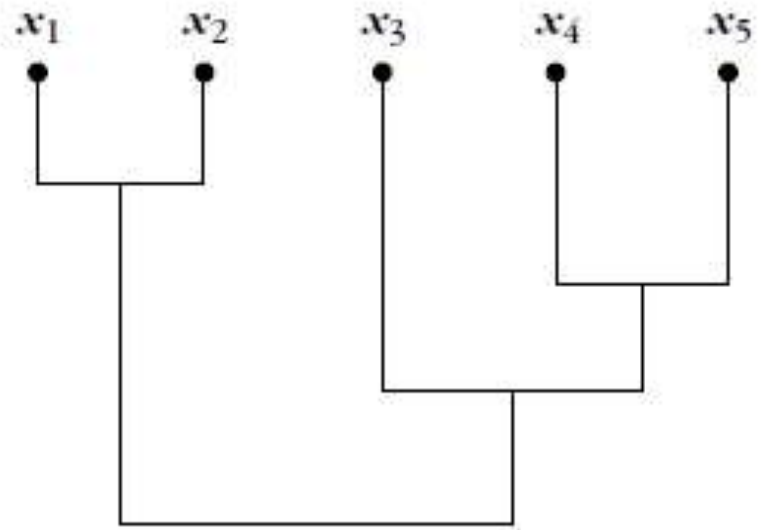
$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1, x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1, x_2\},\{x_3\},\{x_4, x_5\}\}$

$\{\{x_1, x_2\},\{x_3, x_4, x_5\}\}$

$\{\{x_1, x_2, x_3, x_4, x_5\}\}$

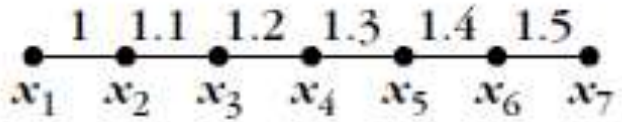


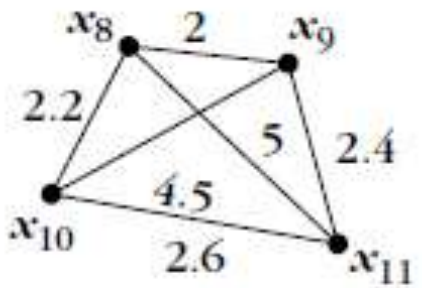
层次聚类

归并算法

例子 2:

- 数据



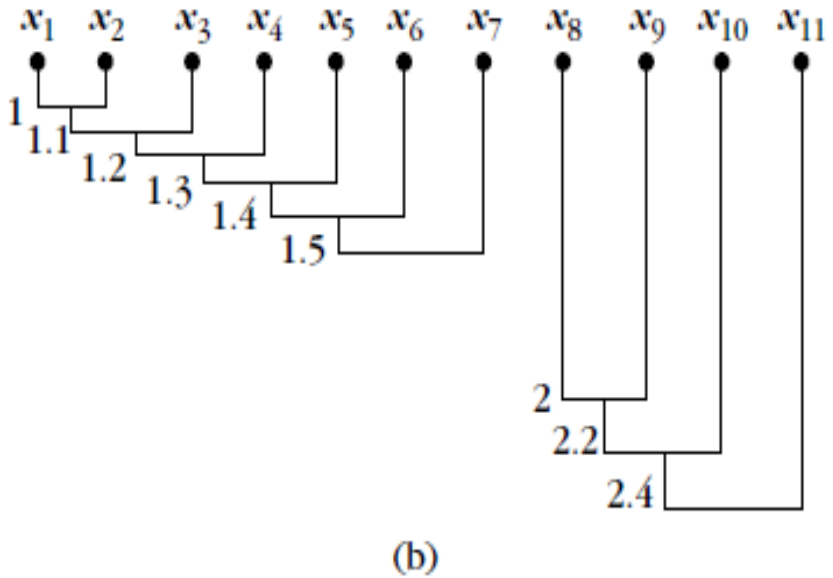


(a)

层次聚类

归并算法

- 基于 similarity function



(b)

层次聚类

归并算法

基于矩阵的归并算法

利用矩阵记录类簇间的相似性

(a) 删除对应合并的两行和列

(b) 增加一行和列： 新类簇与其他类簇的相似度

优点：不必重新计算“没有合并的类簇间”的相似性

层次聚类

归并算法

基于矩阵的归并算法

需要计算的相似性：

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

合并后类簇相似性矩阵：

	C1	C2	C3	C4
C1				
C2				
C3				
C4				

层次聚类

归并算法

基于矩阵的归并算法

Matrix Updating Algorithmic Scheme (MUAS)

- Initialization:
  - $\Re_0 = \{\{\boldsymbol{x}_i\}, i = 1, \ldots, N\}$.
  - $P_0 = P(X)$.
  - $t = 0$
- Repeat:
  - $t = t + 1$
  - Find C_i, C_j such that $d(C_i, C_j) = \min_{r,s=1,\ldots,N,\, r \neq s} d(C_r, C_s)$
  - Merge C_i, C_j into a single cluster C_q and form $\Re_t = (\Re_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$.
  - Define the proximity matrix P_t from P_{t-1} as explained in the text.
- Until $\Re_{N-1}$ clustering is formed, that is, all vectors lie in the same cluster.

层次聚类

分化算法

过程与归并相反;

第 i 次迭代：

在所有类簇的所有划分中，计算所有两个类簇相似性，

选择最不相似的类簇集合划分，更新类别划分 R_i

缺点：没有划分的类簇间相似性，被重复计算

层次聚类

分化算法

- Initialization
  - Choose $\Re_0 = \{X\}$ as the initial clustering.
  - $t = 0$
- Repeat
  - $t = t + 1$
  - For $i = 1$ to t
    - Among all possible pairs of clusters (C_r, C_s) that form a partition of $C_{t-1,i}$, find the pair $(C^1_{t-1,i}, C^2_{t-1,i})$ that gives the maximum value for g.
  - Next i
  - From the t pairs defined in the previous step choose the one that maximizes g. Suppose that this is $(C^1_{t-1,j}, C^2_{t-1,j})$.
  - The new clustering is

$$\Re_t = (\Re_{t-1} - \{C_{t-1,j}\}) \cup \{C^1_{t-1,j}, C^2_{t-1,j}\}$$

  - Relabel the clusters of $\Re_t$.
- Until each vector lies in a single distinct cluster.

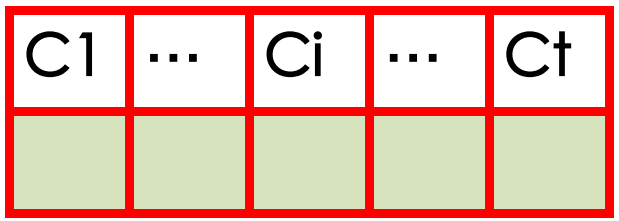
类簇数量

对于包含有 n 个样本的类簇，可能的分化有几种？
$(2^n/2)-1=2^{n-1}-1$

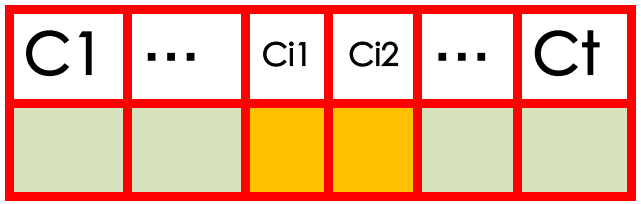
层次聚类

分化算法

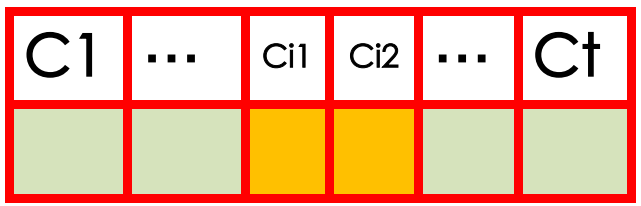
类簇相似性矩阵



分化后类簇相似性矩阵



下一轮，只需计算对增加的类簇，进行分化计算：



层次聚类

例子

数据（《机器学习》，周志华，2016， 表 9.1）

表 9.1 西瓜数据集 4.0

编号	密度	含糖率	编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	0.149	17	0.719	0.103	27	0.532	0.472
8	0.437	0.211	18	0.359	0.188	28	0.473	0.376
9	0.666	0.091	19	0.339	0.241	29	0.725	0.445
10	0.243	0.267	20	0.282	0.257	30	0.446	0.459

层次聚类

例子

归并过程

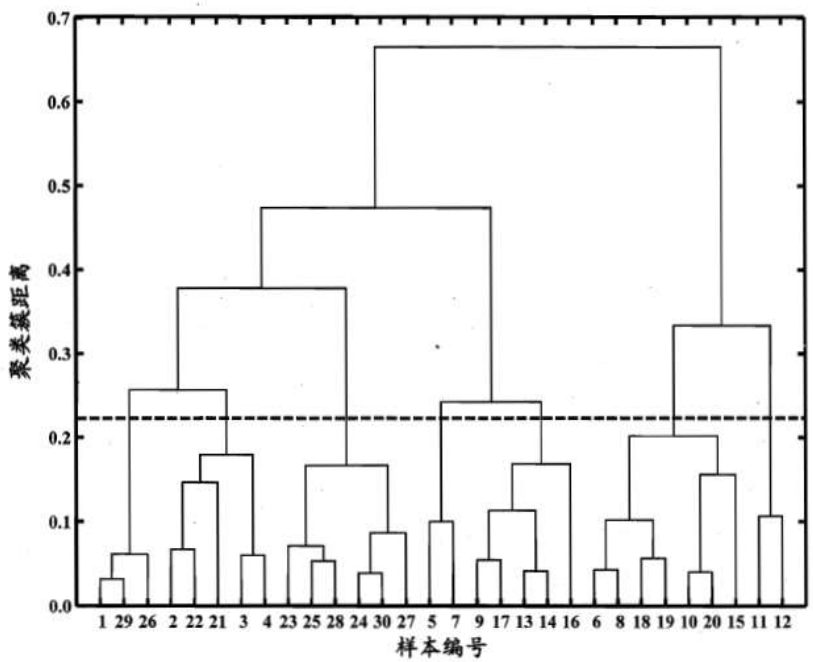
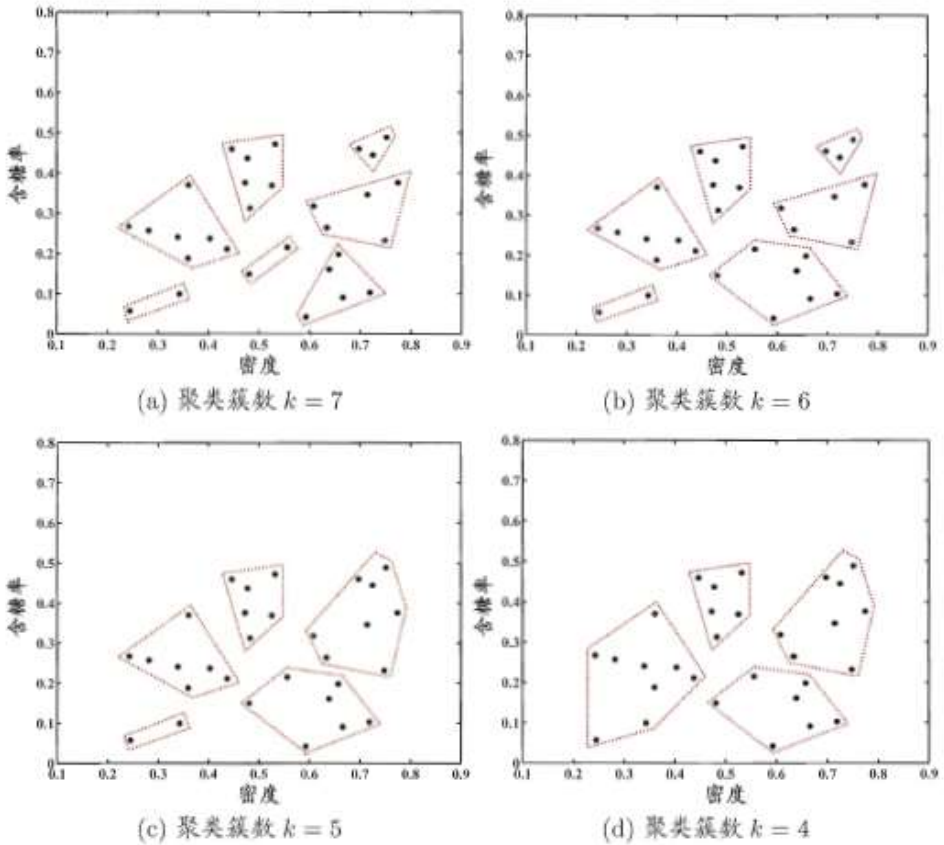


图 9.12 西瓜数据集 4.0 上 AGNES 算法生成的树状图(采用 $d_{\max}$). 横轴对应于样本编号, 纵轴对应于聚类簇距离.

层次聚类

如何确定聚类个数？



(a) 聚类簇数 $k = 7$　(b) 聚类簇数 $k = 6$

(c) 聚类簇数 $k = 5$　(d) 聚类簇数 $k = 4$

第六章 聚类分析

6.1 概述

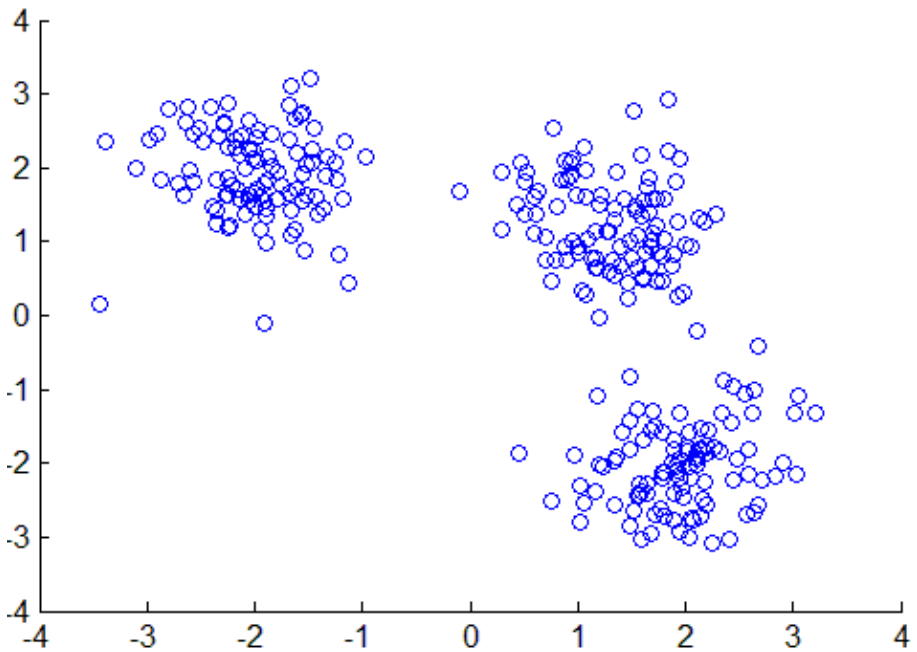
6.2 序贯方法

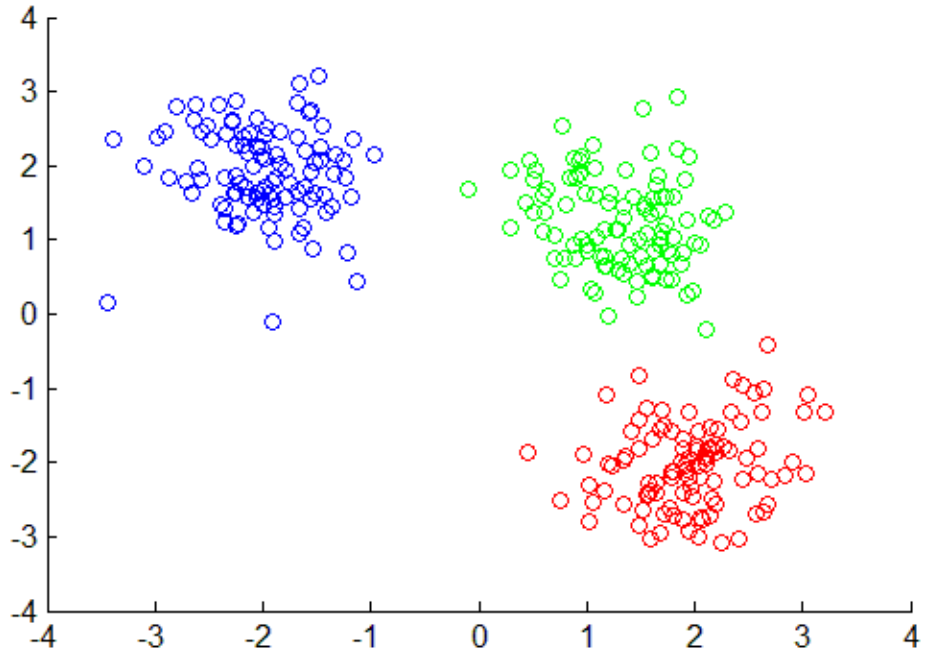
6.3 层次聚类

6.4 K 均值聚类

K 均值聚类

示例

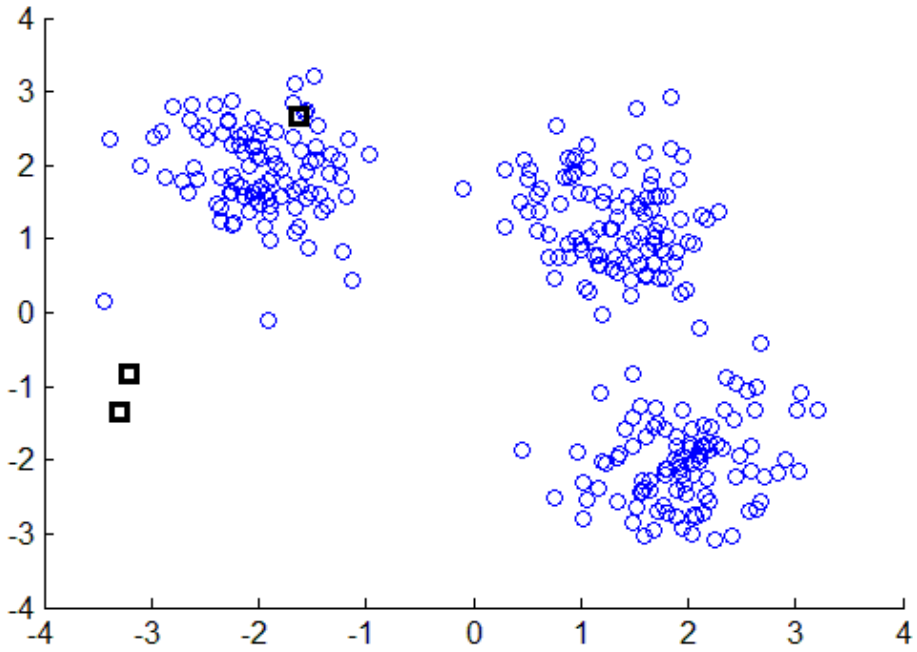




什么样的聚类算法，能实现类内距离最小？

K 均值聚类

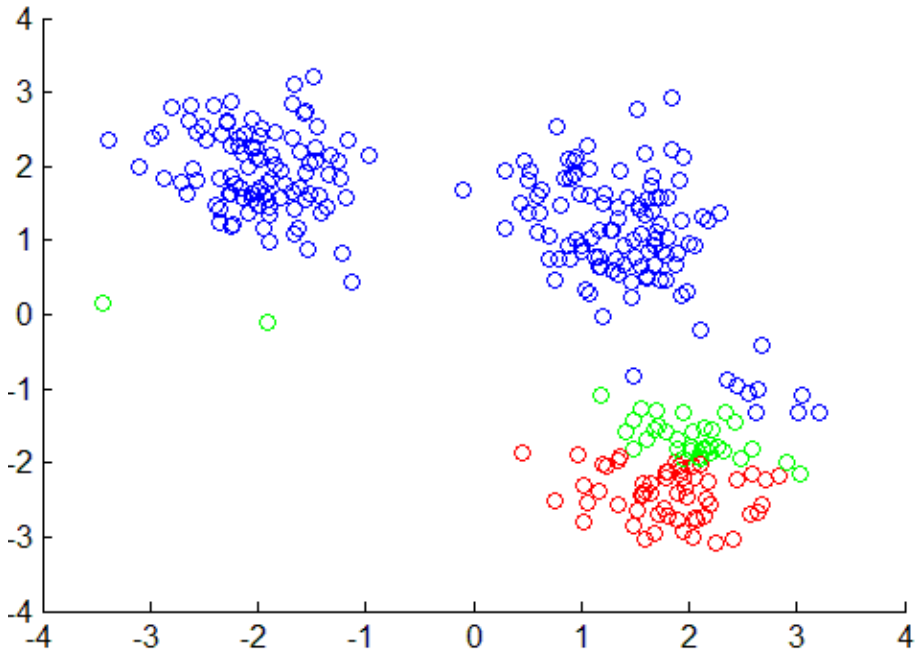
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

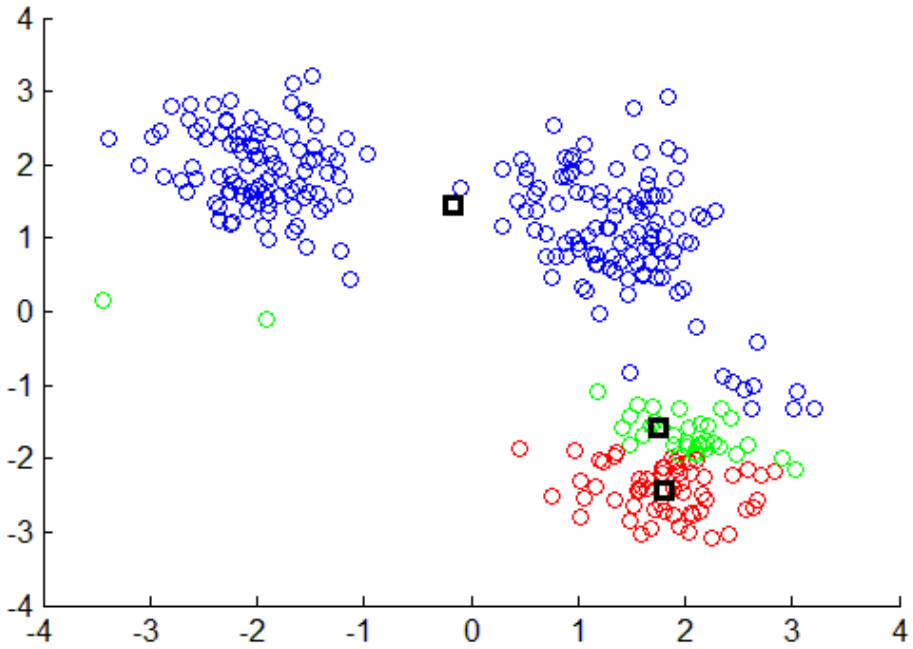
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

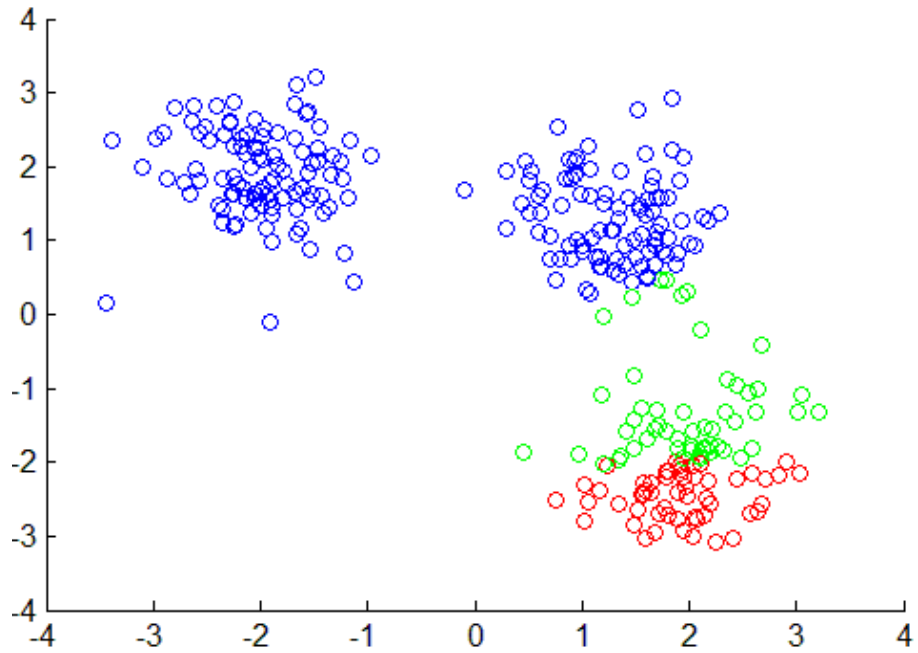
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

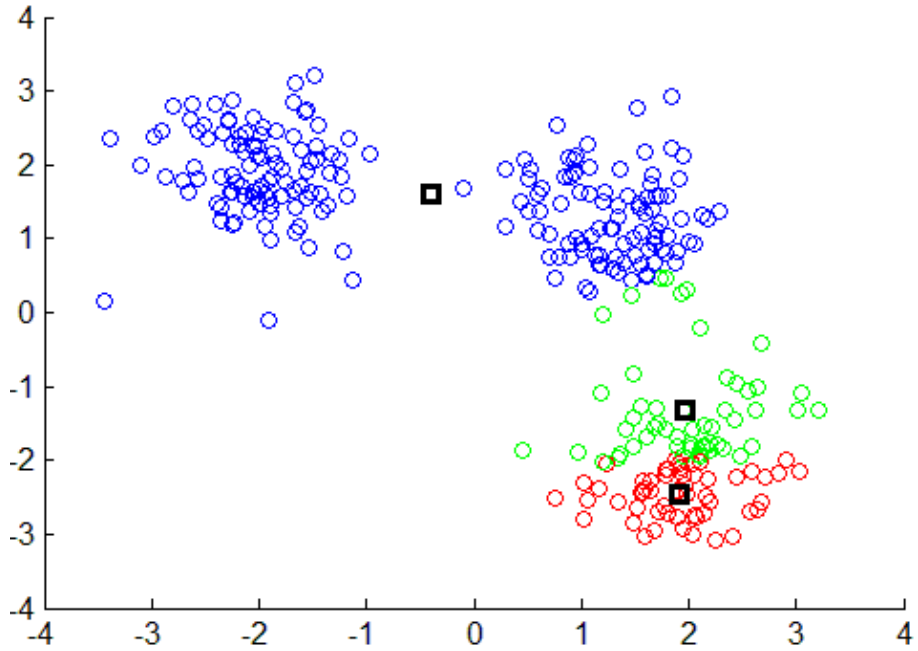
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

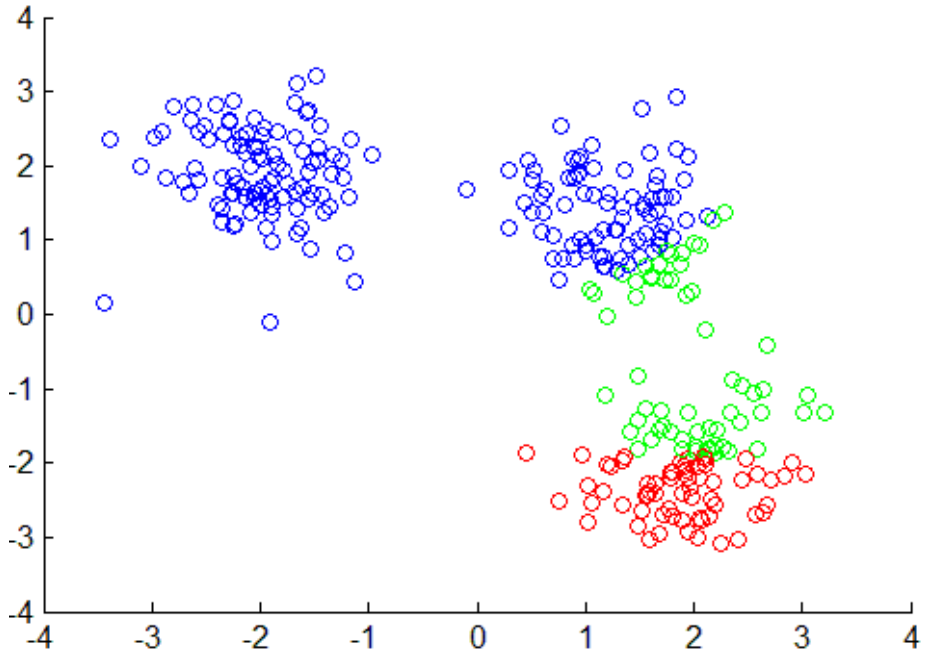
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

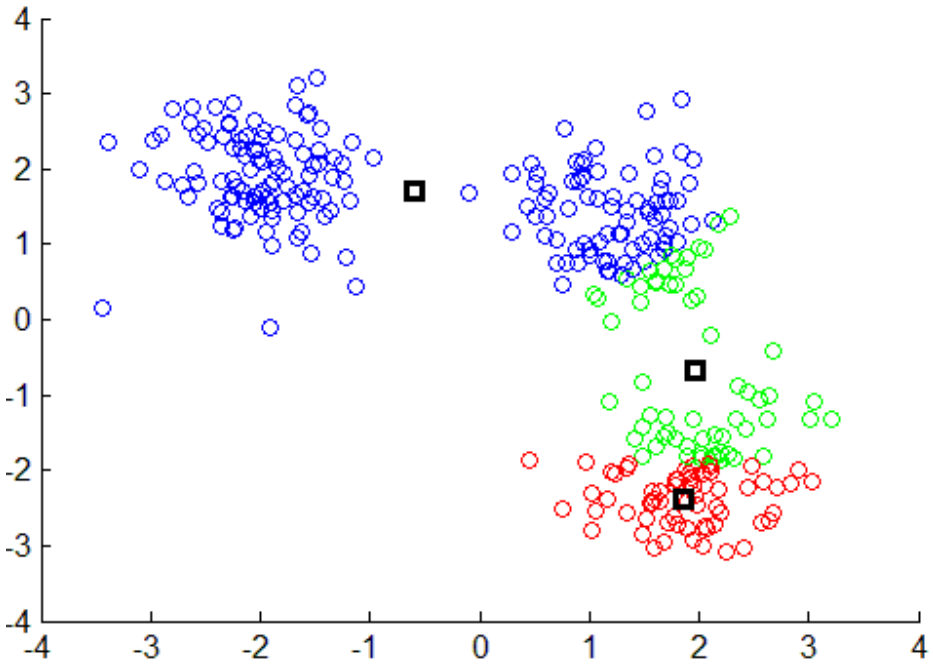
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

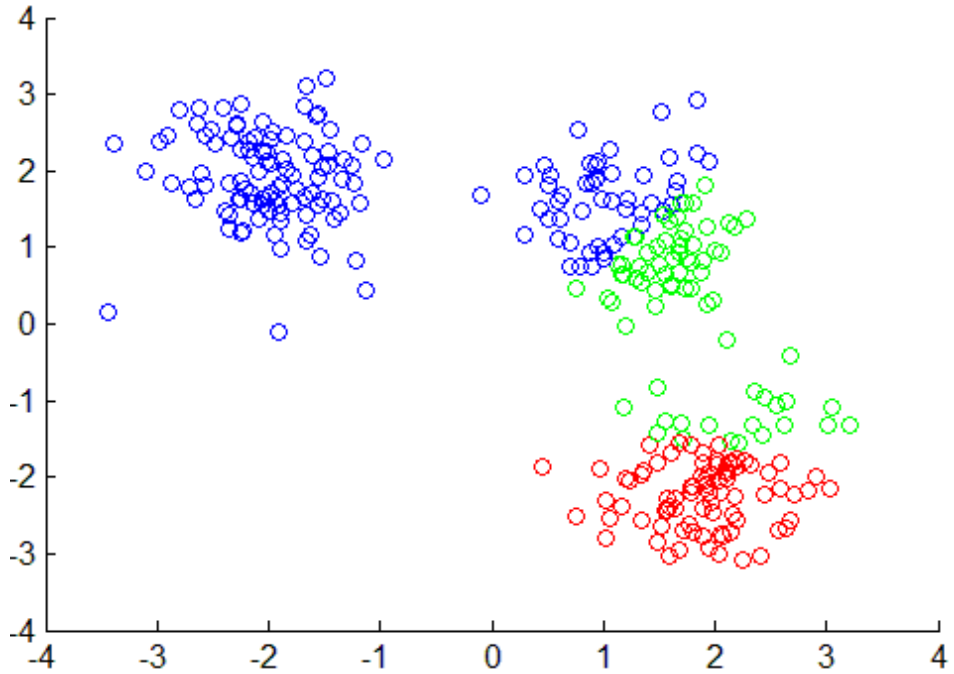
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

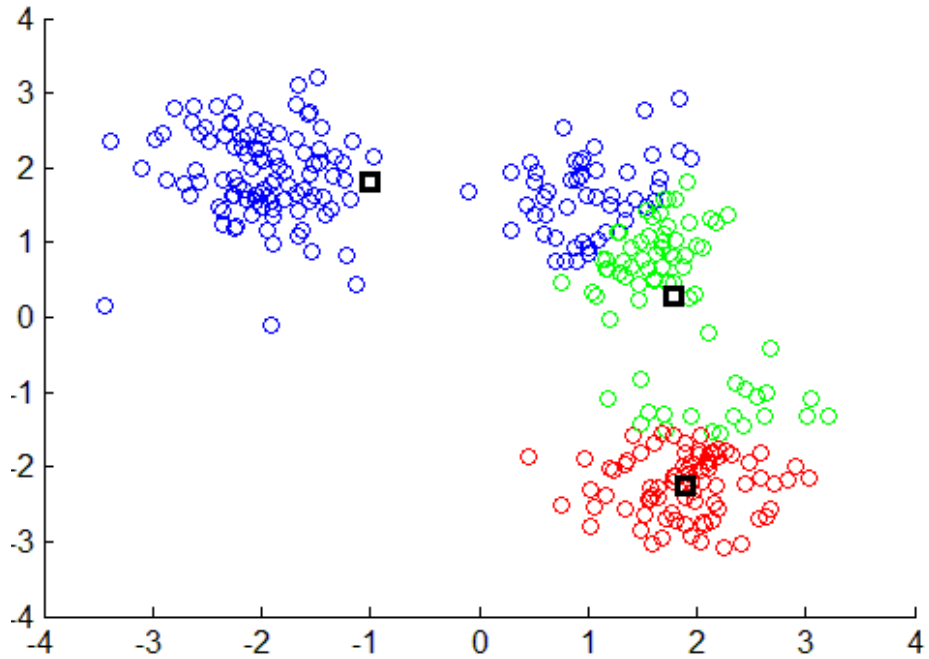
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

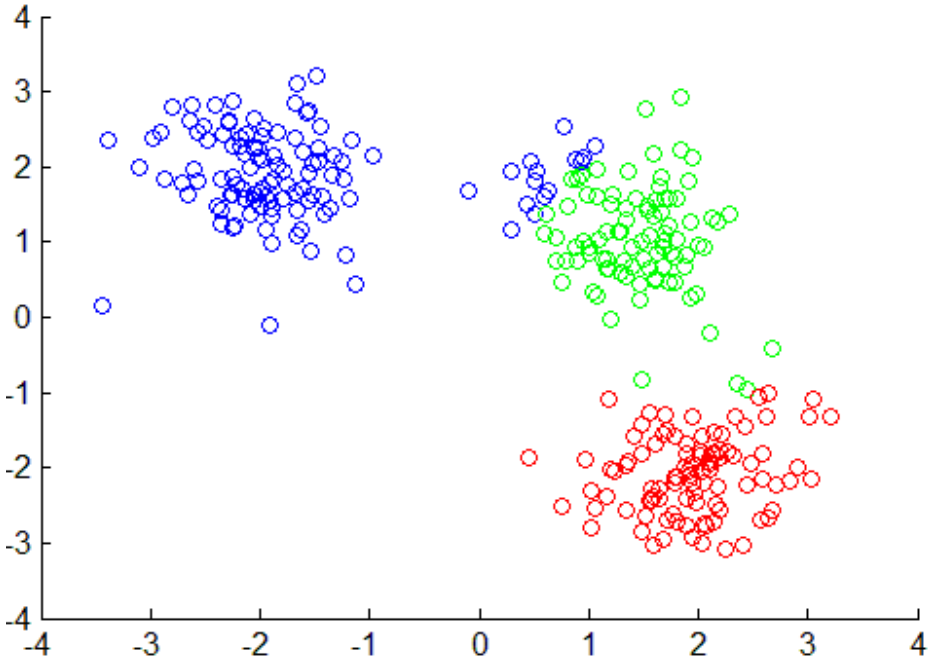
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

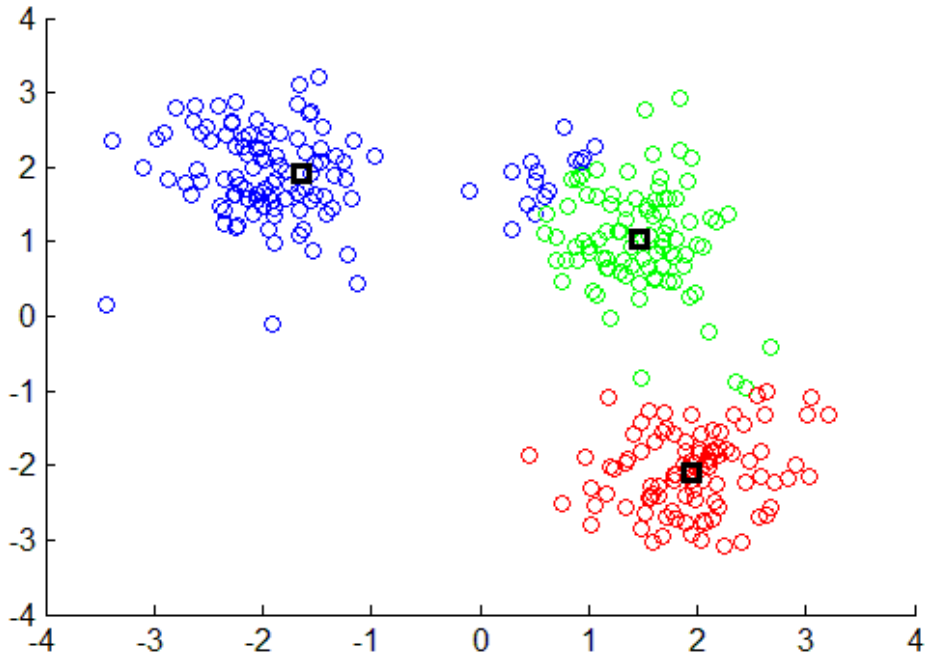
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

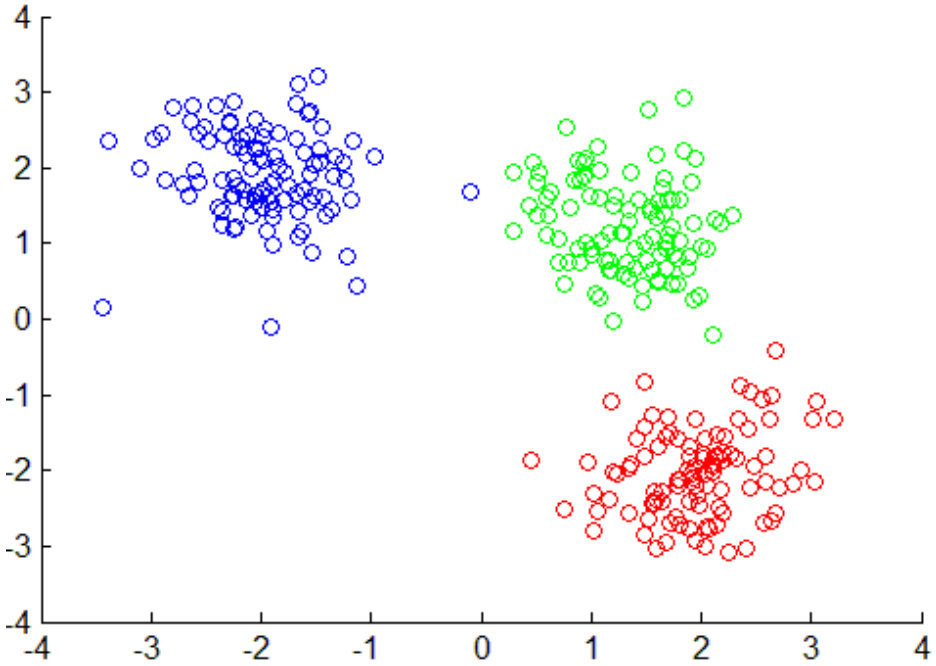
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

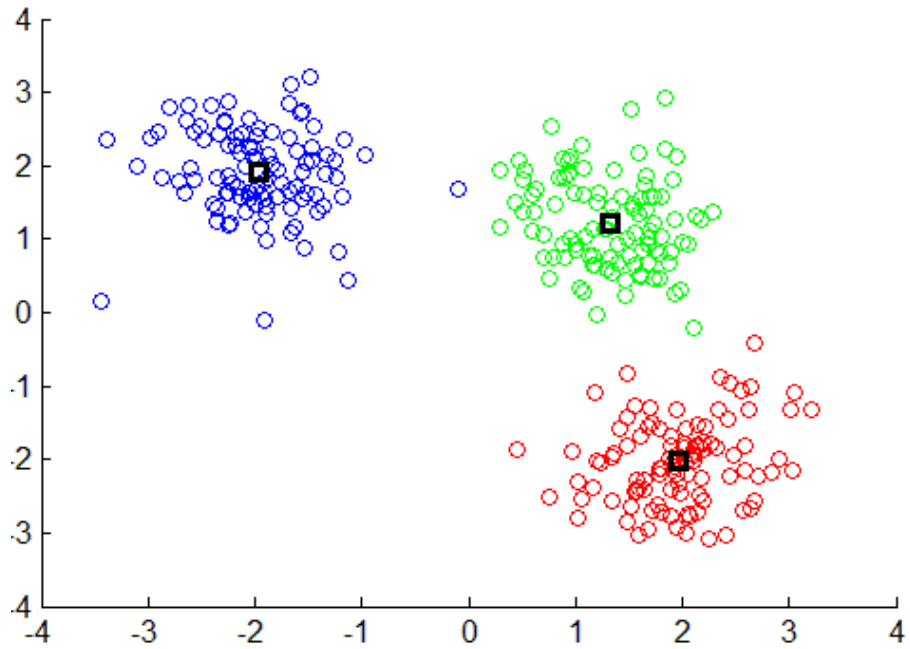
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

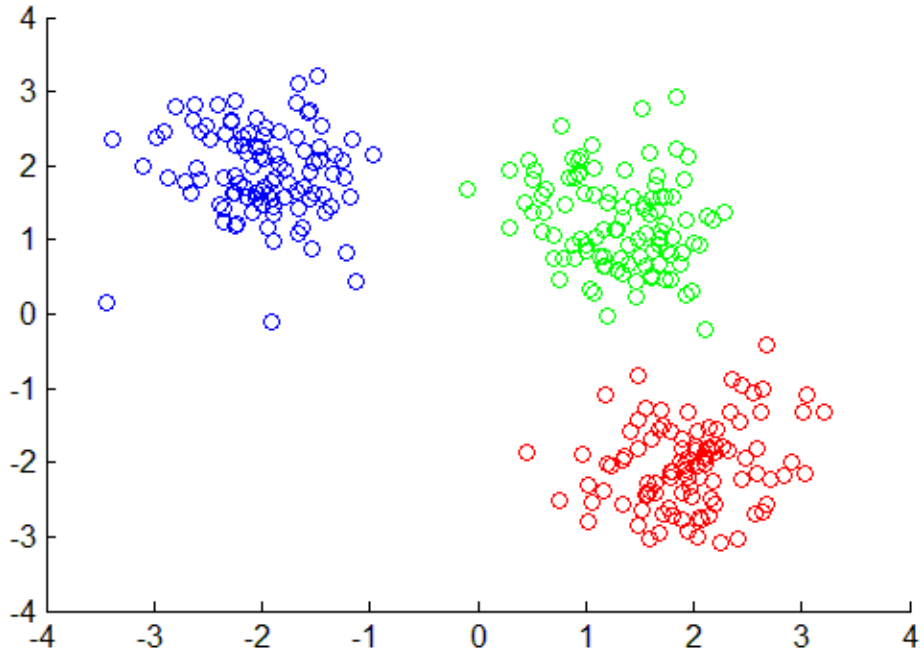
示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

示例



Kmeans：将样本分给最近的类心，然后重新调整类心；通过多次迭代，逐步进行类别划分。

K 均值聚类

最优准则

最小化误差平方和

$$Je=\sum_{i=1}^{K}\sum_{y\in C_i}\|y-m_i\|^2=\sum_{i=1}^{K}Je_i$$

$$m_i=\frac{1}{N_i}\sum_{y\in C_i}y$$

$y\in C_i$ 是第 i 个类簇的样本

⇨

误差的扩展：也可以采用余弦距离，或其他反映距离和误差的度量。

$$J=-\sum_{i=1}^{K}\sum_{y\in C_i}\frac{y\cdot m_i}{\|y\|\|m_i\|}$$

K 均值聚类

K-means

一般方法：最近类心原则，批量划分后修正类心(如示例过程)

For

（1）类簇划分

$$\lambda_j = \underset{i\in\{12,\ldots,k\}}{\operatorname{argmin}} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|_2$$

$$C_{\lambda_j} = C_{\lambda_j} \cup \{\boldsymbol{x}_j\}$$

（2） 更新类簇中心

$$\boldsymbol{\mu}_i' = \frac{1}{|C_i|} \sum_{\boldsymbol{x}\in C_i} \boldsymbol{x}$$

End For

存在的问题：

（1） 可能导致空的类簇

（2） 批量修正使得划分并不能最好的收敛目标（最小化误差平方和）

K 均值聚类

K-means

一般性的流程可以如下：

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
聚类簇数 k.

过程：

1: 从 D 中随机选择 k 个样本作为初始均值向量 $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$
2: **repeat**
3: 令 $C_i = \varnothing \ (1 \leqslant i \leqslant k)$
4: **for** $j = 1, 2, \ldots, m$ **do**
5: 计算样本 $\boldsymbol{x}_j$ 与各均值向量 $\boldsymbol{\mu}_i \ (1 \leqslant i \leqslant k)$ 的距离: $d_{ji} = \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|_2$;
6: 根据距离最近的均值向量确定 $\boldsymbol{x}_j$ 的簇标记: $\lambda_j = \arg\min_{i \in \{1,2,\ldots,k\}} d_{ji}$;
7: 将样本 $\boldsymbol{x}_j$ 划入相应的簇: $C_{\lambda_j} = C_{\lambda_j} \bigcup \{\boldsymbol{x}_j\}$;
8: **end for**
9: **for** $i = 1, 2, \ldots, k$ **do**
10: 计算新均值向量: $\boldsymbol{\mu}_i' = \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x}$;
11: **if** $\boldsymbol{\mu}_i' \neq \boldsymbol{\mu}_i$ **then**
12: 将当前均值向量 $\boldsymbol{\mu}_i$ 更新为 $\boldsymbol{\mu}_i'$
13: **else**
14: 保持当前均值向量不变
15: **end if**
16: **end for**
17: **until** 当前均值向量均未更新

输出： 簇划分 $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

K 均值聚类

K-means

改进方法：单个划分最优原则，单个划分后修正类心

把 y 从 Γ_i 移到 Γ_k 中；

两个类别由 y 引起的类心变化：

$$\boldsymbol{m}_i = \boldsymbol{m}_i + \frac{1}{N_i - 1}(\boldsymbol{m}_i - \boldsymbol{y})$$

$$\boldsymbol{m}_k = \boldsymbol{m}_k + \frac{1}{N_k + 1}(\boldsymbol{y} - \boldsymbol{m}_k)$$

两个类别由 y 引起的均方误差变化：

$$Je_i = Je_i - \frac{N_i}{N_i - 1}\|\boldsymbol{y} - \boldsymbol{m}_i\|^2$$

$$Je_k = Je_k + \frac{N_k}{N_k + 1}\|\boldsymbol{y} - \boldsymbol{m}_k\|^2$$

K 均值聚类

K-means

推导过程

$$Je_i^* = \left(\sum_{x \in D_i} \| x - m_i^* \|^2 \right) - \| y - m_i^* \|^2$$

m*

$$= \sum_{x \in D_i} \| x - m_i - \frac{(m_i - y)}{N_i - 1} \|^2 - \| \frac{N_i}{N_i - 1} (y - m_i) \|^2$$

$$= \sum_{x \in D_i} \left(\| x - m_i \|^2 + \frac{2}{N_i - 1} (x - m_i)^T (y - m_i) + \frac{\| y - m_i \|^2}{(N_i - 1)^2} \right) - \| \frac{N_i}{N_i - 1} (y - m_i) \|^2$$

$$= Je_i + \frac{2}{N_i - 1} (m_i - y)^T \sum_{x \in D_i} (x - m_i) + \frac{N_i \| y - m_i \|^2}{(N_i - 1)^2} - \| \frac{N_i}{N_i - 1} (y - m_i) \|^2$$

等于0

$$= Je_i - \frac{N_i \| y - m_i \|^2}{N_i - 1}$$

K 均值聚类

K-means

推导过程

$$Je_k^* = \sum_{x \in D_k} \| x - m_k^* \|^2 + \| y - m_k^* \|^2$$

$$= \sum_{x \in D_k} \| x - m_k - \frac{(y - m_k)}{N_k + 1} \|^2 + \| \frac{N_k}{N_k + 1}(y - m_k) \|^2$$

$$= \sum_{x \in D_k} \left(\| x - m_k \|^2 - \frac{2}{N_k + 1}(x - m_k)^T (y - m_k) + \frac{\| y - m_k \|^2}{(N_k + 1)^2} \right) + \| \frac{N_k}{N_k + 1}(y - m_k) \|^2$$

$$= Je_k - \frac{2}{N_k + 1}(y - m_k)^T \sum_{x \in D_k} (x - m_k) + \frac{N_k \| y - m_k \|^2}{(N_k + 1)^2} + \| \frac{N_k}{N_k + 1}(y - m_k) \|^2$$

等于 0

$$= Je_k + \frac{N_k \| y - m_k \|^2}{N_k + 1}$$

K 均值聚类

K-means

For a sample in Class Γ_i:

(1) 如果 N_i=1，则放弃该样本； 否则继续；

(2) 计算与各类别 Γ_i 的相似度：

$$\rho_j=\begin{cases}\dfrac{N_j}{N_j+1}\|\boldsymbol{y}-\boldsymbol{m}_j\|^2, & j\neq i\\ \dfrac{N_i}{N_i-1}\|\boldsymbol{y}-\boldsymbol{m}_j\|^2, & j=i\end{cases}$$

(3) 根据与各类别的相似性 ρ_j，
将样本划分为最近类簇。若 ρ_k 最小，
把 y 从 Γ_i 移到 Γ_k 中；

（4）修整被调整的两个类的类心 m_i 和 m_k，$i=1,2,\ldots,K$；

$$\boldsymbol{m}_i=\boldsymbol{m}_i+\frac{1}{N_i-1}(\boldsymbol{m}_i-\boldsymbol{y})$$

$$\boldsymbol{m}_k=\boldsymbol{m}_k+\frac{1}{N_k+1}(\boldsymbol{y}-\boldsymbol{m}_k)$$

（5）计算 Je，若 N 步后，Je 不变，算法停止。

$$Je_i=Je_i-\frac{N_i}{N_i-1}\|\boldsymbol{y}-\boldsymbol{m}_i\|^2$$

$$Je_k=Je_k+\frac{N_k}{N_k+1}\|\boldsymbol{y}-\boldsymbol{m}_k\|^2$$

End For

K-means

分析(3)和(5)

$$\rho_j = \begin{cases} \dfrac{N_j}{N_j+1}\|\boldsymbol{y}-\boldsymbol{m}_j\|^2, & j \neq i \\ \dfrac{N_i}{N_i-1}\|\boldsymbol{y}-\boldsymbol{m}_j\|^2, & j = i \end{cases}$$

保证了每个样本迭代后误差越来越小。

K 均值聚类

例子

表 9.1 西瓜数据集 4.0

编号	密度	含糖率	编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	0.149	17	0.719	0.103	27	0.532	0.472
8	0.437	0.211	18	0.359	0.188	28	0.473	0.376
9	0.666	0.091	19	0.339	0.241	29	0.725	0.445
10	0.243	0.267	20	0.282	0.257	30	0.446	0.459

K 均值聚类

例子

(1) 初始化

假定聚类簇数 $k=3$, 算法开始时随机选取三个样本 $\boldsymbol{x}_6$, $\boldsymbol{x}_{12}$, $\boldsymbol{x}_{27}$ 作为初始均值向量, 即

$$\boldsymbol{\mu}_1=(0.403;0.237),\ \boldsymbol{\mu}_2=(0.343;0.099),\ \boldsymbol{\mu}_3=(0.532;0.472)\ .$$

(2) 样本划分

$$C_1=\{\boldsymbol{x}_5,\boldsymbol{x}_6,\boldsymbol{x}_7,\boldsymbol{x}_8,\boldsymbol{x}_9,\boldsymbol{x}_{10},\boldsymbol{x}_{13},\boldsymbol{x}_{14},\boldsymbol{x}_{15},\boldsymbol{x}_{17},\boldsymbol{x}_{18},\boldsymbol{x}_{19},\boldsymbol{x}_{20},\boldsymbol{x}_{23}\};$$

$$C_2=\{\boldsymbol{x}_{11},\boldsymbol{x}_{12},\boldsymbol{x}_{16}\};$$

$$C_3=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4,\boldsymbol{x}_{21},\boldsymbol{x}_{22},\boldsymbol{x}_{24},\boldsymbol{x}_{25},\boldsymbol{x}_{26},\boldsymbol{x}_{27},\boldsymbol{x}_{28},\boldsymbol{x}_{29},\boldsymbol{x}_{30}\}.$$

K 均值聚类

例子

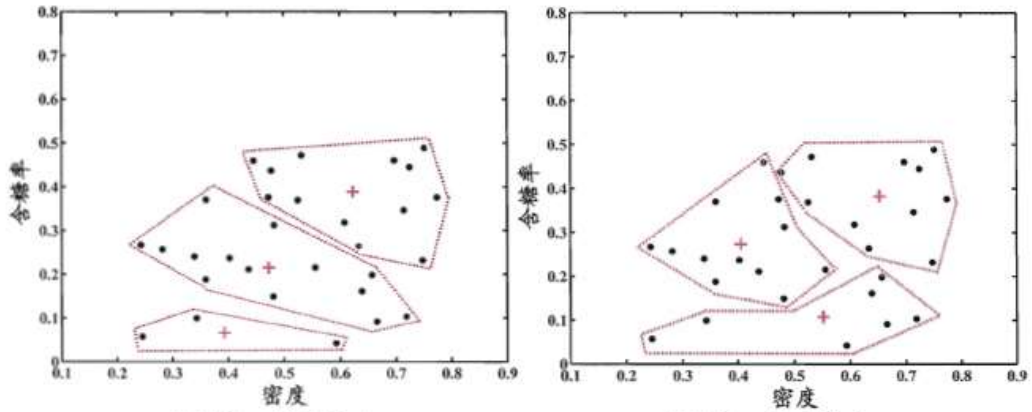
（3）均值更新

$$\boldsymbol{\mu}_1' = (0.473; 0.214),\ \boldsymbol{\mu}_2' = (0.394; 0.066),\ \boldsymbol{\mu}_3' = (0.623; 0.388)$$

重复（2）和（3）样本划分 。。。均值更新。。。

K 均值聚类

例子



(a) 第一轮迭代后　(b) 第二轮迭代后

(c) 第三轮迭代后　(d) 第四轮迭代后

参考文献

1. 周志华，机器学习，清华大学出版社，2016.
2. Duda, R.O. et al. Pattern classification. 2nd, 2003.
3. 边肇祺，张学工等编著，模式识别(第二版)，清华大学，1999。
4. Chris Bishop. Pattern recognition and Machine Learning. Springer, 2006.（PR&ML）