河北师范大学软件学院
Software College of Hebei Normal University

非监督式机器学习

聚类(Clustering)

河北师范大学软件学院

2018.04.03-04.12

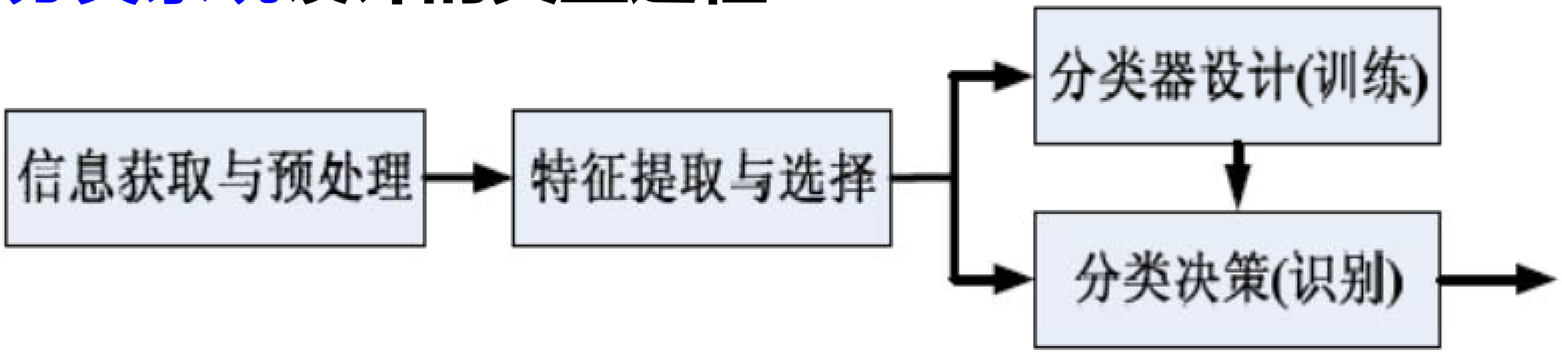
主要内容

1. 聚类的引入
2. 动态聚类
3. 基于模型的聚类
4. 密度聚类
5. 层次聚类

6. 其它聚类算法

问题的引入

分类系统设计的典型过程



监督式学习

不足：需要大量已知类别标签的样本集；
收集与标记费时、费力

实际面临问题：样本数据多是无标签数据

非监督式学习(密度函数估计、**聚类**、降维...)

例：**聚类系统设计的典型过程**

适用：

(1)大型数据挖掘：大量未标记数据训练分类器；

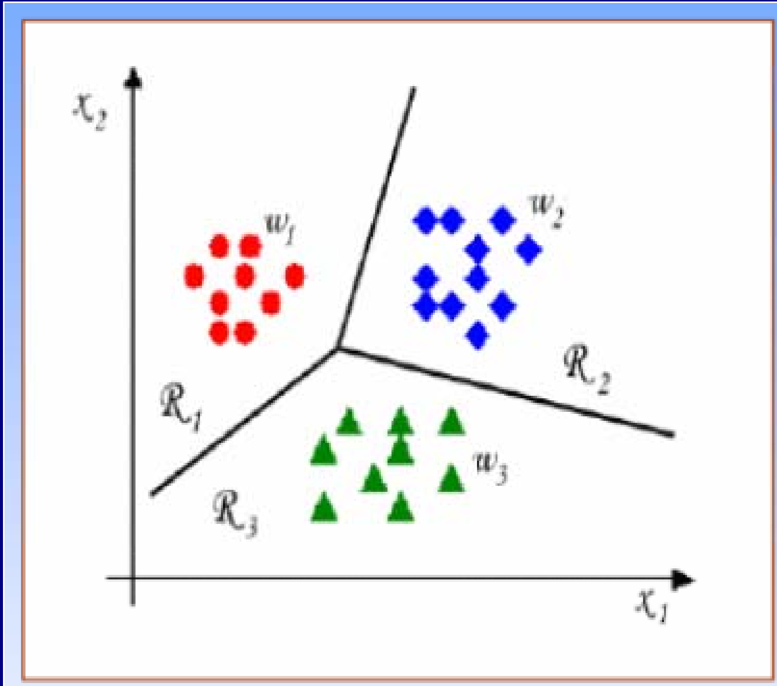
人工标记分组结果

(2)揭示观测数据的内在结构特性

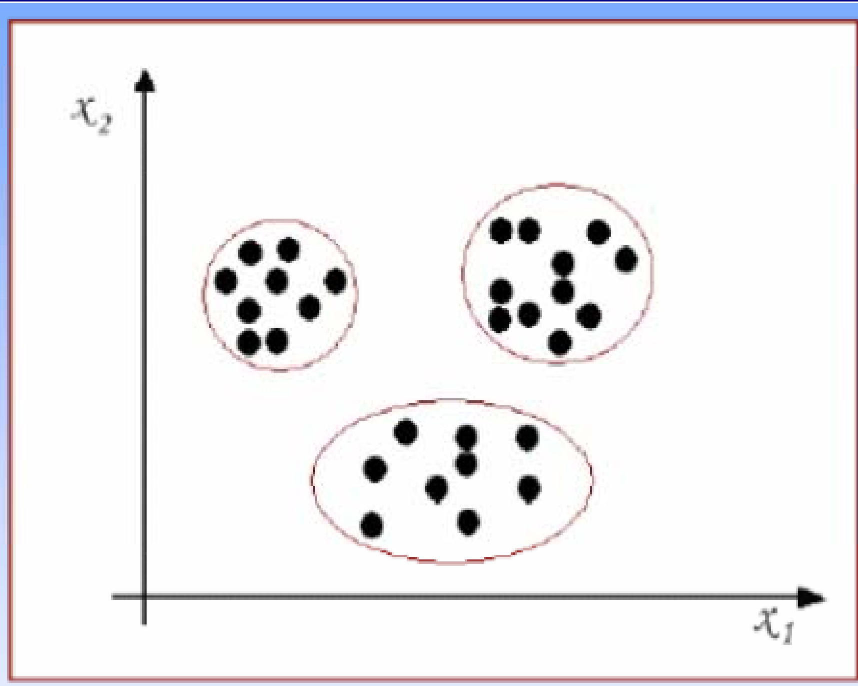
(3)是分类或其它学习任务的前驱阶段

提取数据的基本特征，进一步用于分类...

分类(Classification)与聚类(Clustering)



Given labeled training patterns, construct decision boundaries or partition the feature space



Given some patterns, discover the underlying structure (categories) in the data

什么是聚类(clustering)

http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/

To find a structure in a collection of unlabeled data.

- ***A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way".***

- ***A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.***

- ***High intra-class(cluster) similarity***
  Low inter-class(cluster) similarity

聚类问题的描述

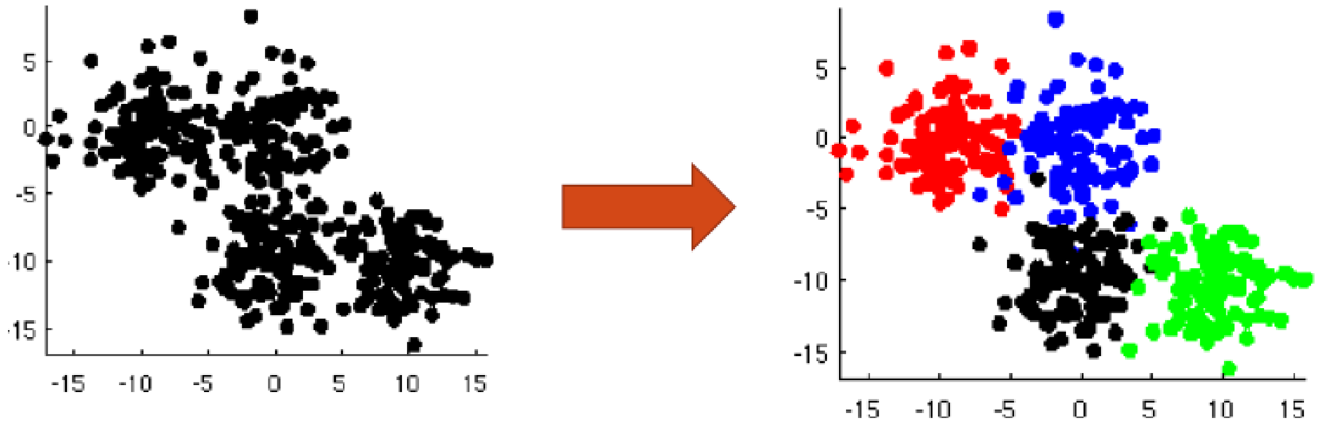
河北师范大学软件学院
Software College of Hebei Normal University

输入：无标签数据集 $\boldsymbol{D}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$, $\boldsymbol{x}_i=\begin{bmatrix}x_{i1} & \cdots & x_{id}\end{bmatrix}^T\in R^d$

要生成的簇的数目k

输出：k个互不相交的簇$\{\boldsymbol{C}_l\mid l=1,\ldots,k\}$

$$\begin{cases}\boldsymbol{C}_i\bigcap_{i\neq j}\boldsymbol{C}_j=\Phi\\ \bigcup_{l=1}^{k}\boldsymbol{C}_l=\boldsymbol{D}\end{cases}$$



λ_j——样本$\boldsymbol{x}_j\in\boldsymbol{D}$的簇标记，$\lambda_j\in\{1,2,\ldots,k\}$

数据集$\boldsymbol{D}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}$的标签集合$\boldsymbol{\lambda}=\{\lambda_1,\ldots,\lambda_m\}$

聚类有很多典型应用，如：

- 相似功能的基因分组
- 相似政见的个体划分
- 相似主题的文档划分

…

聚类任务的遵循步骤及有关问题

- 特征选择及样本描述

  选择什么样的特征？是否需要规范化预处理？

- 近邻测度

  如何度量样本之间如何“相似”或“相异”

- 聚类准则

  依赖于专家对“可判别”的解释，聚类准则应以蕴涵于数据集内类的类型为基础。

- 聚类算法

  选择特定的算法，用于揭示数据集的聚类结构

- 聚类性能评价、结果的解释

近邻测度与聚类准则

A. 向量(点)之间的测度

(1)相异性度量-- dist

$d: \ \boldsymbol{X} \times \boldsymbol{X} \to \Re$

对于$\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \boldsymbol{X}$, $dist(\bullet,\bullet)$须满足：

$$\begin{cases} \text{非负性：} \forall \boldsymbol{x}, \boldsymbol{y}, \ dist(\boldsymbol{x}, \boldsymbol{y}) \geq 0 \\ \text{同一性：} dist(\boldsymbol{x}, \boldsymbol{y}) = 0 \text{当且仅当} \boldsymbol{x}=\boldsymbol{y} \\ \text{对称性：} dist(\boldsymbol{x}, \boldsymbol{y}) = dist(\boldsymbol{y}, \boldsymbol{x}) \\ \text{直递性：} dist(\boldsymbol{x}, \boldsymbol{y}) \leq dist(\boldsymbol{x}, \boldsymbol{z}) + dist(\boldsymbol{y}, \boldsymbol{z}) \end{cases}$$

(2)Similarity Metric -- s

$s: \ \boldsymbol{X} \times \boldsymbol{X} \to \Re$

对于$\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \boldsymbol{X}$, $s(\bullet)$须满足：

$$\begin{cases} \exists s_0 \ \ -\infty < s(\boldsymbol{x}, \boldsymbol{y}) \leq s_0 < +\infty \qquad \forall \boldsymbol{x}, \boldsymbol{y} \\ \qquad\qquad s(\boldsymbol{x}, \boldsymbol{x}) = s_0 \\ s(\boldsymbol{x}, \boldsymbol{y}) = s(\boldsymbol{y}, \boldsymbol{x}) \\ s(\boldsymbol{x}, \boldsymbol{z}) s(\boldsymbol{y}, \boldsymbol{z}) \leq [s(\boldsymbol{x}, \boldsymbol{z}) + s(\boldsymbol{y}, \boldsymbol{z})] s(\boldsymbol{x}, \boldsymbol{y}) \end{cases}$$

例：向量之间的几种典型距离度量

有序属性之间的距离

对于$\forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X} \subset \boldsymbol{R}^n$ $\quad \boldsymbol{x} = [x_1 \ \cdots \ x_n]^T \quad \boldsymbol{y} = [y_1 \ \cdots \ y_n]^T$

A. 闵可夫斯基距离(*Minkowski distance*)

$$dist_{mk}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

B. 曼哈顿距离(*Manhattan distance*) $\quad dist_{man}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_1 = \sum_{i=1}^{n} |x_i - y_i|$

C. 欧式距离(*Eculidean distance*) $\quad dist_{ed}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2 = \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)^{\frac{1}{2}}$

D. 切氏距离(*Chebyshev distance*) $\quad dist_{che}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_\infty = \max_{1 \le i \le n} |x_i - y_i|$

E. 马氏距离(*Mahalanobis distance*) $\quad d_{mah}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{y})}$

F. Camberra距离(*Lance* 距离,*Williams* 距离)

$$dist_{cam}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i + y_i|} \qquad x_i, y_i \ge 0 \text{且} x_i + y_i \ne 0$$

例：向量之间的几种典型距离度量

河北师范大学软件学院
Software College of Hebei Normal University

$\boldsymbol{m}_{u,a}$ ——属性**u**上取值为***a***的样本数

$\boldsymbol{m}_{u,a,i}$ ——第***i***簇中，属性**u**上取值为***a***的样本数

k ——样本集划分的聚类簇数目

无序属性之间的距离

[1]无序属性u上两个离散值a,b之间的**VDM距离**(*Value Difference Metric*)

$$\boldsymbol{VDM}_p(a,b)=\sum_{i=1}^{k}\left|\frac{\boldsymbol{m}_{u,a,i}}{\boldsymbol{m}_{u,a}}-\frac{\boldsymbol{m}_{u,b,i}}{\boldsymbol{m}_{u,b}}\right|^p$$

[2]基于混合属性(n_c个有序属性以及$n-n_c$个无序属性)的距离

$$\boldsymbol{MinkovDM}_p(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{u=1}^{n_c}|x_{iu}-x_{ju}|^p+\sum_{u=n_c+1}^{n}\boldsymbol{VDM}_p(x_{iu},x_{ju})\right)^{\frac{1}{p}}$$

[3]**加权闵可夫斯基距离**(***Weighted Minkowski distance***)

$$dist_{wmk}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{u=1}^{n}w_u|x_{iu}-x_{ju}|^p\right)^{\frac{1}{p}}\qquad w_u\geq 0,\qquad \sum_{u=1}^{n}w_u=1$$

例：向量之间的相似性度量

对于$\forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X} \subset \boldsymbol{R}^n$　　$\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T$　$\boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^T$

A. 内积　　$s_{inner}(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i$

B. 余弦相似度　$s_{\cos ine}(\boldsymbol{x}, \boldsymbol{y}) = \dfrac{\boldsymbol{x}^T \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|}$

C. Pearson相关系数　$r_{Pearson}(\boldsymbol{x}, \boldsymbol{y}) = \dfrac{(\boldsymbol{x} - \overline{x}\boldsymbol{1})^T (\boldsymbol{y} - \overline{y}\boldsymbol{1})}{\|\boldsymbol{x} - \overline{\boldsymbol{x}}\| \|\boldsymbol{y} - \overline{\boldsymbol{y}}\|}$

其中　$\overline{x} = \dfrac{\sum_{i=1}^{n} x_i}{n}$　$\overline{y} = \dfrac{\sum_{i=1}^{n} y_i}{n}$

D. Tanimoto测度　$s_T(\boldsymbol{x}, \boldsymbol{y}) = \dfrac{\boldsymbol{x}^T \boldsymbol{y}}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - \boldsymbol{x}^T \boldsymbol{y}}$

B. 样本与集合之间的测度

方式1.集合中所有样本对近邻测度$\wp(x,C)$均有贡献。

设观测样本x, y之间近邻测度为$\wp(x, y)$

则样本x与聚类(或簇)C之间的近邻函数$\wp(x,C)$,可以是:

最大近邻函数 $\wp_{\max}^{ps}(x,C)=\max_{y\in C}\wp(x,y)$

最小近邻函数 $\wp_{\min}^{ps}(x,C)=\min_{y\in C}\wp(x,y)$

平均近邻函数 $\wp_{avg}^{ps}(x,C)=\frac{1}{n_C}\sum_{y\in C}\wp(x,y)$ n_C为集合C的势

方式2.近邻性以样本x与集合C的表示之间的近邻性度量。

--集合的表示通常有点、超平面、超球面等.

C. 两集合之间的近邻函数

设两观测样本$\boldsymbol{x}, \boldsymbol{y}$之间近邻测度为$\wp(\boldsymbol{x}, \boldsymbol{y})$

对于给定的两个向量集合$\boldsymbol{D}_i, \boldsymbol{D}_j$，近邻函数常见：

最大近邻函数 $$\wp_{\max}^{ss}(\boldsymbol{D}_i, \boldsymbol{D}_j) = \max_{\boldsymbol{x}\in\boldsymbol{D}_i, \boldsymbol{y}\in\boldsymbol{D}_j} \wp(\boldsymbol{x}, \boldsymbol{y})$$

最小近邻函数 $$\wp_{\min}^{ss}(\boldsymbol{D}_i, \boldsymbol{D}_j) = \min_{\boldsymbol{x}\in\boldsymbol{D}_i, \boldsymbol{y}\in\boldsymbol{D}_j} \wp(\boldsymbol{x}, \boldsymbol{y})$$

平均近邻函数 $$\wp_{\text{avg}}^{ss}(\boldsymbol{D}_i, \boldsymbol{D}_j) = \frac{1}{n_{D_i} n_{D_j}} \sum_{\boldsymbol{x}\in\boldsymbol{D}_i} \sum_{\boldsymbol{y}\in\boldsymbol{D}_j} \wp(\boldsymbol{x}, \boldsymbol{y})$$

其中$n_{D_i} n_{D_j}$为集合$\boldsymbol{D}_i, \boldsymbol{D}_j$的势

均值近邻函数 $$\wp_{\text{mean}}^{ss}(\boldsymbol{D}_i, \boldsymbol{D}_j) = \wp\left(\boldsymbol{m}_{D_i}, \boldsymbol{m}_{D_j}\right)$$

$\boldsymbol{m}_{D_i}, \boldsymbol{m}_{D_j}$是关于集合$\boldsymbol{D}_i, \boldsymbol{D}_j$的点描述，如均值点、中值等。

其它 $$\wp_{\text{e}}^{ss}(\boldsymbol{D}_i, \boldsymbol{D}_j) = \sqrt{\frac{n_{D_i} n_{D_j}}{n_{D_i} + n_{D_j}}} \wp\left(\boldsymbol{m}_{D_i}, \boldsymbol{m}_{D_j}\right)$$

D.聚类准则

- 类内距离准则
- 类间距离准则
- 基于类内、类间距离的准则函数
- 基于模式与类核的距离的准则函数

聚类的典型实现方式

(1) **基于模型的方法**

--基于概率密度函数估计的方法

(2) **基于数据的方法**

--基于样本间距离或相似性度量的聚类

共性：

(1)将样本集划分为若干子集；

(2)直接解决分类问题，或者将其作为“标注的”**训练样本集**进行后续监督式分类器设计

聚类(clustering)算法的分类

根据算法设计不同，可以是：

(1)Exclusive Clustering

K-means Clustering

(2)Overlapping Clustering

Fuzzy C-means

(3)Hierarchical Clustering

Agglomerative Clustering

Divisive Clustering

(4)Probabilistic Clustering

Mixture of Gaussian

河北师范大学软件学院
Software College of Hebei Normal University

主要内容

1. 聚类的引入
2. 动态聚类
    - 2.1 K-Means Clustering
    - 2.2 Fuzzy K-Means Clustering
    - 2.3 学习向量量化
4. 基于模型的聚类
5. 密度聚类
6. 层次聚类
7. 其它聚类算法

动态聚类的三个要点：

① 选定某种**距离度量**作为样本间的相异性度量

② 确定某个**准则函数**，用于评价聚类结果的质量

③ 给定初始分类方法，以迭代算法找出使准则函数取极值的最好聚类结果

动态聚类过程：

多次迭代,逐步调整类别划分，最终使准则最优。

- C-Means Clustering
- Fuzzy C-Means Clustering
- …

K-Means Clustering 聚类问题描述

输入： 样本集 $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}, \boldsymbol{x}_i = [\boldsymbol{x}_{i1} \quad \cdots \quad \boldsymbol{x}_{id}]^T \in \boldsymbol{R}^d$

要生成的簇的数目$\boldsymbol{k}$

输出： $\boldsymbol{k}$个互不相交的簇$\boldsymbol{C} = \{\boldsymbol{C}_l \mid \boldsymbol{l} = 1, \ldots, \boldsymbol{k}\}$

1. 聚类准则--"最小误差平方和"准则

将样本集D划分成k个簇：$D = C_1 \cup \cdots \cup C_k$

误差平方和目标函数 $E(\mu_1, \ldots, \mu_k) = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$

其中 C_i --第i个簇(聚类)，$i = 1, \ldots, k$　　$k \ll m$

N_i --第i个簇的样本数目

μ_i --第i个聚类中心，$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$

注：$\{\mu_i,\ i = 1, \ldots, k\}$ --$codebook$

$\sum_{x \in C_i} \|x - \mu_i\|^2$ -- 簇内样本紧致性的度量

河北师范大学软件学院
Software College of Hebei Normal University

2. 算法实现

初始化方式

交叉迭代的动态更新

$$\left\{\mu_i^{(j)},\quad i=1,\ldots,k\right\}$$

$$\to\left\{C_i^{(j+1)},\quad i=1,\ldots,k\right\}$$

$$\to\left\{\mu_i^{(j+1)},\quad i=1,\ldots,k\right\}$$

$$\to\ldots$$

算法终止条件

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

过程:

- 初始化方式
- 交叉迭代的动态更新方式
- 迭代终止条件

```
从 D 中随机选择 k 个样本作为初始均值向量 {μ_1, μ_2, ..., μ_k}
repeat
    令 C_i = ∅ (1 ⩽ i ⩽ k)
    for j = 1, 2, ..., m do
        计算样本 x_j 与各均值向量 μ_i (1 ⩽ i ⩽ k) 的距离: d_ji = ||x_j − μ_i||_2;
        根据距离最近的均值向量确定 x_j 的簇标记: λ_j = arg min_{i∈{1,2,...,k}} d_ji;
        将样本 x_j 划入相应的簇: C_{λ_j} = C_{λ_j} ⋃ {x_j};
    end for
    for i = 1, 2, ..., k do
        计算新均值向量: μ'_i = (1/|C_i|) Σ_{x∈C_i} x;
        if μ'_i ≠ μ_i then
            将当前均值向量 μ_i 更新为 μ'_i
        else
            保持当前均值向量不变
        end if
    end for
until 当前均值向量均未更新
```

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

A. 初始化方式之一 --聚类中心的初始化

"代表点"选择

样本集D划分之前，先选择代表点作为初始聚类核心，

再将其余样本初始分类。

迭代结果与初始代表点选择有关

几种"代表点"的选择方法：

--经验选择

--将全部数据随机分成k类，以每类重心作为该类代表点

--"密度法"选择代表点

--按照样本自然顺序或随机排序，用前k个样本点作为代表点

--用$(k-1)$聚类划分求k个代表点

B. 初始化方式之二 --样本集的初始划分

选定代表点后，即可进行初始划分。

几种初始分类的方法:

(a)直接划分初始分类

(b)**先将数据标准化，基于各样本特征之和的初始分类。**

设**标准化**后的样本集$\mathcal{Y}^N \subset \mathcal{R}^d$， $\boldsymbol{y}_i \in \mathcal{Y}^m, \boldsymbol{i}=1,\ldots,\boldsymbol{m}$

$$\boldsymbol{y}_i = \begin{bmatrix} y_{i1} & \cdots & y_{id} \end{bmatrix}^T$$

$$\boldsymbol{SUM}(\boldsymbol{i}) = \sum_{j=1}^{d} y_{ij}$$

$$\boldsymbol{MA} = \max_i \boldsymbol{SUM}(\boldsymbol{i})$$

$$\boldsymbol{MI} = \min_i \boldsymbol{SUM}(\boldsymbol{i})$$

欲将样本集划分为$\boldsymbol{k}$类，若对每个样本$\boldsymbol{y}_i$

$$\boldsymbol{j} \Leftarrow \operatorname{int}\left\{\frac{(\boldsymbol{k}-1)\cdot(\boldsymbol{SUM}(\boldsymbol{i})-\boldsymbol{MI})}{\boldsymbol{MA}-\boldsymbol{MI}}+1+0.5\right\}$$

则将样本$\boldsymbol{y}_i$归入第$\boldsymbol{j}$类， $\boldsymbol{j}=1,\ldots,\boldsymbol{k}$

C. 聚类数k的确定

通常要求事先给定**聚类数k**.

若类别数目未知，可按如下方法确定

(a)**一般根据领域先验知识确定**;

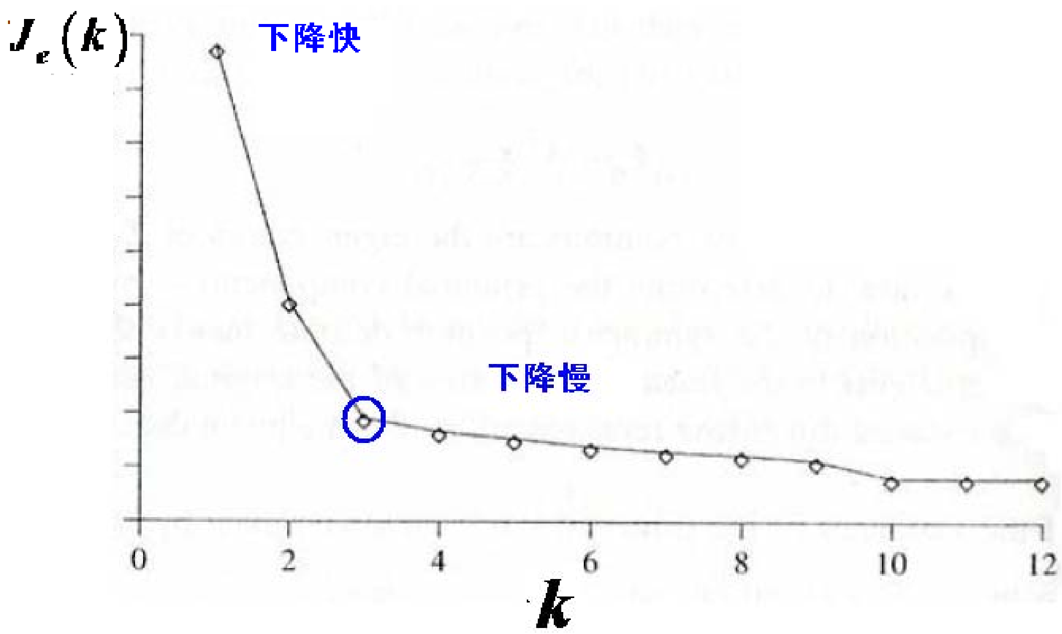
(b)**实验确定**:

令$k=1,2,3,\ldots.,$分别进行聚类，得$J_e(k)$,

绘制$J_e(k)$-k曲线图;

找出拐点，对应聚类

数目为最终类别数。



该方法并不

总是有效。

D. 其它

是否需要进行样本集的规范化预处理？--是

是否需要重复多次？

--从多个聚类结果中选择最好的那个

任意形状的聚类都可处理吗?--**不**

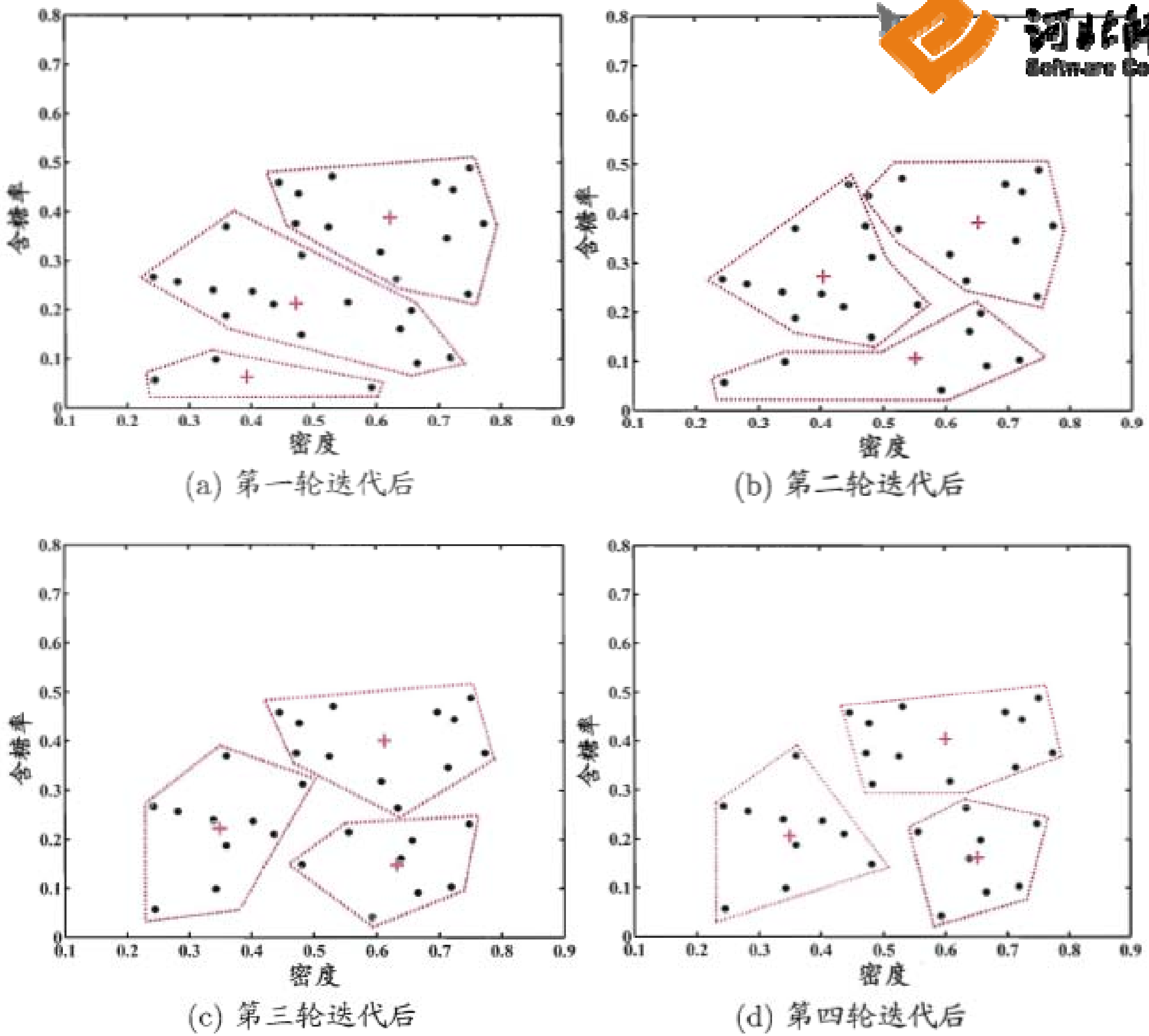


图 9.3 西瓜数据集 4.0 上 k 均值算法($k=3$)在各轮迭代后的结果. 样本点与均值向量分别用"●"与"+"表示, 红色虚线显示出簇划分.

主要内容

1. 聚类的引入
2. 性能度量
3. 动态聚类
    3.1 K-Means Clustering
    3.2 Fuzzy K-Means Clustering
    3.3 学习向量量化
4. 基于模型的聚类
5. 密度聚类
6. 层次聚类
7. 其它聚类算法

学习向量量化(*Learning Vector Quantization, LVQ*)

问题描述

输入： 样本集 $\boldsymbol{D}=\{(\boldsymbol{x}_1, y_1),\ldots,(\boldsymbol{x}_m, y_m)\}$

要生成的原型向量数目$\boldsymbol{q}$

其中 $\boldsymbol{x}_i=[\boldsymbol{x}_{i1} \quad \cdots \quad \boldsymbol{x}_{id}]^T \in \boldsymbol{X} \subset \boldsymbol{R}^d$

$\boldsymbol{y}_i \in \boldsymbol{Y}$

输出：

(1)$\boldsymbol{q}$个原型向量$\{\boldsymbol{p}_l \mid \boldsymbol{l}=1,\ldots,\boldsymbol{q}\}$,

每个原型向量代表一个聚类簇，簇的标记值$\boldsymbol{t}_l \in \boldsymbol{Y}$

(2)基于学得的原型向量，对原始空间进行划分

第一阶段,原型向量的学习

输入: 样本集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
原型向量个数 q, 各原型向量预设的类别标记 $\{t_1, t_2, \ldots, t_q\}$;
学习率 $\eta \in (0, 1)$.

过程:

```
初始化一组原型向量 {p_1, p_2, ..., p_q}
repeat
    从样本集 D 随机选取样本 (x_j, y_j);
    计算样本 x_j 与 p_i (1 ⩽ i ⩽ q) 的距离: d_ji = ||x_j − p_i||_2;
    找出与 x_j 距离最近的原型向量 p_i*, i* = arg min_{i∈{1,2,...,q}} d_ji;
    if y_j = t_i* then
        p' = p_i* + η · (x_j − p_i*)
    else
        p' = p_i* − η · (x_j − p_i*)
    end if
    将原型向量 p_i* 更新为 p'
until 满足停止条件
```

输出: 原型向量 $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_q\}$

若$\boldsymbol{p}_{i*}$与$\boldsymbol{x}_j$**类别一致**，则$\boldsymbol{p}_{i*}$更新后为 $\boldsymbol{p}'=\boldsymbol{p}_{i*}+\eta\left(\boldsymbol{x}_j-\boldsymbol{p}_{i*}\right)$

$$\left\|\boldsymbol{p}'-\boldsymbol{x}_j\right\|_2=\left\|\boldsymbol{p}_{i*}+\eta\left(\boldsymbol{x}_j-\boldsymbol{p}_{i*}\right)-\boldsymbol{x}_j\right\|_2$$

$$=\left\|(1-\eta)\,\boldsymbol{p}_{i*}-(1-\eta)\,\boldsymbol{x}_j\right\|_2$$

$$=(1-\eta)\left\|\boldsymbol{p}_{i*}-\boldsymbol{x}_j\right\|_2<\left\|\boldsymbol{p}_{i*}-\boldsymbol{x}_j\right\|_2$$

若$\boldsymbol{p}_{i*}$与$\boldsymbol{x}_j$**类别不一致**，则$\boldsymbol{p}_{i*}$更新后为 $\boldsymbol{p}'=\boldsymbol{p}_{i*}-\eta\left(\boldsymbol{x}_j-\boldsymbol{p}_{i*}\right)$

$$\left\|\boldsymbol{p}'-\boldsymbol{x}_j\right\|_2=\left\|\boldsymbol{p}_{i*}-\eta\left(\boldsymbol{x}_j-\boldsymbol{p}_{i*}\right)-\boldsymbol{x}_j\right\|_2$$

$$=\left\|(1+\eta)\,\boldsymbol{p}_{i*}-(1+\eta)\,\boldsymbol{x}_j\right\|_2$$

$$=(1+\eta)\left\|\boldsymbol{p}_{i*}-\boldsymbol{x}_j\right\|_2>\left\|\boldsymbol{p}_{i*}-\boldsymbol{x}_j\right\|_2$$

第二阶段,基于原型向量的样本空间簇划分。

q个原型向量$\{\boldsymbol{p}_l \mid \boldsymbol{l}=1,\ldots,\boldsymbol{q}\}$

$$\forall \boldsymbol{x} \in \boldsymbol{X} \subset \boldsymbol{R}^d$$

$$\boldsymbol{R}_i=\left\{\boldsymbol{x} \in \boldsymbol{X} \mid \left\|\boldsymbol{x}-\boldsymbol{p}_i\right\|_2 \leq \left\|\boldsymbol{x}-\boldsymbol{p}_{i'}\right\|_2, \boldsymbol{i}' \neq \boldsymbol{i}\right\}$$

$$\boldsymbol{X}=\boldsymbol{R}_1 \cup \boldsymbol{R}_2 \cup \cdots \cup \boldsymbol{R}_q$$

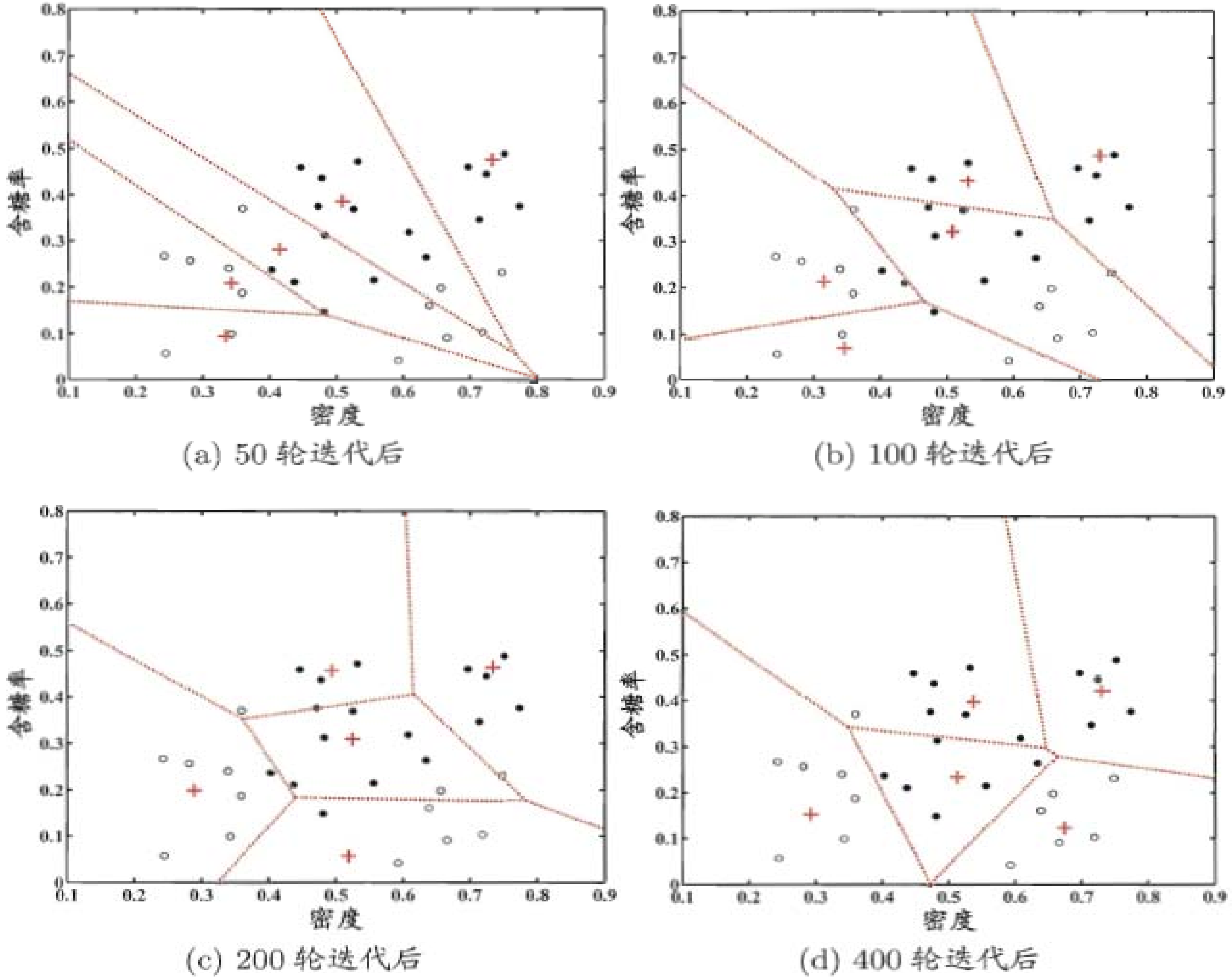


图 9.5 西瓜数据集 4.0 上 LVQ 算法($q=5$)在不同轮数迭代后的聚类结果. c_1, c_2 类样本点与原型向量分别用“●”, “○”与“+”表示, 红色虚线显示出聚类形成的 Voronoi 剖分.

主要内容

1. 聚类的引入
2. 动态聚类
3. 基于模型的聚类

   高斯混合聚类(高斯混合模型+贝叶斯决策)

   Gaussian Mixture Model, GMM

5. 密度聚类
6. 层次聚类
7. 其它聚类算法

问题描述

输入：样本集 $D=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}$, $\boldsymbol{x}_i=[x_{i1} \quad \cdots \quad x_{id}]^T \in \boldsymbol{R}^d$

要生成的簇的数目 $\boldsymbol{k}$

输出：以 $\boldsymbol{k}$ 个**高斯模型的混合**描述的簇 $\boldsymbol{C}=\{\boldsymbol{C}_l \mid \boldsymbol{l}=1,\ldots,\boldsymbol{k}\}$

混合高斯模型 $p_M(\boldsymbol{x})=\sum_{j=1}^{k}\alpha_i p(\boldsymbol{x} \mid \boldsymbol{\mu}_i,\Sigma_i)$

第 $\boldsymbol{i}$ 个**高斯成分**的概率密度函数(第 $\boldsymbol{i}$ 类的条件概率密度函数)

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}_i,\Sigma_i)=\frac{1}{(2\pi)^{d/2}\left|\Sigma_i\right|^{1/2}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right]$$

高斯成分的混合系数： 第 $\boldsymbol{i}$ 个高斯成分的先验概率 α_i $\quad \sum_{j=1}^{k}\alpha_j=1$

参数集合： $\{(\alpha_i,\boldsymbol{\mu}_i,\Sigma_i) \mid \boldsymbol{i}=1,\ldots,\boldsymbol{k}\}$

河北师范大学软件学院

Software College of Hebei Normal University

D中的每个观测样本按照相同的混合概率密度函数独立抽取，样本集D满足**独立、同分布**。

混合概率密度函数的形式已知，但参数未知。

利用样本集D估计已知形式的概率密度函数的未知参数，就叫做**概率密度函数的参数估计问题**。

由于D内每个样本都是无标签的，若采用无类标签的样本数据，进行概率密度函数的参数估计，就是**概率密度函数的非监督式参数估计，常采用最大似然估计法.**

然后利用估计得到的混合高斯模型，对数据集D划分。

高斯混合聚类的问题分解

STEP1. 高斯混合模型的非监督式参数估计.

基于样本集 $D=\{x_1,\ldots,x_m\}$ 估计混合高斯模型的参数:

$$\boldsymbol{\alpha}=\{\alpha_1,\ldots,\alpha_k\}$$

$$\boldsymbol{\mu}=\{\mu_1,\ldots,\mu_k\} \qquad \Sigma=\{\Sigma_1,\ldots,\Sigma_k\}$$

$$p_M(x;\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma)=\sum_{j=1}^{k}\alpha_i p(x\mid\mu_i,\Sigma_i)$$

STEP2. 基于最大后验概率的样本集划分.

$$P_M(z_j=i\mid x_j)=\frac{P(z_j=i)\,p(x_j\mid z_j=i)}{p_M(x_j)}=\frac{\alpha_i p(x_j\mid\mu_i,\Sigma_i)}{\sum_{l=1}^{k}\alpha_l p(x\mid\mu_l,\Sigma_l)}$$

$$\lambda_j=\mathop{\arg\max}_{i\in\{1,2,\ldots,k\}} P_M(z_j=i\mid x_j) \qquad j=1,\ldots,m$$

1.样本集$D=\{x_1,\ldots,x_m\}$的随机抽取

假设样本集$\boldsymbol{D}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}$中的各样本按照独立**同分布**的方式随机抽取得到.

$$p_M(\boldsymbol{x};\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma)=\sum_{j=1}^{k}\alpha_j p(\boldsymbol{x}\mid\boldsymbol{\mu}_j,\Sigma_j)$$

抽取过程可以分解为：

for $i=1,\ldots,m$ 抽取样本$\boldsymbol{x}_i\in\boldsymbol{D}$:

按照$\alpha_1,\ldots,\alpha_k$定义的先验分布，随机选择相应的高斯成分，其中α_j为第$\boldsymbol{j}$个高斯成分出现的先验概率;

按照被选中成分的高斯分布，进行随机抽取得到观测样本$\boldsymbol{x}_i$.

2.似然函数及对数似然函数

(1)观测样本$\boldsymbol{x}_i \in \boldsymbol{D}$的**似然值**

$$p_M\left(\boldsymbol{x}_i;\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)=\sum_{j=1}^{k}\alpha_j p\left(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j,\Sigma_j\right)$$

(2)**样本集**$\boldsymbol{D}$的**似然值**(***likelihood***，或$\boldsymbol{n}$个样本的联合似然值)

$\boldsymbol{D}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}$内各样本独立抽取，似然值：

$$L\left(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)=\prod_{i=1}^{m}p_M\left(\boldsymbol{x}_i;\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)=\prod_{i=1}^{m}\left\{\sum_{j=1}^{k}\alpha_j p\left(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j,\Sigma_j\right)\right\}$$

(3)**样本集**$\boldsymbol{D}$的**对数似然值**(***loglikelihood***)

$$\boldsymbol{H}\left(\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)=LL\left(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)=\ln L\left(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)=\ln\left\{\prod_{i=1}^{m}p_M\left(\boldsymbol{x}_i;\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)\right\}$$

$$=\sum_{i=1}^{m}\ln\left\{p_M\left(\boldsymbol{x}_i;\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma\right)\right\}=\sum_{i=1}^{m}\ln\left\{\sum_{j=1}^{k}\alpha_j p\left(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j,\Sigma_j\right)\right\}$$

3.高斯混合模型的学习--最大似然估计

样本集D的**对数似然值**$(loglikelihood)$

$$LL(D;\alpha,\mu,\Sigma)=\sum_{i=1}^{m}\ln\{p_M(x_i;\alpha,\mu,\Sigma)\}=\sum_{i=1}^{m}\ln\left\{\sum_{j=1}^{k}\alpha_j p(x_i\mid\mu_j,\Sigma_j)\right\}$$

因样本集D为给定信息，对数似然值的大小取决于参数集合α,μ,Σ，因此称$H(\alpha,\mu,\Sigma)$为基于样本集D的**对数似然函数**.

最大似然估计$(Maximum\ \ Likelihood\ \ Estimation, MLE)$：

$$\{\hat{\alpha},\hat{\mu},\hat{\Sigma}\}=\underset{\alpha,\mu,\Sigma}{\arg\max}\, L(D;\alpha,\mu,\Sigma)=\underset{\alpha,\mu,\Sigma}{\arg\max}\, LL(D;\alpha,\mu,\Sigma)$$

$$=\underset{\alpha,\mu,\Sigma}{\arg\max}\sum_{i=1}^{m}\ln\left\{\sum_{j=1}^{k}\alpha_j p(x_i\mid\mu_j,\Sigma_j)\right\}$$

最优化问题：

$$\max_{\alpha,\mu,\Sigma} \sum_{i=1}^{m} \ln\left\{\sum_{j=1}^{k} \alpha_j \boldsymbol{p}\left(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j\right)\right\}$$

$$\text{s.t.} \quad \alpha_i \geq 0 \qquad \sum_{j=1}^{k} \alpha_j = 1$$

选择何种最优化方法估计上述目标函数的最优值解？
EM算法(交叉迭代，“E步+M步”)

4.最大似然估计的具体实现过程

最优化问题 $\begin{cases} \max\limits_{\alpha,\mu,\Sigma} \sum_{j=1}^{m} \ln\left\{\sum_{i=1}^{k} \alpha_i p\left(x_j \mid \mu_i, \Sigma_i\right)\right\} \\ \text{s.t.} \quad \alpha_i \geq 0 \qquad \sum_{i=1}^{k} \alpha_i = 1 \end{cases}$

$$LL\left(D;\alpha,\mu,\Sigma\right) = \sum_{j=1}^{m} \ln\left\{\sum_{i=1}^{k} \alpha_i p\left(x_j \mid \mu_i, \Sigma_i\right)\right\}$$

$$= \sum_{j=1}^{m} \ln\left\{\sum_{i=1}^{k} \frac{\alpha_i}{(2\pi)^{d/2}\left|\Sigma_i\right|^{1/2}} \exp\left[-\frac{1}{2}(x_j - \mu_i)^T \Sigma_i^{-1}(x_j - \mu_i)\right]\right\}$$

$$\nabla LL\left(D;\alpha,\mu,\Sigma\right) = ?$$

$$\frac{\partial}{\partial \mu_i} LL\left(D;\alpha,\mu,\Sigma\right) = ? \quad \frac{\partial}{\partial \Sigma_i} LL\left(D;\alpha,\mu,\Sigma\right) = ? \quad \frac{\partial}{\partial \alpha_i} LL\left(D;\alpha,\mu,\Sigma\right) = ?$$

河北师范大学软件学院
Software College of Hebei Normal University

$$\boxed{\frac{\partial}{\partial \boldsymbol{\mu}_i} LL\left(\mathbf{D};\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=?}$$

由于 $LL\left(\mathbf{D};\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\sum_{j=1}^{m}\ln\left\{\sum_{i=1}^{k}\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\right\}$

并且 $\boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)=\frac{1}{(2\pi)^{d/2}\left|\Sigma_i\right|^{1/2}}\exp\left[-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right]$

$$\frac{\partial}{\partial \boldsymbol{\mu}_i}\boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)=\boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\Sigma_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)$$

所以 $\frac{\partial}{\partial \boldsymbol{\mu}_i} LL\left(\mathbf{D};\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\sum_{j=1}^{m}\frac{\partial}{\partial \boldsymbol{\mu}_i}\ln\left\{\sum_{i=1}^{k}\alpha_j \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\right\}$

$$=\sum_{j=1}^{m}\frac{\sum_{i=1}^{k}\alpha_i\frac{\partial}{\partial \boldsymbol{\mu}_i}\boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}{\sum_{i=1}^{k}\alpha_j \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}=\sum_{i=1}^{m}\frac{\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\left[\Sigma_i^{-1}\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)\right]}{\sum_{i=1}^{k}\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}$$

$$=\sum_{i=1}^{m}\boldsymbol{P}_M\left(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j\right)\left[\Sigma_i^{-1}\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)\right]$$

令 $\frac{\partial}{\partial \boldsymbol{\mu}_i} LL(\boldsymbol{D}; \boldsymbol{\alpha}, \boldsymbol{\mu}, \Sigma) = \sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)\left[\Sigma_i^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)\right] = 0$

两边左乘Σ_i，有 $\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)\left[(\boldsymbol{x}_j - \boldsymbol{\mu}_i)\right] = 0$

即 $\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)\boldsymbol{x}_j = \sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)\boldsymbol{\mu}_i$

$$\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)\boldsymbol{x}_j = \left[\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)\right]\boldsymbol{\mu}_i$$

所以

$$\boxed{\hat{\boldsymbol{\mu}}_i = \frac{\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j = \boldsymbol{i} \mid \boldsymbol{x}_j)} = \frac{\sum_{j=1}^{m} \gamma_{ji}\boldsymbol{x}_j}{\sum_{j=1}^{m} \gamma_{ji}} \qquad \boldsymbol{i} = 1, \ldots, \boldsymbol{k}}$$

西瓜书209页式(9.34)

$$\boxed{\frac{\partial}{\partial \Sigma_i} LL(\mathbf{D};\alpha,\mu,\Sigma) = ?}$$

由于 $LL(\mathbf{D};\alpha,\mu,\Sigma) = \sum_{j=1}^{m} \ln \left\{ \sum_{i=1}^{k} \alpha_i \mathbf{p}(x_j \mid \mu_i, \Sigma_i) \right\}$

并且 $\mathbf{p}(x_j \mid \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2} (x_j - \mu_i)^T \Sigma_i^{-1} (x_j - \mu_i) \right]$

$$\frac{\partial}{\partial \Sigma_i}(|\Sigma_i|) = |\Sigma_i| (\Sigma_i^{-1})^T = |\Sigma_i| \Sigma_i^{-T} = |\Sigma_i| \Sigma_i^{-1}$$

$$\frac{\partial}{\partial \Sigma_i}(a^T \Sigma_i^{-1} b) = \frac{\partial}{\partial \Sigma_i} tr(a^T \Sigma_i^{-1} b) = -\Sigma_i^{-1} b a^T \Sigma_i^{-1}$$

$$\frac{\partial}{\partial \Sigma_i}\left((x_j - \mu_i)^T \Sigma_i^{-1} (x_j - \mu_i)\right) = -\Sigma_i^{-1} (x_j - \mu_i)(x_j - \mu_i)^T \Sigma_i^{-1}$$

河北师范大学软件学院
Software College of Hebei Normal University

$$\frac{\partial}{\partial \Sigma_i} p\left(x_j \mid \mu_i, \Sigma_i\right)$$

$$= \frac{-\frac{1}{2}\left|\Sigma_i\right|^{-3/2}\left|\Sigma_i\right|\Sigma_i^{-1}}{(2\pi)^{d/2}} \exp\left[-\frac{1}{2}(x_j-\mu_i)^T \Sigma_i^{-1}(x_j-\mu_i)\right]$$
$$+\frac{1}{(2\pi)^{d/2}\left|\Sigma_i\right|^{1/2}} \exp\left[-\frac{1}{2}(x_j-\mu_i)^T \Sigma_i^{-1}(x_j-\mu_i)\right]\left[\frac{1}{2}\Sigma_i^{-1}\left(x_j-\mu_i\right)\left(x_j-\mu_i\right)^T \Sigma_i^{-1}\right]$$

$$= -\frac{1}{2} p\left(x_j \mid \mu_i, \Sigma_i\right)\Sigma_i^{-1} + \frac{1}{2} p\left(x_j \mid \mu_i, \Sigma_i\right)\left[\Sigma_i^{-1}\left(x_j-\mu_i\right)\left(x_j-\mu_i\right)^T \Sigma_i^{-1}\right]$$

$$= \frac{1}{2} p\left(x_j \mid \mu_i, \Sigma_i\right)\Sigma_i^{-1}\left[\left(x_j-\mu_i\right)\left(x_j-\mu_i\right)^T \Sigma_i^{-1} - I\right]$$

由于 $LL(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\Sigma})=\sum_{j=1}^{m}\ln\left\{\sum_{i=1}^{k}\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\right\}$

所以 $$\frac{\partial}{\partial \Sigma_i}LL(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\Sigma})=\sum_{j=1}^{m}\frac{\partial}{\partial \Sigma_i}\ln\left\{\sum_{i=1}^{k}\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\right\}=\sum_{j=1}^{m}\frac{\alpha_i\frac{\partial}{\partial \Sigma_i}\boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}{\sum_{i=1}^{k}\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}$$

$$=\sum_{j=1}^{m}\frac{\alpha_i\frac{1}{2}\boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\Sigma_i^{-1}\left[\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)^{T}\Sigma_i^{-1}-\boldsymbol{I}\right]}{\sum_{i=1}^{k}\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}$$

$$=\frac{1}{2}\sum_{j=1}^{m}\frac{\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\Sigma_i^{-1}\left[\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)^{T}\Sigma_i^{-1}-\boldsymbol{I}\right]}{\sum_{i=1}^{k}\alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}$$

$$=\frac{1}{2}\sum_{j=1}^{m}\boldsymbol{P}_M\left(\boldsymbol{z}_j=\boldsymbol{i} \mid \boldsymbol{x}_j\right)\Sigma_i^{-1}\left[\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)^{T}\Sigma_i^{-1}-\boldsymbol{I}\right]$$

即： $$\boxed{\frac{\partial}{\partial \Sigma_i}LL(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{1}{2}\sum_{j=1}^{m}\boldsymbol{P}_M\left(\boldsymbol{z}_j=\boldsymbol{i} \mid \boldsymbol{x}_j\right)\Sigma_i^{-1}\left[\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i\right)^{T}\Sigma_i^{-1}-\boldsymbol{I}\right]}$$

$$\frac{\partial}{\partial \Sigma_i} LL(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma)=\frac{1}{2}\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)\Sigma_i^{-1}\left[(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}-\boldsymbol{I}\right]$$

令 $\frac{\partial}{\partial \Sigma_i} LL(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma)=0$

则 $\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)\Sigma_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}=\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)\Sigma_i^{-1}$

即 $\Sigma_i^{-1}\left[\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\right]\Sigma_i^{-1}=\left[\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)\right]\Sigma_i^{-1}$

等号两侧同时**左乘**、**右乘**Σ_i

$$\Sigma_i\Sigma_i^{-1}\left[\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\right]\Sigma_i^{-1}\Sigma_i=\Sigma_i\left[\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)\right]\Sigma_i^{-1}\Sigma_i$$

即：

$$\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T=\Sigma_i\sum_{j=1}^{m} \boldsymbol{P}_M(\boldsymbol{z}_j=\boldsymbol{i}\mid \boldsymbol{x}_j)$$

$$\sum_{j=1}^{m} P_M\left(z_j = i \mid x_j\right)\left(x_j - \mu_i\right)\left(x_j - \mu_i\right)^T = \Sigma_i \sum_{j=1}^{m} P_M\left(z_j = i \mid x_j\right)$$

所以 $$\hat{\Sigma}_i = \frac{\sum_{j=1}^{m} P_M\left(z_j = i \mid x_j\right)\left(x_j - \mu_i\right)\left(x_j - \mu_i\right)^T}{\sum_{j=1}^{m} P_M\left(z_j = i \mid x_j\right)}$$

即：

$$\boxed{\hat{\Sigma}_i = = \frac{\sum_{j=1}^{m} \gamma_{ji}\left(x_j - \hat{\mu}_i\right)\left(x_j - \hat{\mu}_i\right)^T}{\sum_{j=1}^{m} \gamma_{ji}} \qquad i = 1, \ldots, k}$$

西瓜书209页式(9.35)

$$\boxed{\frac{\partial}{\partial \alpha_i} LL(\boldsymbol{D};\boldsymbol{\alpha},\boldsymbol{\mu},\Sigma)=?}$$

固定$\boldsymbol{\mu},\Sigma$，估计混合系数$\boldsymbol{\alpha}$，有：
$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} \sum_{j=1}^{m} \ln\left\{\sum_{i=1}^{k} \alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\right\} \\ \text{s.t.} \quad \alpha_i \geq 0 \qquad \sum_{i=1}^{k} \alpha_i = 1 \end{cases}$$

构造***Lagrange***辅助函数

$$\boldsymbol{H}(\boldsymbol{\alpha},\lambda)=\sum_{j=1}^{m} \ln\left\{\sum_{i=1}^{k} \alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)\right\}+\lambda\left[\sum_{i=1}^{k} \alpha_i - 1\right]$$

令$$\frac{\boldsymbol{H}(\boldsymbol{\alpha},\lambda)}{\partial \alpha_i}=\sum_{j=1}^{m} \frac{\boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}{\sum_{i=1}^{k} \alpha_i \boldsymbol{p}\left(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i,\Sigma_i\right)}+\lambda=0 \qquad \boldsymbol{i}=1,\ldots,\boldsymbol{k}$$

由贝叶斯公式知 $$\sum_{j=1}^{m} \frac{\boldsymbol{P}_M\left(\boldsymbol{z}_j=\boldsymbol{i} \mid \boldsymbol{x}_j\right)}{\alpha_i}+\lambda=0 \qquad \boldsymbol{i}=1,\ldots,\boldsymbol{k}$$

$$\sum_{j=1}^{m}\frac{P_M\left(z_j=i\,|\,x_j\right)}{\alpha_i}+\lambda=0 \qquad i=1,\ldots,k$$

则有 $\sum_{j=1}^{m}P_M\left(z_j=i\,|\,x_j\right)=-\alpha_i\lambda \qquad i=1,\ldots,k$

从而有 $\sum_{i=1}^{k}\left[\sum_{j=1}^{m}P_M\left(z_j=i\,|\,x_j\right)\right]=\sum_{i=1}^{k}\left[-\alpha_i\lambda\right]$

$$\sum_{j=1}^{m}\sum_{i=1}^{k}P_M\left(z_j=i\,|\,x_j\right)=-\lambda\sum_{i=1}^{k}\alpha_i$$

由于 $\sum_{i=1}^{k}P_M\left(z_j=i\,|\,x_j\right)=1, \sum_{i=1}^{k}\alpha_i=1$ 所以 $m=-\lambda$

河北师范大学软件学院
Software College of Hebei Normal University

$$\sum_{j=1}^{m} P_M\left(z_j = i \mid x_j\right) = -\alpha_i \lambda \qquad i = 1, \ldots, k$$

$$m = -\lambda$$

$$\sum_{j=1}^{m} P_M\left(z_j = i \mid x_j\right) = \alpha_i m \qquad i = 1, \ldots, k$$

即

$$\hat{\alpha}_i = \frac{1}{m} \sum_{j=1}^{m} P_M\left(z_j = i \mid x_j\right) = \frac{1}{m} \sum_{j=1}^{m} \gamma_{ji}$$

$$i = 1, \ldots, k$$

西瓜书209页式(9.38)

GMM模型参数估计的交叉迭代

(1) **初始化** $\hat{\mu}_i(0), \hat{\Sigma}_i(0), \hat{\alpha}_i(0), i=1,\ldots,k$

(2) **交叉迭代**

$$E-STEP \quad \gamma_{ji}(n+1)=\frac{\alpha_i(n)\,p\left(x_j \mid \mu_i(n), \Sigma_i(n)\right)}{\sum_{l=1}^{k}\alpha_l(n)\,p\left(x \mid \mu_l(n), \Sigma_l(n)\right)} \qquad \begin{cases} i=1,\ldots,k \\ j=1,\ldots,m \end{cases}$$

$$M-STEP \begin{cases} \hat{\mu}_i(n+1)=\dfrac{\sum_{j=1}^{m}\gamma_{ji}(n+1)\,x_j}{\sum_{j=1}^{m}\gamma_{ji}(n+1)} & i=1,\ldots,k \\ \hat{\Sigma}_i(n+1)==\dfrac{\sum_{j=1}^{m}\gamma_{ji}(n+1)\left(x_j-\hat{\mu}_i(n+1)\right)\left(x_j-\hat{\mu}_i(n+1)\right)^T}{\sum_{j=1}^{m}\gamma_{ji}(n+1)} & i=1,\ldots,k \\ \hat{\alpha}_i(n+1)=\dfrac{1}{m}\sum_{j=1}^{m}\gamma_{ji}(n+1) & i=1,\ldots,k \end{cases}$$

输入: 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
高斯混合成分个数 k.

如何初始化GMM模型的参数?

过程:

```
初始化高斯混合分布的模型参数 {(α_i, μ_i, Σ_i) | 1 ⩽ i ⩽ k}
repeat
    for j = 1, 2, ..., m do
        根据式(9.30)计算 x_j 由各混合成分生成的后验概率, 即
        γ_ji = p_M(z_j = i | x_j) (1 ⩽ i ⩽ k)
    end for
    for i = 1, 2, ..., k do
        计算新均值向量: μ'_i = Σ_{j=1}^m γ_ji x_j / Σ_{j=1}^m γ_ji;
        计算新协方差矩阵: Σ'_i = Σ_{j=1}^m γ_ji (x_j − μ'_i)(x_j − μ'_i)^T / Σ_{j=1}^m γ_ji;
        计算新混合系数: α'_i = Σ_{j=1}^m γ_ji / m;
    end for
    将模型参数 {(α_i, μ_i, Σ_i) | 1 ⩽ i ⩽ k} 更新为 {(α'_i, μ'_i, Σ'_i) | 1 ⩽ i ⩽ k}
until 满足停止条件
C_i = ∅ (1 ⩽ i ⩽ k)
for j = 1, 2, ..., m do
    根据式(9.31)确定 x_j 的簇标记 λ_j;
    将 x_j 划入相应的簇: C_{λ_j} = C_{λ_j} ∪ {x_j}
end for
```

其中: $\boldsymbol{\mu}'_i = \frac{\sum_{j=1}^m \gamma_{ji}\boldsymbol{x}_j}{\sum_{j=1}^m \gamma_{ji}}$; $\boldsymbol{\Sigma}'_i = \frac{\sum_{j=1}^m \gamma_{ji}(\boldsymbol{x}_j-\boldsymbol{\mu}'_i)(\boldsymbol{x}_j-\boldsymbol{\mu}'_i)^{\mathrm{T}}}{\sum_{j=1}^m \gamma_{ji}}$; $\alpha'_i = \frac{\sum_{j=1}^m \gamma_{ji}}{m}$

PART1. 基于EM算法的GMM模型的学习

PART2. 根据GMM模型生成最终划分结果.

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

主要内容

1. 聚类的引入
2. 动态聚类
3. 基于模型的聚类　　高斯混合聚类
4. 密度聚类(density-based clustering)

DBSCAN

Density - Based Spatial Clustering of Applications with Noise

6. 层次聚类
7. 其它聚类算法

- DBSCAN算法是一种基于高密度连通区域的、基于密度的聚类算法

- 能够将具有足够高密度的区域划分为簇

- 能够在具有噪声的数据中发现任意形状的簇

- 能够发现异常点

1. 有关概念 $\left[\text{给定样本集} D=\{x_1,\cdots,x_m\}\right]$

[1]两个**全局邻域参数** $(\varepsilon, MinPts)$

ε--邻域最大半径

$MinPts$--给定样本的 $\varepsilon-$**邻域**内最小样本数.

[2] $\varepsilon-$**邻域**

对于 $\forall x_j \in D$， x_j 的 $\varepsilon-$邻域为 $N_\varepsilon(x_j)=\{x_i \in D \mid dist(x_i, x_j) \le \varepsilon\}$

[3]**核心对象** $(core\ \ object)$

若 $|N_\varepsilon(x_j)| \ge MinPts$，则称 x_j 为一个**核心对象**.

[4]**密度直达** $(directly\ \ density-reacheable)$

若 $x_j \in N_\varepsilon(x_i)$，并且 x_i 为一个核心对象，则称 x_j 为由 x_i **密度直达**.

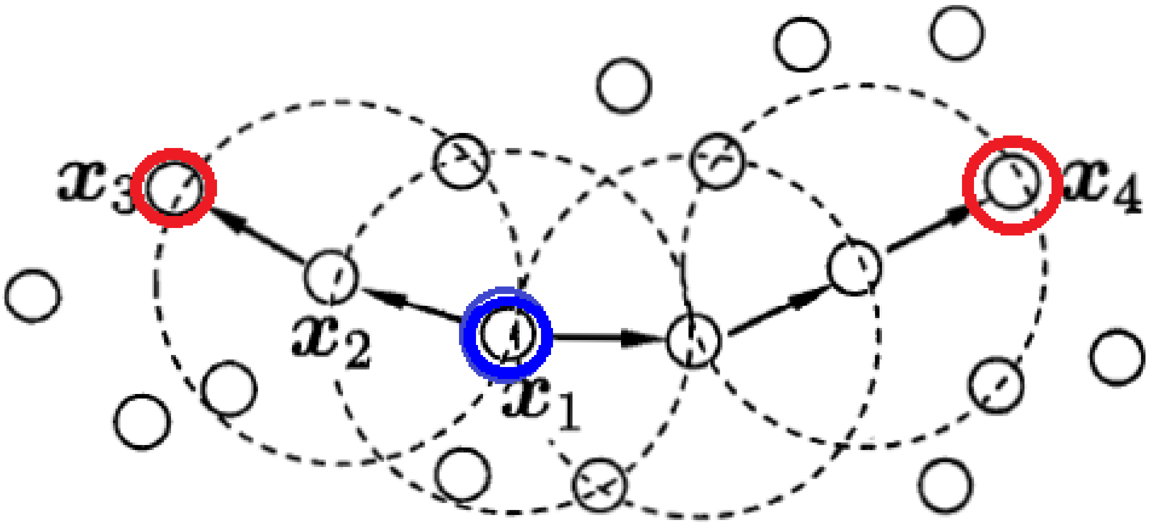
[5]**密度可达** $(density-reacheable)$

对于 x_i, x_j，若存在样本序列 $p_1, p_2, \ldots, p_n$，其中 $p_1=x_i$，$p_n=x_j$，且 p_{i+1} 由 p_i 密度直达，则称 x_j 由 x_i **密度可达**.

[6]**密度相连**$(density-connected)$

对于x_i与x_j，若存在样本x_k,使得x_i与x_j均由x_k**密度可达**，则称x_i与x_j**密度相连**.

邻域参数$(\varepsilon, MinPts)$，$\varepsilon-$邻域，**核心对象**，**密度直达**，**密度可达**，**密度相连**



DBSCAN 定义的基本概念($MinPts = 3$)：虚线显示出 ϵ-邻域，x_1 是核心对象，x_2 由 x_1 密度直达，x_3 由 x_1 密度可达，x_3 与 x_4 密度相连.

$[7]$ **边界对象** $(border\ \ object)$

若 $N_\varepsilon(x_i) \ge MinPts$， $x_j \in N_\varepsilon(x_i)$, 并且 $N_\varepsilon(x_j) < MinPts$,

则称 x_j 为一个边界对象.

边界对象位于聚类簇的边界处.

$[8]$ **噪声对象** $(noise\ \ object)$

核心对象、边界对象以外的其它样本.

$[9]$ **簇**

由**密度可达**关系导出的**最大密度相连样本集**。

给定邻域参数 $(\varepsilon, MinPts)$， 簇 $C \subseteq D$ 为满足以下性质的非空样本子集:

连接性 $(connectivity)$: $x_i, x_j \in C \Rightarrow x_i$ 与 x_j **密度相连**.

最大性 $(maximality)$: $x_i \in C$， 并且 x_j 由 x_i **密度可达** $\Rightarrow x_j \in C$.

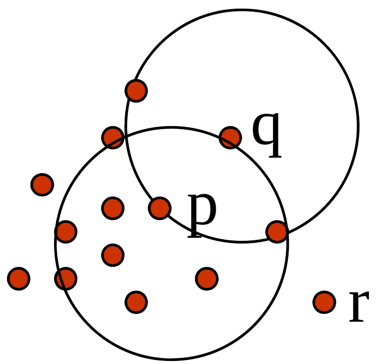
样本集*D*的主要组成

- **核心对象**(聚类簇的种子点)
- **边界对象**(可由核心对象密度可达的非核心对象)
- **噪声对象**(异常点，孤立点)

核心对象、边界对象 → 聚类簇

高密度区

低密度区



MinPts = 5

Eps = 1 cm

2. DBSCAN算法纲要

特点

--无需指定聚类簇的数目;

--只需输入两个全局参数(**全局参数取值如何选？**).

启发

--识别样本集内的核心对象;

--任选一个核心对象，作为一个聚类簇的种子点;

--获取该种子点及其所有密度可达样本，构成一个聚类簇.

- 遍历整个样本集D，确定核心对象、边界对象、及噪声对象
- 将所有噪声对象标记为异常.
- 考查每个核心对象：核心对象彼此可密度直达，则应划分至相同的聚类簇。
- 所有边界对象：若它们是密度相连的，应划分至与其核心对象一致的聚类簇内.

2. 算法描述

STEP1.识别给定样本集D的所有核心对象.

得到核心对象集合 Ω

STEP2. 初始化聚类簇的数目为0；初始化未被访问的样本集为整个数据集D.

STEP3. 重复如下过程，生成一系列聚类簇,直到核心对象集合为空.

- 从核心对象集合中，任选1核心对象，作为聚类簇的一个种子点，找出其密度可达的所有样本，构成1个聚类簇.
- 更新核心对象集合；
- 更新未访问的样本集合

STEP4. 输出所有聚类簇.

输入: 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
邻域参数 $(\epsilon, MinPts)$.

过程:

初始化核心对象集合: $\Omega = \varnothing$
for $j = 1, 2, \ldots, m$ **do**
  确定样本 $\boldsymbol{x}_j$ 的 ϵ-邻域 $N_\epsilon(\boldsymbol{x}_j)$;
  if $|N_\epsilon(\boldsymbol{x}_j)| \geqslant MinPts$ **then**
    将样本 $\boldsymbol{x}_j$ 加入核心对象集合: $\Omega = \Omega \bigcup \{\boldsymbol{x}_j\}$
  end if
end for
初始化聚类簇数: $k = 0$
初始化未访问样本集合: $\Gamma = D$
while $\Omega \neq \varnothing$ **do**
  记录当前未访问样本集合: $\Gamma_{\text{old}} = \Gamma$;
  随机选取一个核心对象 $\boldsymbol{o} \in \Omega$, 初始化队列 $Q = <\boldsymbol{o}>$;
  $\Gamma = \Gamma \setminus \{\boldsymbol{o}\}$;
  while $Q \neq \varnothing$ **do**
    取出队列 Q 中的首个样本 $\boldsymbol{q}$;
    if $|N_\epsilon(\boldsymbol{q})| \geqslant MinPts$ **then**
      令 $\Delta = N_\epsilon(\boldsymbol{q}) \bigcap \Gamma$;
      将 Δ 中的样本加入队列 Q;
      $\Gamma = \Gamma \setminus \Delta$;
    end if
  end while
  $k = k + 1$, 生成聚类簇 $C_k = \Gamma_{\text{old}} \setminus \Gamma$;
  $\Omega = \Omega \setminus C_k$
end while

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

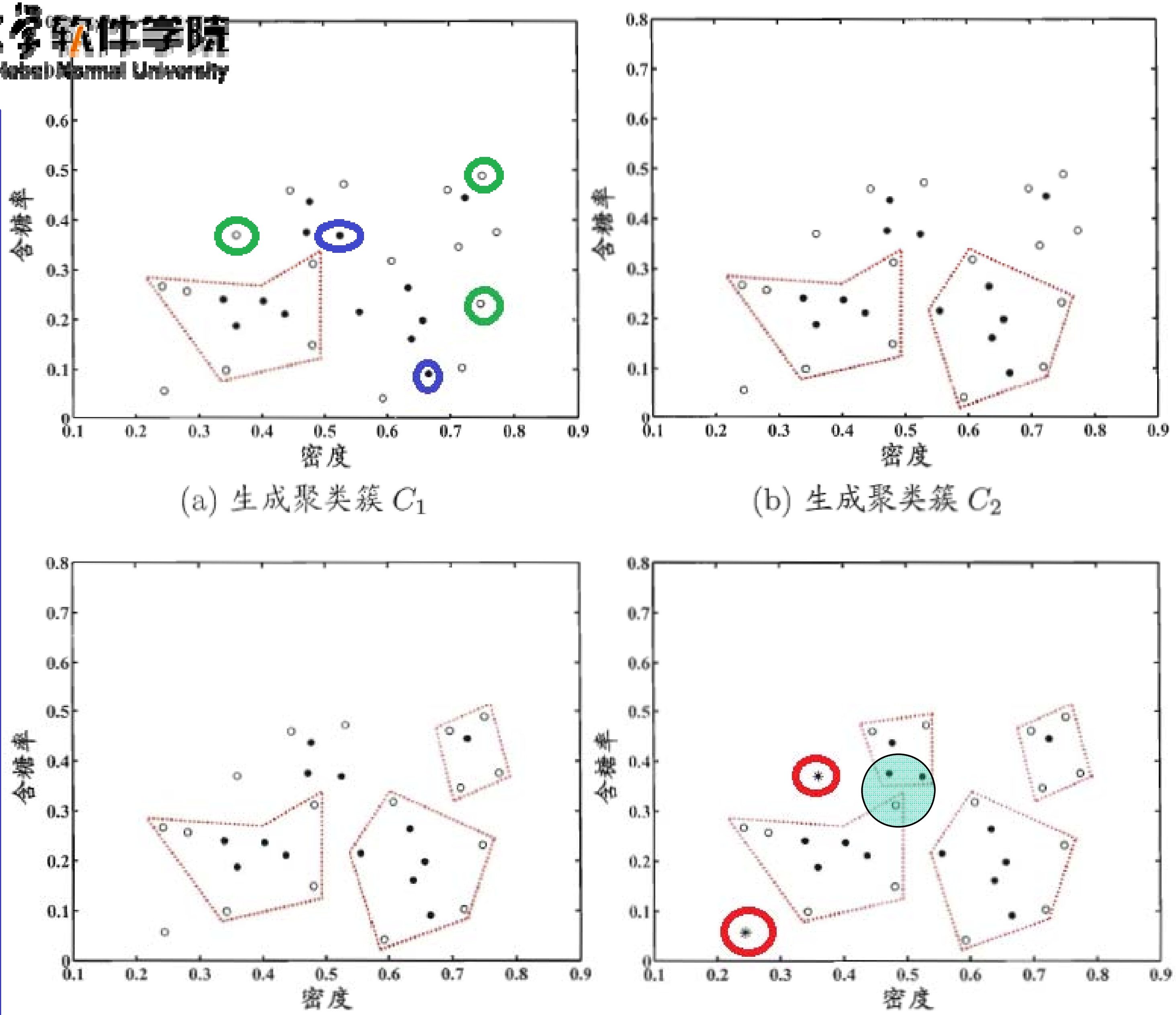
DBSCAN算法的聚类结果

核心对象
非核心对象

密度直达
密度可达
密度相连

聚类簇

噪声或异常点

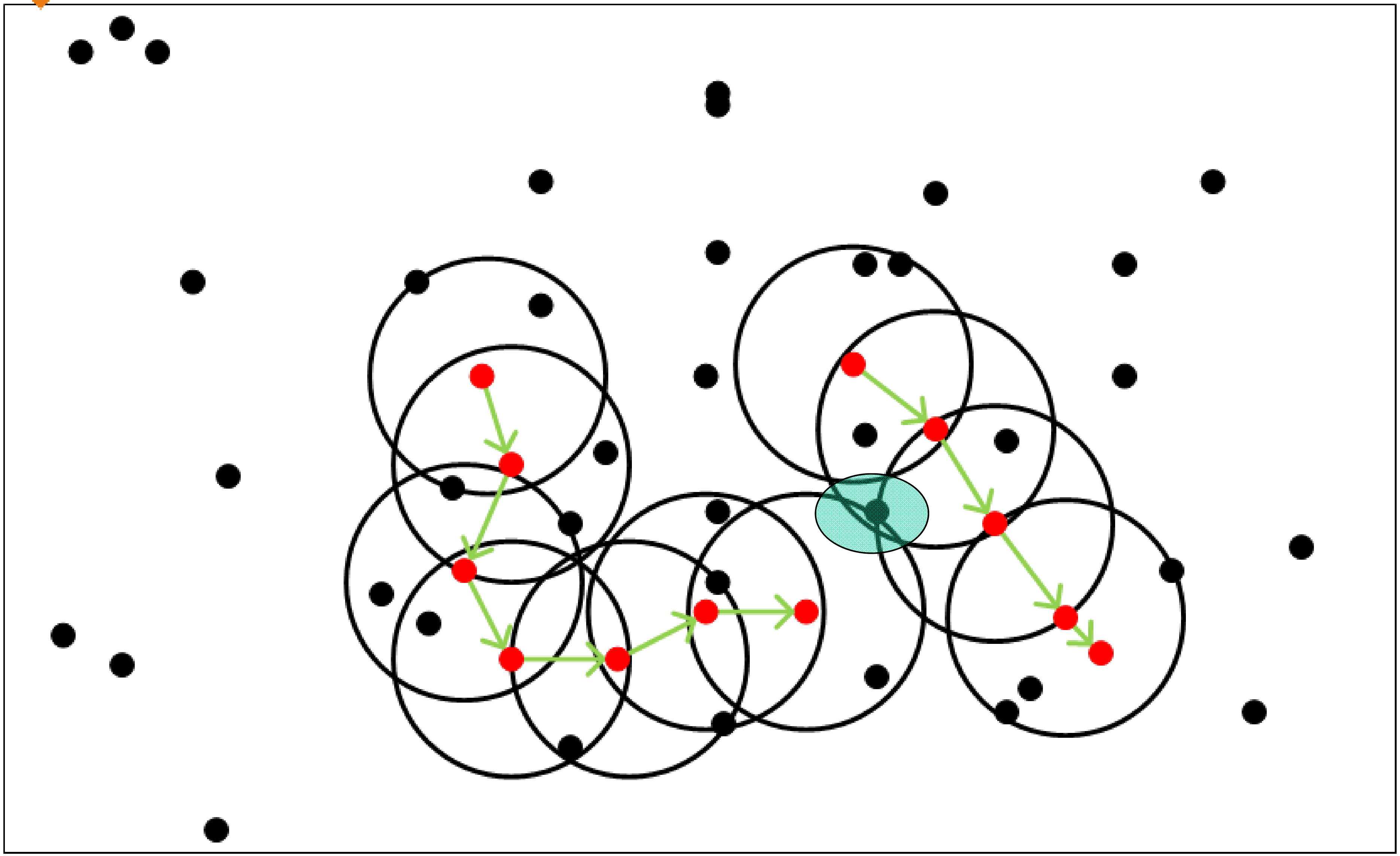


(a) 生成聚类簇 C_1

(b) 生成聚类簇 C_2

(c) 生成聚类簇 C_3

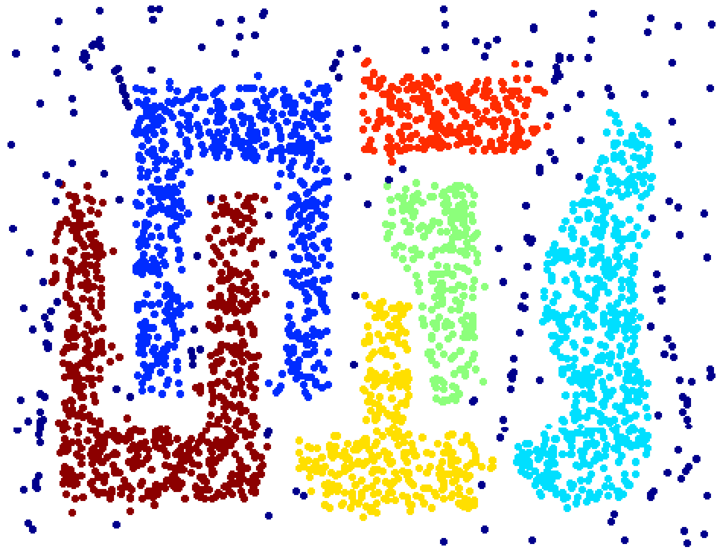
(d) 生成聚类簇 C_4



DBSCAN聚类算法的细节

DBSCAN运行效果好的时候

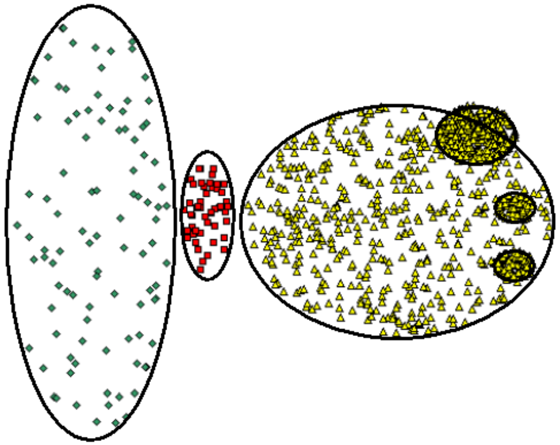
Original Points



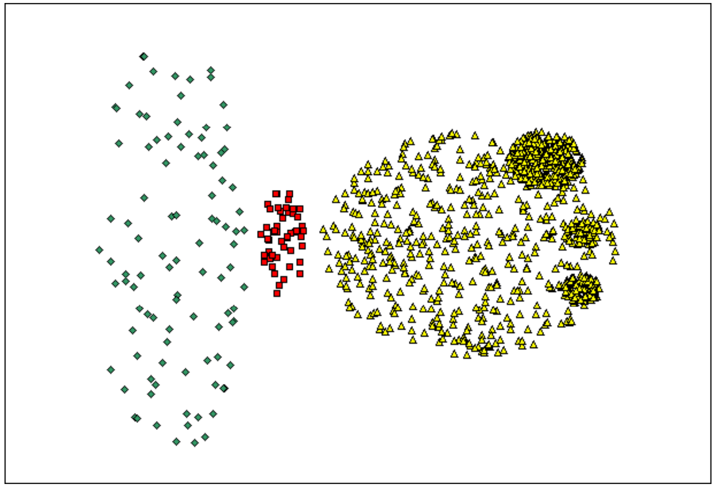
Clusters

- 对噪声不敏感
- 可处理不同形状和大小的数据

DBSCAN运行不好的效果

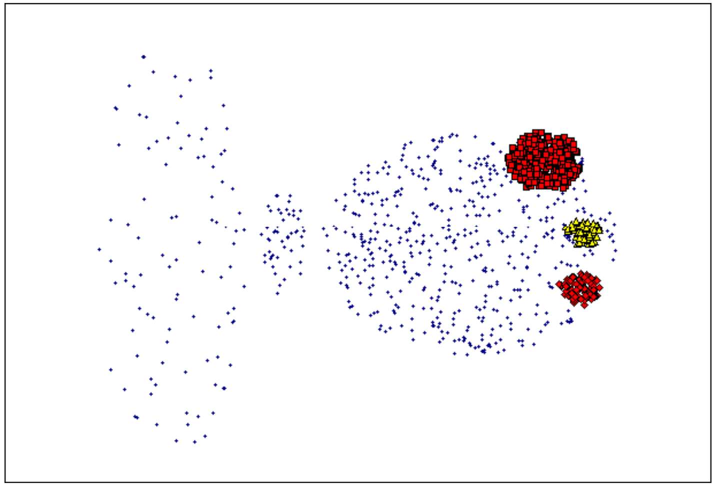


Original Points



(MinPts=4, Eps=9.75).

- 密度变化的数据
- 高维数据



(MinPts=4, Eps=9.92)

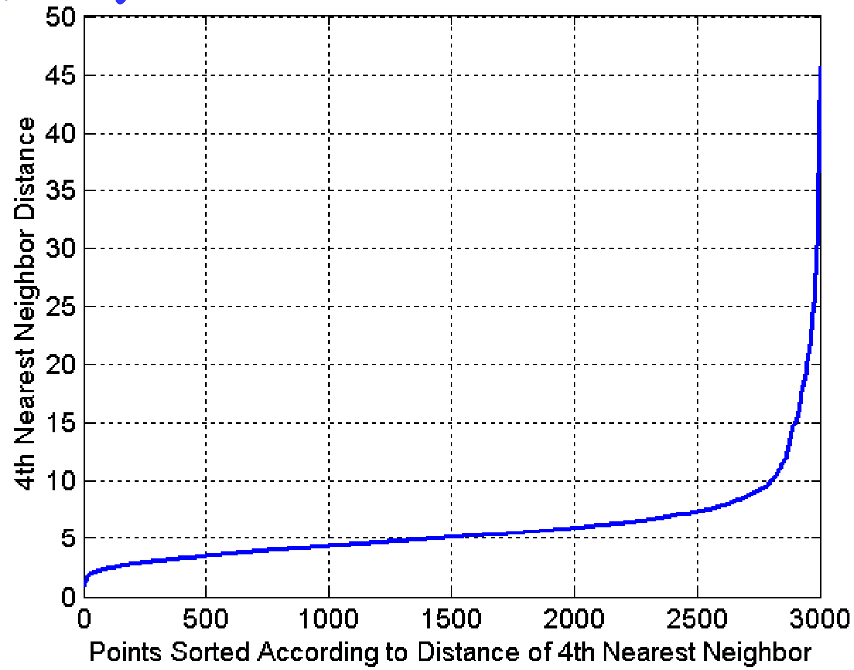
如何适当选取EPS和MinPts
--基于k-距离

- 位于同一聚类簇的所有样本点,它们到其第k个最近邻的距离应大致一样
- 噪声点到其第k个最近邻的距离比较远
- 获取每个样本到其第k个最近邻的距离，从小到大升序排列 得到k-距离变化曲线图
- 找到曲线的拐点(变化剧烈的位置)

- 然后:

  Eps即为变化剧烈位置对应的K-距离.

  MinPts取k



DBSCAN算法的优缺点

- 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

密度分布不均匀、或密度变化较大的簇时，会有麻烦；

邻域半径小，数量多的小簇，核心点数量减小

邻域半径大，本不属于同一簇的样本会聚至相同簇

对于高维问题，密度定义是个比较麻烦的问题

河北师范大学软件学院
Software College of Hebei Normal University

主要内容

1. 聚类的引入
2. 动态聚类
3. 基于模型的聚类　　高斯混合聚类
4. 密度聚类(density-based clustering) DBSCAN
5. 层次聚类

hierarchical clustering

也称系统聚类、分级聚类

6. 其它聚类算法

1.层次聚类的引入

(1)问题描述

数据集$\boldsymbol{D}=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m\}\Rightarrow$划分为合理的$\boldsymbol{k}$个聚类簇。

$\boldsymbol{1}\le \boldsymbol{k}\le \boldsymbol{m}$

$\boldsymbol{k}$值的极端情况$\begin{cases}\text{最多}\boldsymbol{m}\text{簇，每簇包含一个样本}\\ \text{最少1簇，所有样本同属一簇}\end{cases}$

(2)层次聚类的实现方式

样本数据的递归聚类；聚类过程中逐级考察簇间的距离，以此决定类别数.

合并式(聚合式)聚类(*agglomerative clustering*)

从聚类簇数目最多开始，**自底向上**，逐级**合并**距离最近的两个聚类簇，聚类簇的数目**逐渐减少**，直至与预设的聚类簇数目一致. 经常使用.

如：***AGNES*算法**(***AGglomerative NESting***)

分裂式聚类(*divisive clustering*)

从聚类簇数目最少开始，**自顶向下**，逐级**分裂**每级中最松散的聚类簇，聚类簇的数目逐级增加，直至与预设聚类簇数目一致.

如：***TSVQ***,可用于码本(***codebook***)的生成

(3)层次聚类的几个关键问题

相似性(相异性)度量

聚类簇之间；

簇内样本之间

聚类簇合并/分裂停止的条件

距离阈值；

预设的聚类簇数目

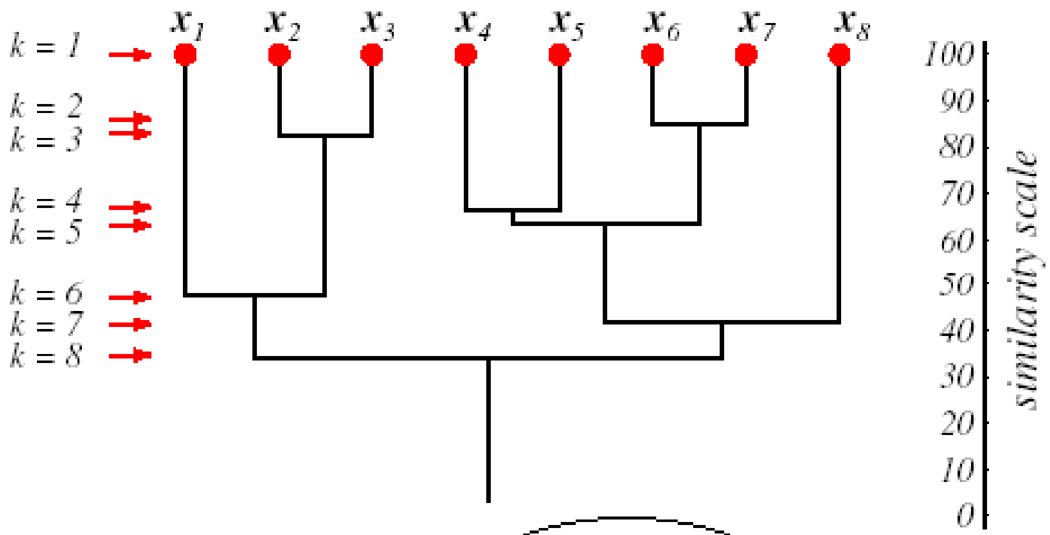
计算复杂度

(4)**聚合式**系统聚类结果的表示

A.树状图(*dendrogram*),聚类树

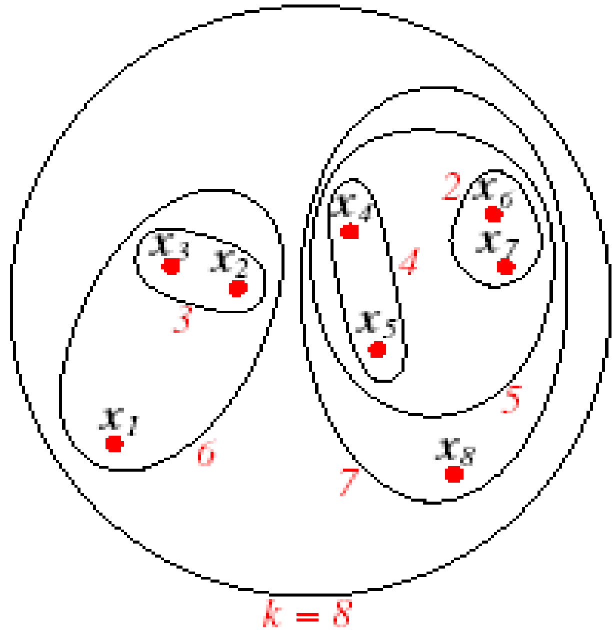
可定量表示聚类簇间的相似性(或相异性),

簇间相似性标尺随着层数增加，逐渐减小。



B.维恩图(*venn diagram*),集合图

只能**定性表示类间相似性**



2. 聚合式系统聚类算法

给定样本集$D=\{x_1,\cdots,x_m\}$，选定簇间距离度量$d(C_i,C_j)$，将样本D聚成k簇.

不同的距离度量，可产生不同的聚类结果

STEP1 初始化.

A. 指定最终类别数k;

B. 初始划分 $\begin{cases}\text{每个样本自成1簇} & C_i \Leftarrow \{x_i\} \quad i=1,\ldots,m \\ \text{初始类别数} & q \Leftarrow m\end{cases}$

STEP2 合并. 重复以下工作，直到$q=k$.

A. 寻找最相似(距离最小)的两簇$C_i,C_j\ (i<j)$

$$d(C_i,C_j)=\min_{\substack{\forall l,m\in\{1,\ldots,q\}\\ \text{并且} l<m}} \mathrm{d}(C_l,C_m)$$

B. 记录最小距离，合并两类C_i,C_j: $q \Leftarrow q-1$

STEP3 输出k个聚类簇.

AGNES算法描述

输入: 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
聚类簇距离度量函数 d;
聚类簇数 k.

过程:

for $j = 1, 2, \ldots, m$ **do**
  $C_j = \{\boldsymbol{x}_j\}$
end for
for $i = 1, 2, \ldots, m$ **do**
  for $j = 1, 2, \ldots, m$ **do**
    $M(i,j) = d(C_i, C_j)$;
    $M(j,i) = M(i,j)$
  end for
end for
设置当前聚类簇个数: $q = m$
while $q > k$ **do**
  找出距离最近的两个聚类簇 C_{i^*} 和 C_{j^*};
  合并 C_{i^*} 和 C_{j^*}: $C_{i^*} = C_{i^*} \bigcup C_{j^*}$;
  for $j = j^* + 1, j^* + 2, \ldots, q$ **do**
    将聚类簇 C_j 重编号为 C_{j-1}
  end for
  删除距离矩阵 M 的第 j^* 行与第 j^* 列;
  for $j = 1, 2, \ldots, q-1$ **do**
    $M(i^*, j) = d(C_{i^*}, C_j)$;
    $M(j, i^*) = M(i^*, j)$
  end for
  $q = q - 1$
end while

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

3.聚类簇C_i, C_j之间的连接$(linkage)$

(1)最小距离

$$d_{\min}\left(C_i, C_j\right)=\min_{\substack{x\in C_i\\ z\in C_j}} dist\left(x, z\right)$$

相应聚类算法就是"单链接"算法$(single\ linkage$ 或 $single-link)$

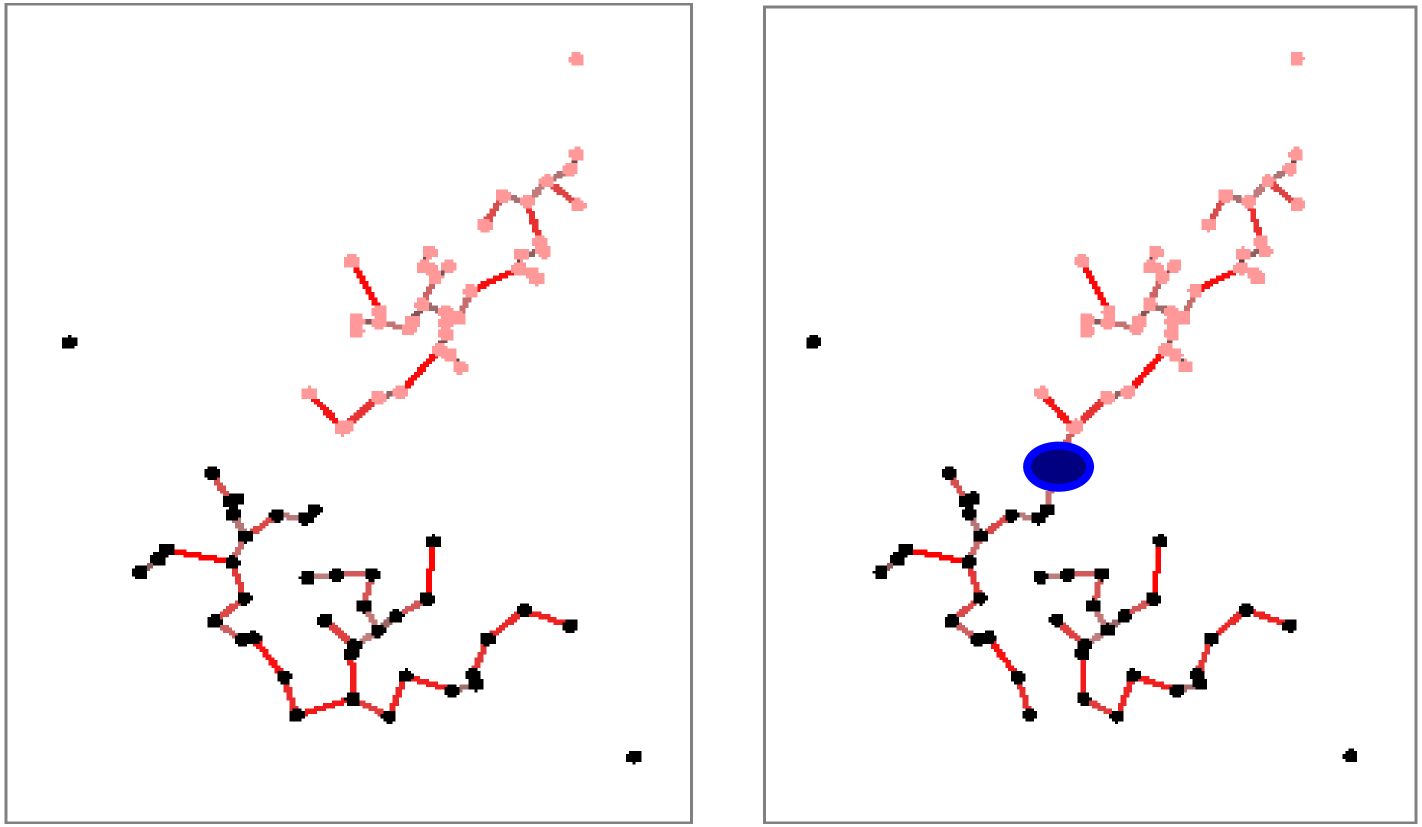
采用该距离度量**聚类簇C_i, C_j之间**的相异程度，进行聚类，就是产生**最小生成树**$(minimal\ spanning\ tree)$

缺陷　　链接效应，产生细长的聚类；
聚类结果对噪声或数据点波动敏感。

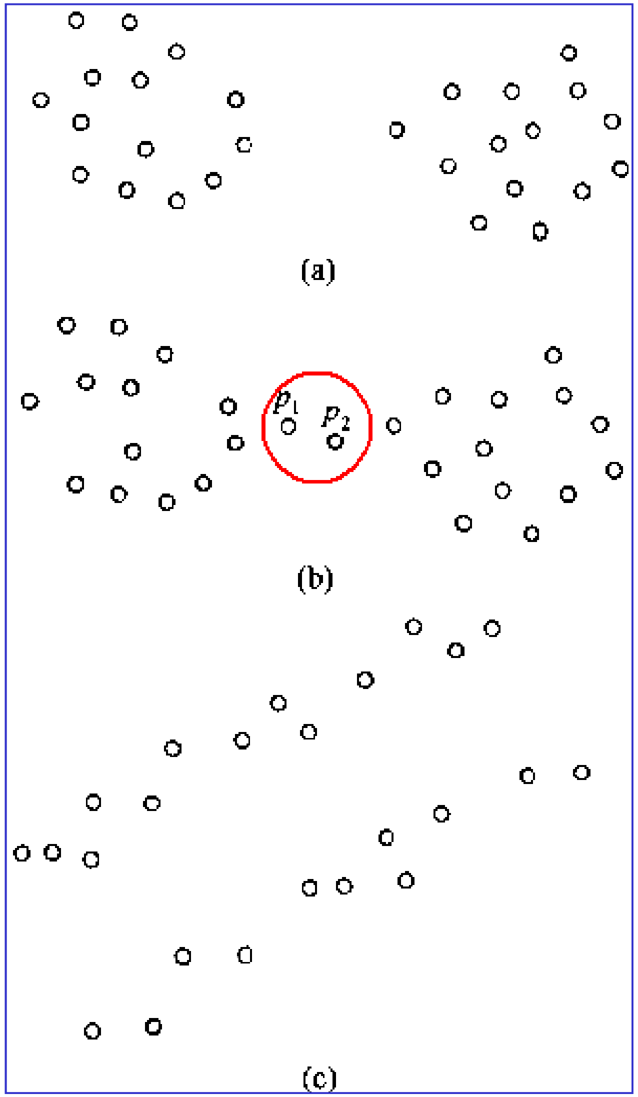
图. 二维高斯样本**最小距离法**层次聚类

左图 无干扰点

右图 有干扰点



最小距离法层次聚类 $(single\ linkage 或\ single-link)$



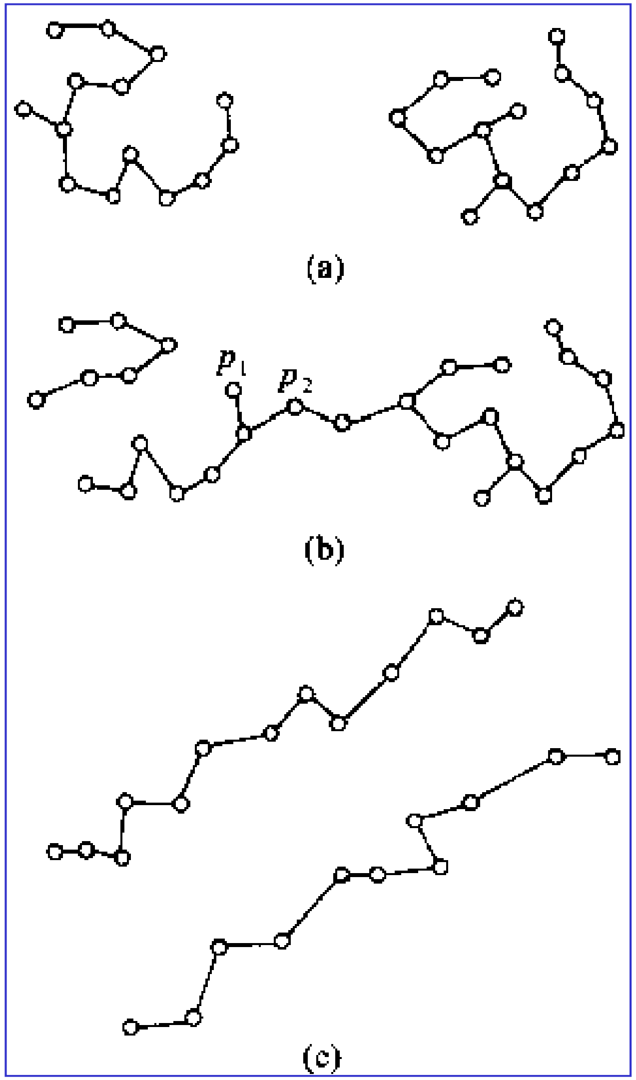
左图：

三种数据分布；

右图：

最近距离法的聚类结果

注意：a,b的区别



(2)最大距离

$$d_{\max}\left(C_i, C_j\right) = \max_{x \in C_i, z \in C_j} dist(x, z)$$

最远邻算法

$(the\ \ farthest\text{-}neighbor\ \ clustering\ \ algorithm)$

全连接算法

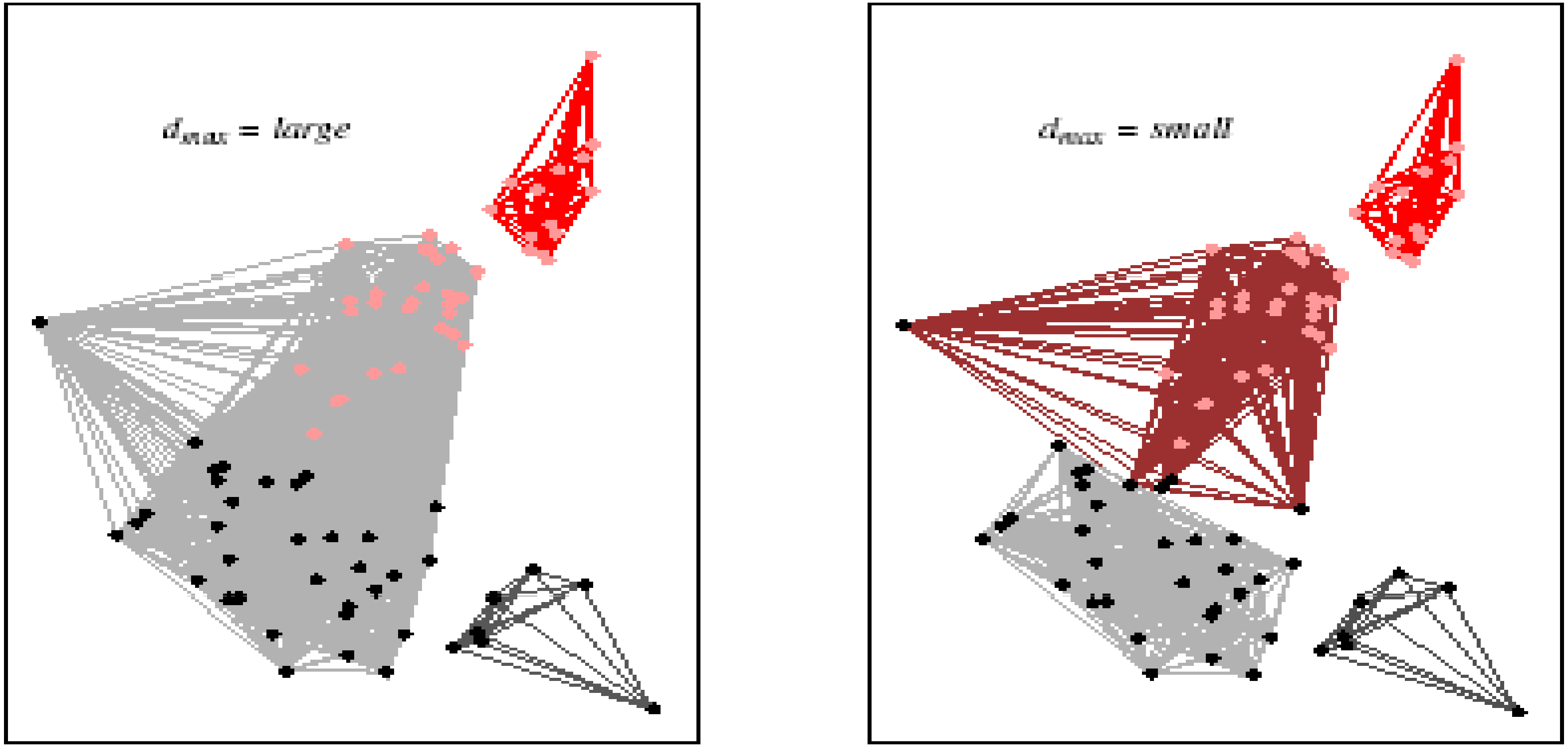
$(complete\text{-}linkage\ algorithm)$

若$d_{\max}\left(C_i, C_j\right) \ge d_0$,则聚类过程结束。

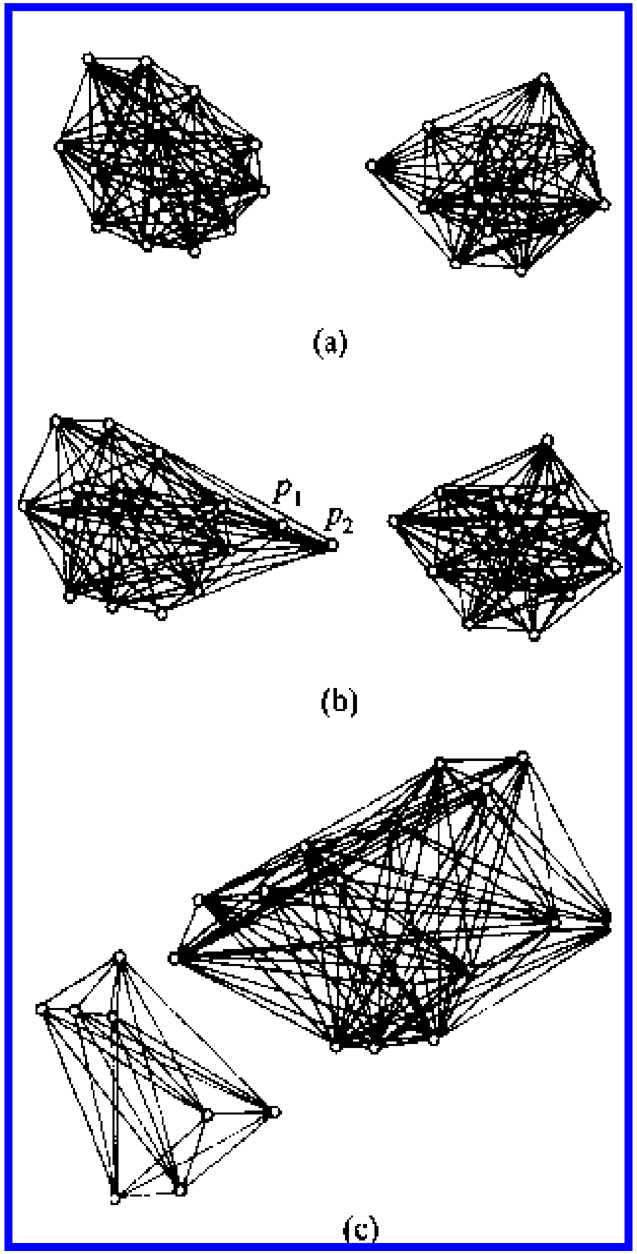
特点

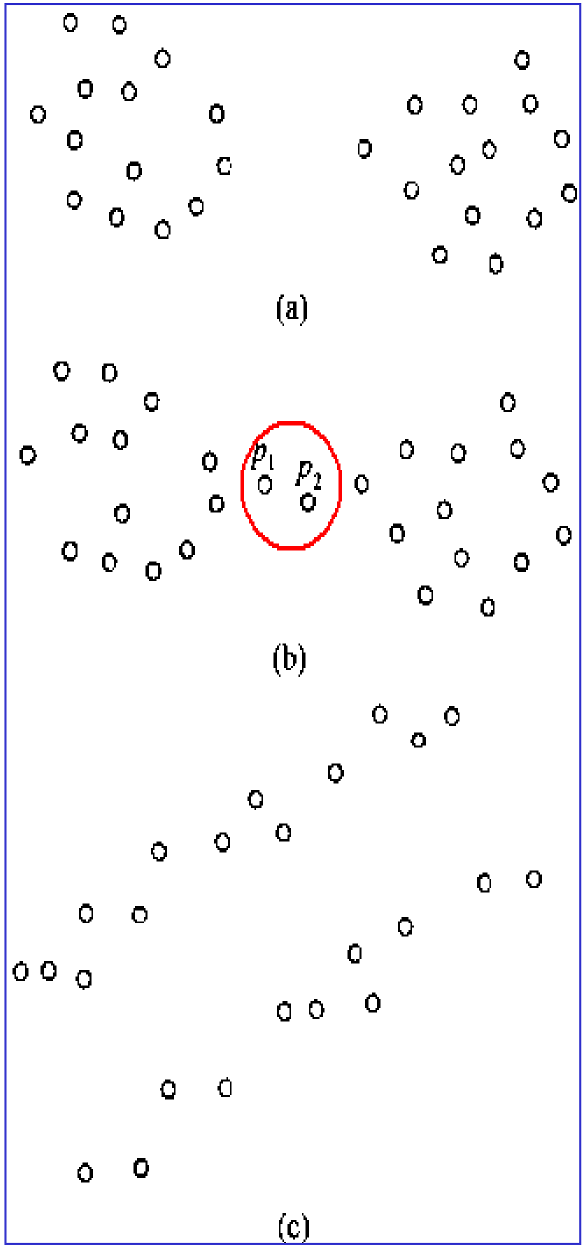
- 防止两个密集点集通过某个路径聚为一簇的可能
- 不能检测出具有长条形状的聚类
- 适合紧密、体积相近的类别划分
- 聚类结果对个别远离点敏感

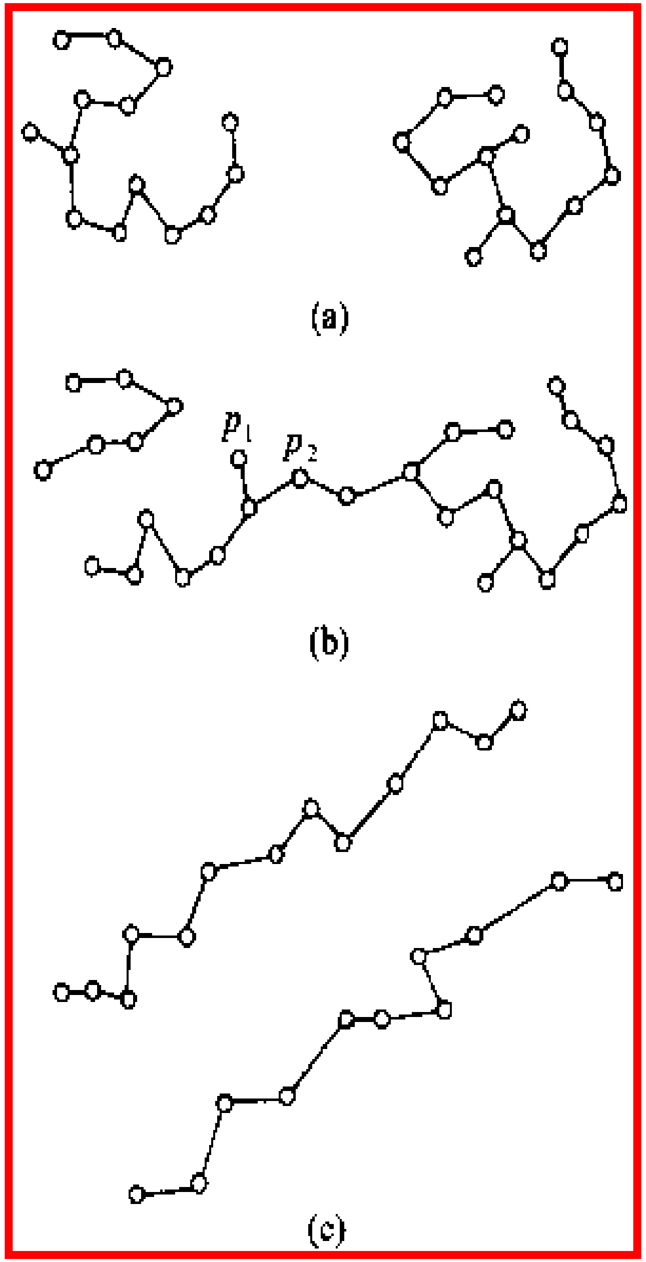
最大距离阈值$\boldsymbol{d}_0$越大，聚类簇的数目越小。



最大距离法与最小距离法聚类结果比较







(3)平均距离

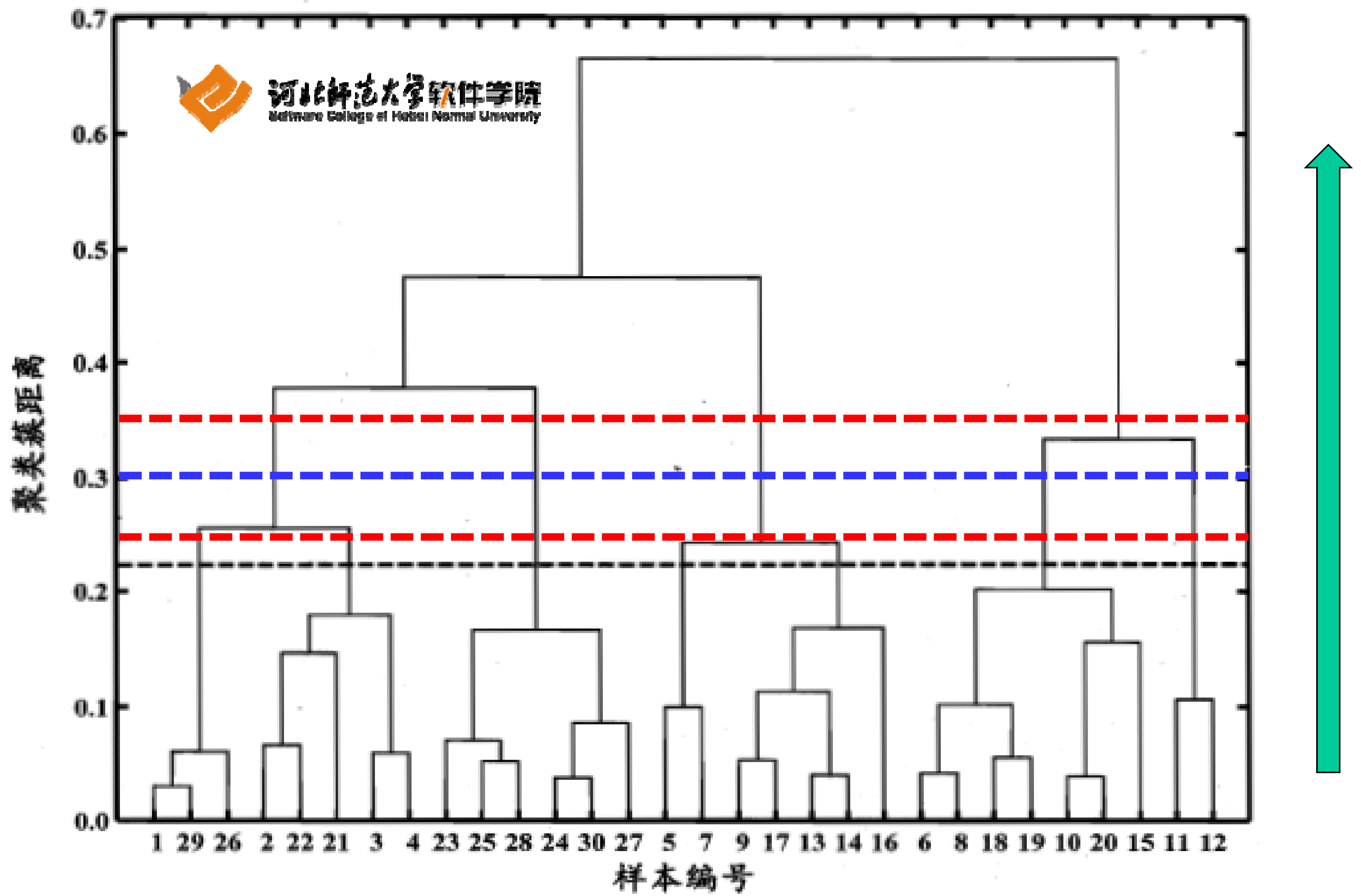
$$d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{z \in C_j} dist(x, z)$$

相应的聚类算法称为"均链接"算法 $(average-link, average\ \ linkage)$.

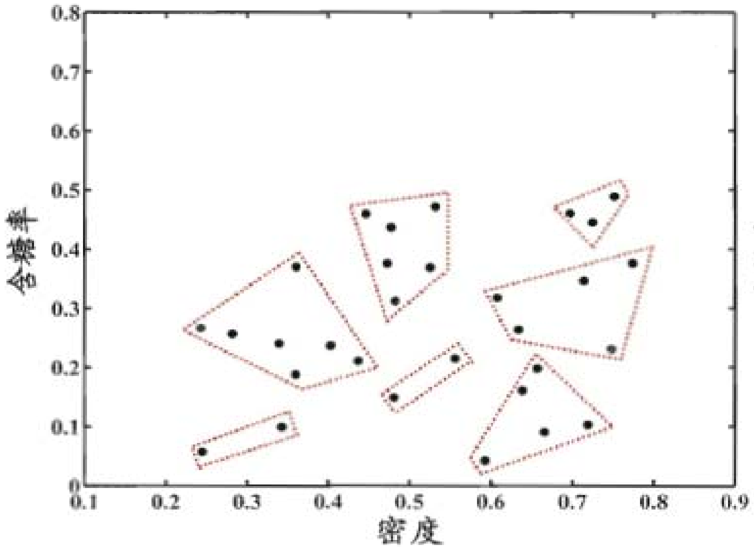
是关于前两种聚类算法的折中；计算简单。

西瓜数据集4.0

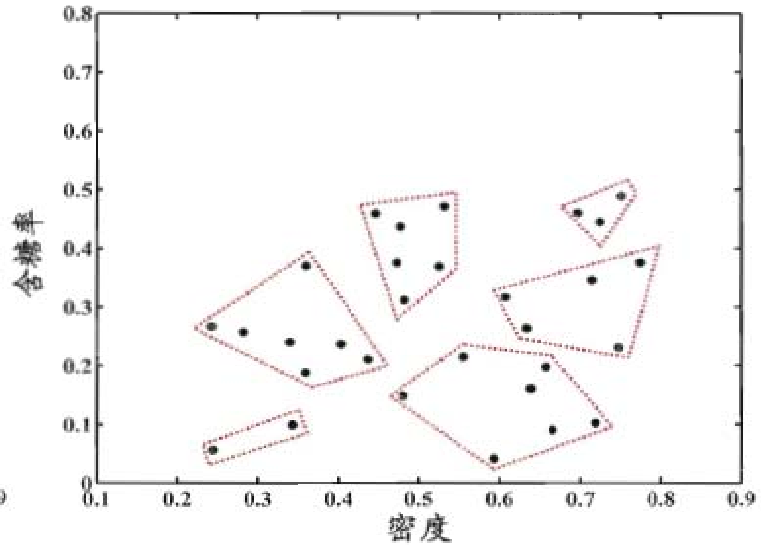
编号	密度	含糖率	编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	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



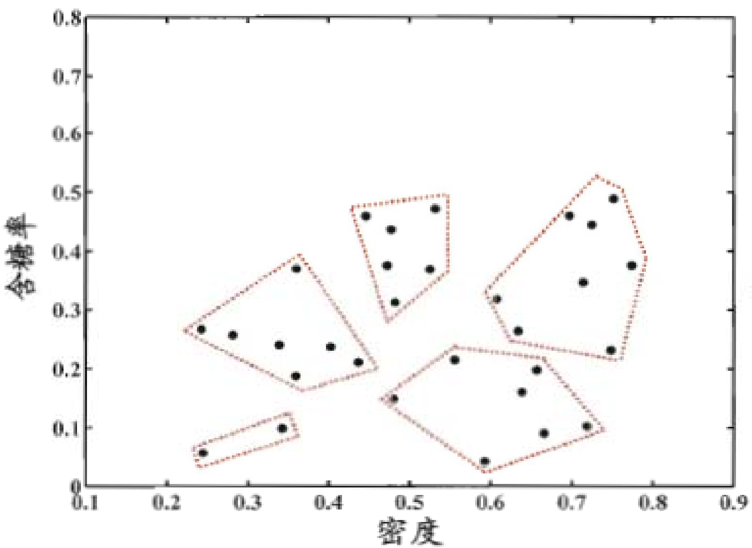
基于西瓜数据集4.0，采用聚合式层次聚类，得到的聚类树状图，簇之间的相异性度量采用最大距离法. 逐步提升分割层，聚类数目逐渐减小。



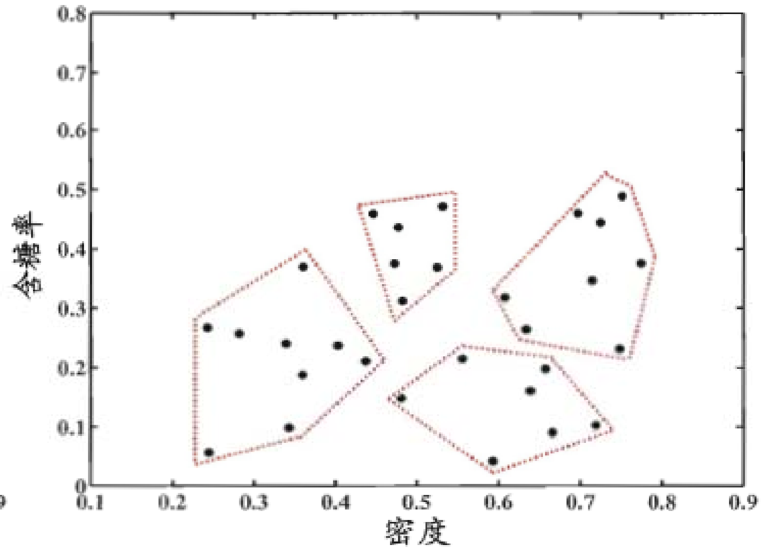
(a) 聚类簇数 $k = 7$



(b) 聚类簇数 $k = 6$



(c) 聚类簇数 $k = 5$



(d) 聚类簇数 $k = 4$

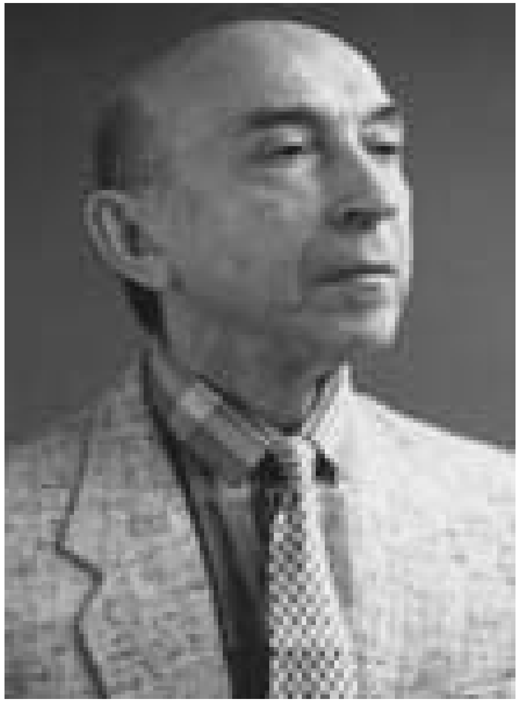
主要内容

1. 聚类的引入
2. 动态聚类
3. 基于模型的聚类 高斯混合聚类
4. 密度聚类(density-based clustering) DBSCAN
5. 层次聚类 hierarchical clustering
6. 其它聚类算法

6.1 Fuzzy K-Means Clustering

6.2 Mean Shift

河北师范大学软件学院
Software College of Hebei Normal University



Professor Lotfi A. Zadeh

1965年, Zadeh提出模糊集理论;
之后，出现了“模糊数学和模糊技术”学科
20世纪90年代初提出 soft computing
模糊逻辑是其三大核心之一。

http://www.cs.berkeley.edu/~zadeh/suprco.html

INFORMATION AND CONTROL 8, 338–353 (1965)

Fuzzy Sets*

L. A. ZADEH

Department of Electrical Engineering and Electronics Research Laboratory, University of California, Berkeley, California

1. 模糊集合的引入

模糊集理论是关于传统集合理论的一种推广；

传统集合可视为模糊集合的一种特别集合。

- 确定集合（清晰集，脆集合，Crisp Set）

  元素与集合关系：属于、或者不属于

- 模糊集合 (Fuzzy Set)

  元素以一定程度属于某集合，

  或以不同程度属于几个集合。

  ----适于表达自然语言变量和常识性知识

(1)隶属度函数$(membership\ \ function)$

$u_A(x)$--对象x隶属于集合A的程度

自变量x

所有可能属于集合A的对象构成**自变量范围**

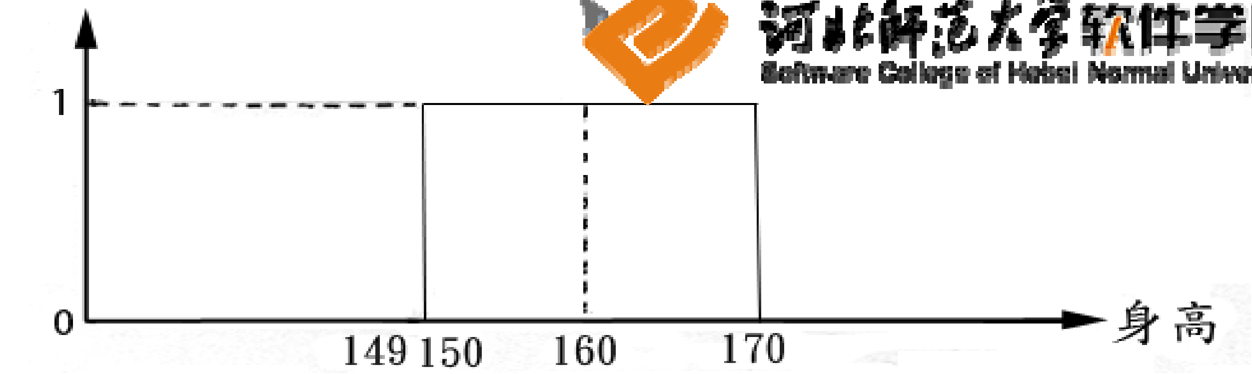
空间$X=\{x\}$ --论域

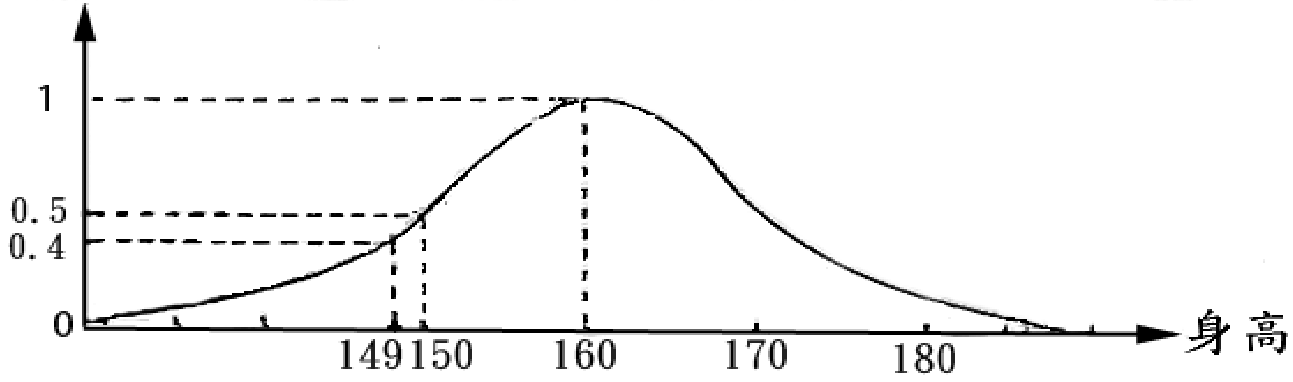
值域$[0,1]$ $0\le u_A(x)\le 1$

$$u_A(x)=0 \iff x\notin A$$

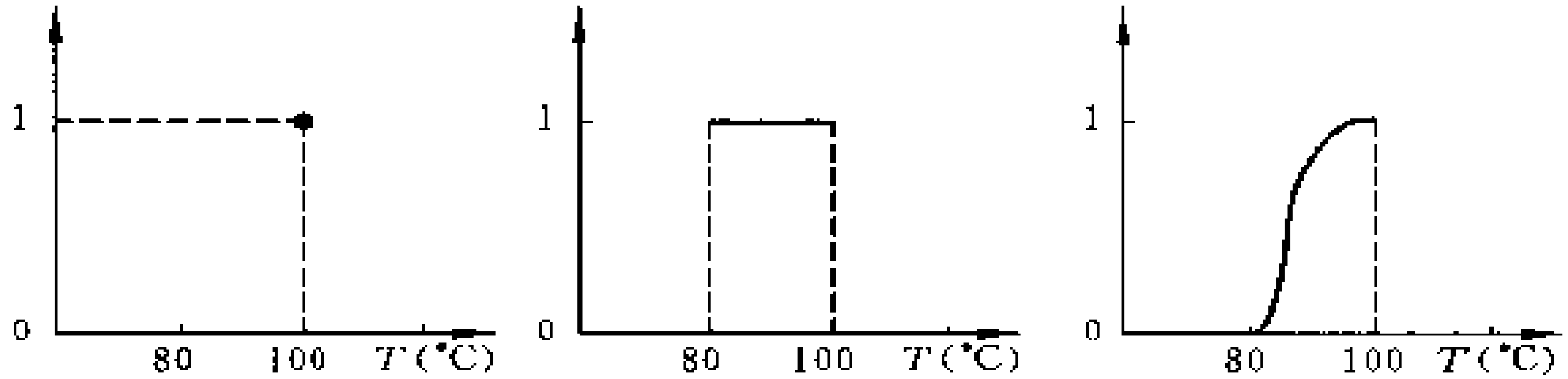
$$u_A(x)=1 \iff x\in A$$

例1“身高适中”的概念表达

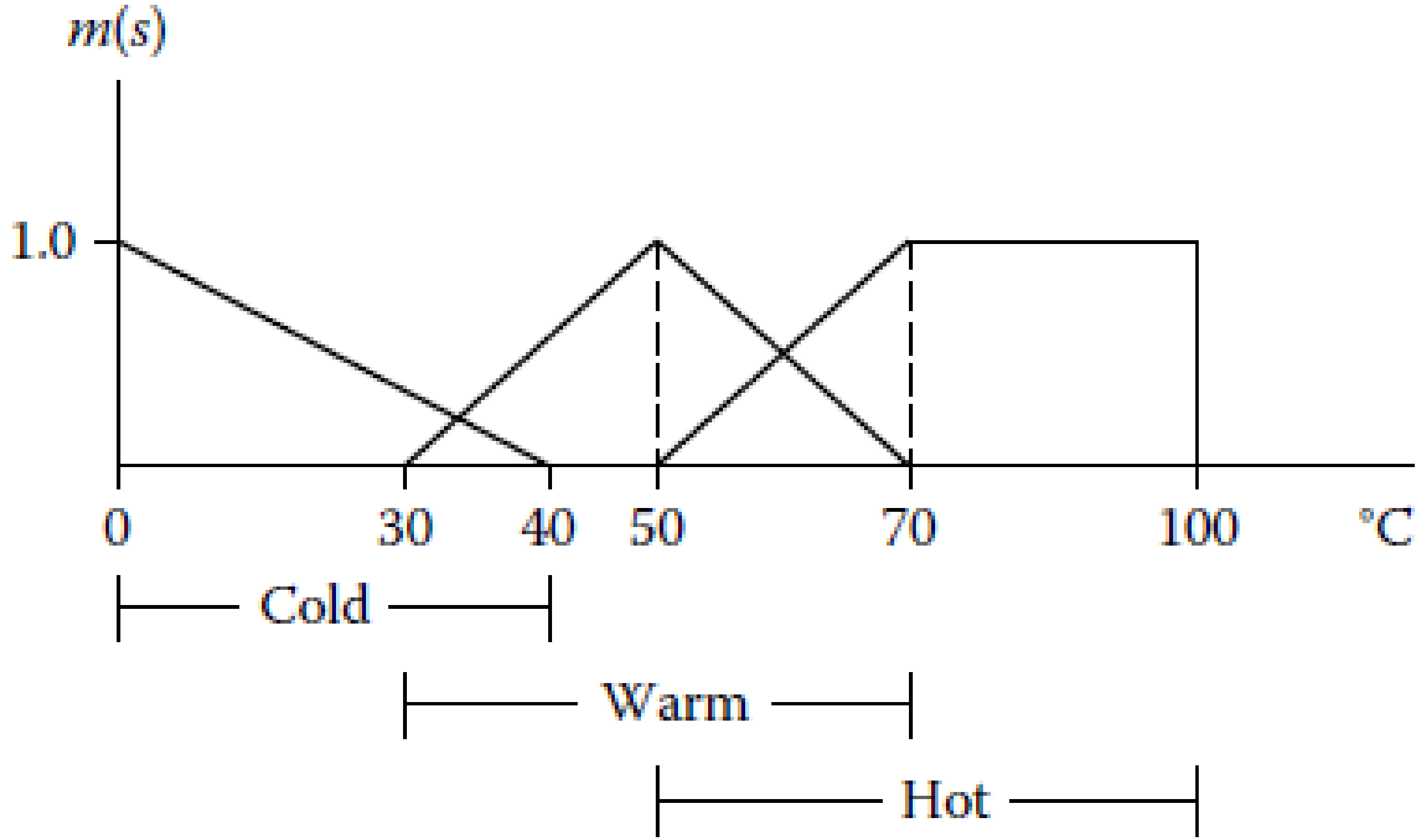




例2“开水”的概念表达



(a) (b) (c)



例3： Monotonic, triangular, and trapezoidal membership functions are used to perform fuzzy partitioning in dealing with the problem of distinguishing hot, warm, and cold water.

(2).模糊集的运算

$$A=\{(u_A(x),x)\mid x\in X\},B=\{(u_B(x),x)\mid x\in X\}$$

并 (*union*)

$$C=A\cup B=\{(u_{A\cup B}(x),x)\mid x\in X\}$$

$$u_{A\cup B}(x)=u_A(x)\vee u_B(x)=\max\{u_A(x),u_B(x)\},\quad \forall x\in X$$

交 (*intersection*)

$$C=A\cap B=\{(u_{A\cap B}(x),x)\mid x\in X\}$$

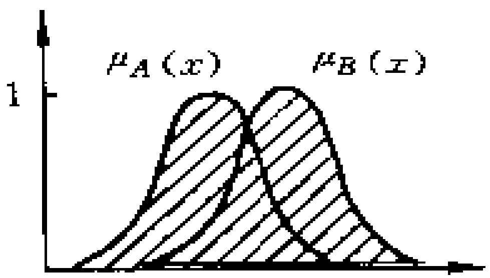
$$u_{A\cap B}(x)=u_A(x)\wedge u_B(x)=\min\{u_A(x),u_B(x)\},\forall x\in X$$

或 $u_c(x)=u_A(x)\cdot u_B(x)$

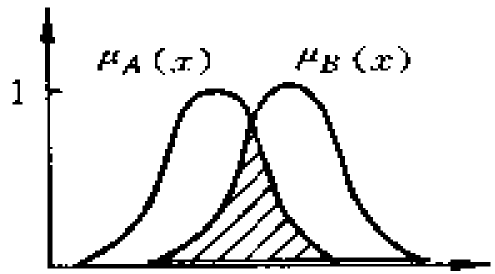
补 (complement)　模糊集A的补集A^c

$$u_{A^c}(x) = 1 - u_A(x), \quad \forall x \in X$$

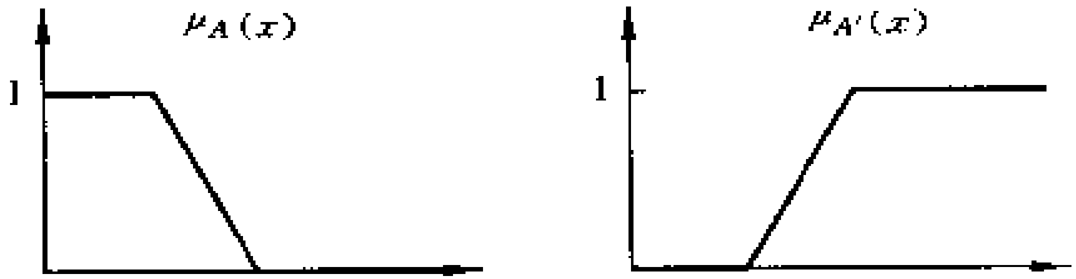
注意：对于模糊集A，$A \cap A^c \neq \Phi$



(a) 并



(b) 交



(c) 补

以下操作用于**增强**/**弱化**集合内元素隶属度值的差异。

$$A = \{(u_A(x), x) | x \in X\}$$

稀释(dilution)

--增加各元素隶属度值，并且隶属度值越小的元素相应隶属度值的增加幅度越大。

例： $dilution(u_A(x)) = \sqrt{u_A(x)}$

浓缩(concentration)

--降低各元素隶属度值，并且那些隶属度值越小的元素相应隶属度值的减小幅度越大。

例： $concentration(u_A(x)) = (u_A(x))^2$

$$\Rightarrow f(u_A(x)) = (u_A(x))^m \qquad \begin{cases} m > 1 & \text{浓缩} \\ 0 < m < 1 & \text{稀释} \end{cases}$$

2.模糊K-均值聚类

河北师范大学软件学院
Software College of Hebei Normal University

(1)**约定**

$$\begin{cases} \text{样本集} & D=\{\boldsymbol{x}_i, \boldsymbol{i}=1,2,\ldots,\mathrm{m}\} \\ \text{预定类别数目} & \boldsymbol{k} \\ \text{聚类中心} & \boldsymbol{\mu}_j, \boldsymbol{j}=1,2,\ldots,\boldsymbol{k} \\ \text{第}\boldsymbol{i}\text{个样本}\boldsymbol{x}_i\text{关于第}\boldsymbol{j}\text{类的隶属度函数} & u_j(\boldsymbol{x}_i) \end{cases}$$

(2)**聚类准则函数**

浓缩

$$J(\boldsymbol{u},\boldsymbol{\mu})=\sum_{j=1}^{k}\sum_{i=1}^{m}\left[u_j(\boldsymbol{x}_i)\right]^b\left\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\right\|^2$$

b--常数，控制聚类结果的模糊程度

$$\begin{cases} b>1, & \text{各样本以不同程度同时属于多类} \\ & \text{通常}\boldsymbol{b}\approx 2\text{，如}\boldsymbol{b}\in[1.5,2.5]\text{或}\boldsymbol{b}\in[1.3,1.8] \\ b\to 1, & \text{等价于均值法} \\ b\to\infty, & \text{完全模糊，各类中心均收敛至样本中心} \end{cases}$$

$$\begin{cases} \min\limits_{\mu,u} \sum_{j=1}^{k} \sum_{i=1}^{m} \left[u_j(x_i) \right]^b \left\| x_i - \mu_j \right\|^2 \\ s.t.\ \ \sum_{j=1}^{k} u_j(x_i) = 1, \quad i = 1, 2, \ldots, m \end{cases}$$

(3) ***Lagrange*** 条件极值法求解： $u_j(x_i)$， μ_j. $\begin{cases} i = 1, 2, \ldots, m \\ j = 1, 2, \ldots, k \end{cases}$

Lagrange 目标函数

$$J'(\mu, u, \lambda) = J_f + \sum_{i=1}^{m} \lambda_i \left[\sum_{j=1}^{k} u_j(x_i) - 1 \right]$$

$$= \sum_{j=1}^{k} \sum_{i=1}^{m} \left[u_j(x_i) \right]^b \left\| x_i - \mu_j \right\|^2 + \sum_{i=1}^{m} \lambda_i \left[\sum_{j=1}^{k} u_j(x_i) - 1 \right]$$

J' 最小时， J_f 最小

$$\begin{cases} \dfrac{\partial J'}{\partial \boldsymbol{\mu}_j} = 0 & j = 1, \ldots, k \\ \dfrac{\partial J'}{\partial u_j(\boldsymbol{x}_i)} = 0 & j = 1, \ldots, k \qquad i = 1, \ldots, m \end{cases}$$

$$\frac{\partial J'}{\partial u_j(\boldsymbol{x}_i)} = 0 \quad \Rightarrow u_j(\boldsymbol{x}_i) = \frac{\left(1 / \left\|\boldsymbol{x}_i - \mu_j\right\|^2\right)^{\frac{1}{b-1}}}{\sum_{j=1}^{k} \left(1 / \left\|\boldsymbol{x}_i - \mu_j\right\|^2\right)^{\frac{1}{b-1}}}$$

$$\frac{\partial J'}{\partial \boldsymbol{\mu}_j} = 0 \qquad \Rightarrow \quad \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{m} \left[u_j(\boldsymbol{x}_i)\right]^b \boldsymbol{x}_j}{\sum_{i=1}^{m} \left[u_j(\boldsymbol{x}_i)\right]^b}$$

上式关于**聚类中心**的估计为非解析的，**可迭代估计。**

(4)模糊k均值算法步骤

step1 设定参数：聚类数目k, 控制参数$b>1$, 终止条件ε

step2 初始化各聚类中心$\mu_i^{(0)}, i=1,2,\ldots,k$

step3 重复如下计算，直到隶属度值稳定$\left\| u^{(l+1)} - u^{(l)} \right\| < \varepsilon$

或聚类中心稳定$\left\| \mu^{(l+1)} - \mu^{(l)} \right\| < \varepsilon$

(1)由当前聚类中心$\mu_j^{(l)}$,计算隶属度函数

$$u_j^{(l+1)}(x_i) = \frac{\left(1 \Big/ \left\| x_i - \mu_j^{(l)} \right\|^2 \right)^{\frac{1}{b-1}}}{\sum_{j=1}^{k} \left(1 \Big/ \left\| x_i - \mu_j^{(l)} \right\|^2 \right)^{\frac{1}{b-1}}} \qquad \begin{array}{l} j=1,\ldots,k \\ i=1,\ldots,m \end{array}$$

(2)由当前隶属度函数$u_j^{(l+1)}(x_i)$, 更新各聚类中心

$$\mu_j^{(l+1)} = \frac{\sum_{i=1}^{m} \left[u_j^{(l+1)}(x_i) \right]^b x_i}{\sum_{i=1}^{m} \left[u_j^{(l+1)}(x_i) \right]^b} \qquad j=1,\ldots,k$$

(4)模糊k均值算法步骤

河北师范大学软件学院 Software College of Hebei Normal University

step4 迭代结束，输出：

$$\begin{cases} \text{聚类中心}\boldsymbol{\mu}_j & \boldsymbol{j}=1,2,\ldots,\boldsymbol{k} \\ \text{隶属度}u_j(\boldsymbol{x}_i) & \begin{matrix} \boldsymbol{j}=1,\ldots,\boldsymbol{k} \\ \boldsymbol{i}=1,\ldots,\boldsymbol{m} \end{matrix} \end{cases}$$

step5 去模糊化.

将模糊聚类转化为确定性分类。

$$\begin{cases} \textbf{for}\quad \boldsymbol{i}=1,\ldots,\boldsymbol{m} \\ \text{若}\boldsymbol{l}=\arg\max\limits_{\boldsymbol{j}\in\{1,\ldots,\boldsymbol{k}\}} u_j(\boldsymbol{x}_i),\text{则}\boldsymbol{x}_i\in\omega_l \end{cases}$$

思考题

1. 什么是聚类？什么是分类？
   请给出二者的区别与联系.

2. 若采用不同模型对给定的数据集D进行划分.请给出不同聚类算法的实现步骤。

(1)k-均值聚类
(2)高斯混合聚类
(3)层次聚类
(4)DBSCAN聚类

3. 上述四种聚类模型的适用场合.