第九章：聚类

主讲：连德富 特任教授 | 博士生导师
邮箱：liandefu@ustc.edu.cn
手机：13739227137
主页：http://staff.ustc.edu.cn/~liandefu

机器学习分类

- 按照标记区分

  - 分类：标记为离散值（二分类、多分类）
  - 回归：标记为连续值（瓜的成熟度）

  监督学习
  Supervised Learning

  - 聚类：没有标记

  无监督学习
  Unsupervised Learning

聚类：将数据集中的样本划分为若干个通常是不相交的子集

每个子集称为一个簇（cluster）

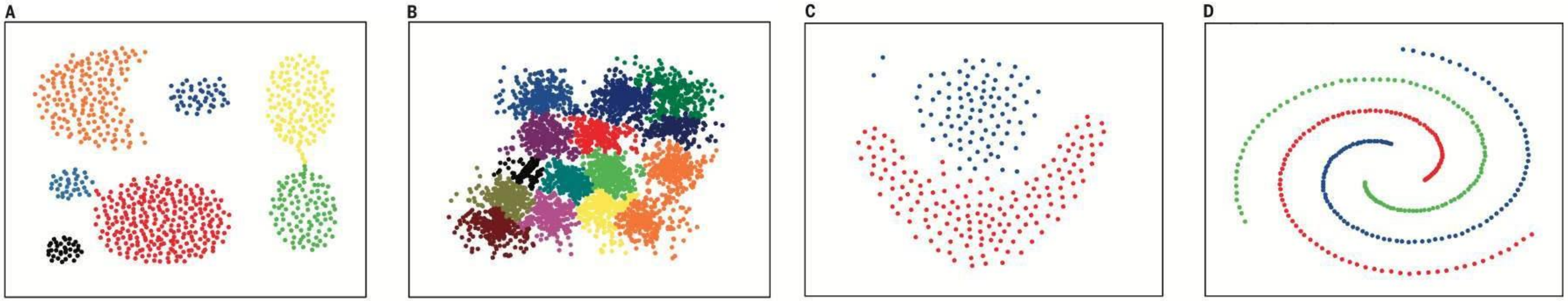
浅色瓜、深色瓜
有籽瓜、无籽瓜
本地瓜、外地瓜

聚类

- 聚类：将数据集中的样本划分为若干个通常是不相交的子集

- 形式化地说，假定样本集$D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$包含m个无标记样本，聚类算法将样本集$D$划分为k个不相交的簇$\{C_l | l = 1,2,\ldots,k\}$，其中$C_{l'} \cap C_l = \emptyset, \forall l' \neq l$ 且$D = \cup_{i=1}^{k} C_l$

- 用$\lambda_j \in \{1,2,\ldots,k\}$表示样本$\boldsymbol{x}_j$的簇标记（cluster label）,即$\boldsymbol{x}_j \in C_{\lambda_j}$



性能度量

- 聚类性能度量，亦称为聚类“有效性指标”（validity index）
- 直观来讲：

希望“物以类聚”，即同一簇的样本尽可能彼此相似，不同簇的样本尽可能不同。换言之，聚类结果的“簇内相似度”（intra-cluster similarity）高，且“簇间相似度”（inter-cluster similarity）低，这样的聚类效果较好

性能度量

- 聚类性能度量：

  - 外部指标 (external index)
    将聚类结果与某个“参考模型”(reference model)进行比较

  - 内部指标 (internal index)
    直接考察聚类结果而不用任何参考模型

性能度量—外部指标

- 对$D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$，通过聚类给出的簇划分为$\mathbb{C} = \{C_1, C_2, \ldots, C_k\}$
- 参考模型给出的簇划分为$\mathbb{C}^\star = \{C_1^\star, C_2^\star, \ldots, C_k^\star\}$
- 令λ与$\lambda^\star$分别表示与C和$C^\star$对应的簇标记向量

性能度量—外部指标

- 将样本两两配对

$$SS = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^\star = \lambda_j^\star, i < j\}, \quad a = |SS|$$
$$SD = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^\star \neq \lambda_j^\star, i < j\}, \quad b = |SS|$$
$$DS = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^\star = \lambda_j^\star, i < j\}, \quad c = |SS|$$
$$DD = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^\star \neq \lambda_j^\star, i < j\}, \quad d = |SS|$$

- Jaccard系数（Jaccard Coefficient, JC）

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

- FM指数（Fowlkes and Mallows Index, FMI）

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

- Rand指数（Rand Index, RI）

$$RI = \frac{2(a+b)}{m(m-1)}$$

上述性能度量的结果值均在[0,1]之间，值越大越好

性能度量—内部指标

- 考虑聚类结果的簇划分$\mathbb{C} = \{C_1, C_2, \ldots, C_k\}$

簇内样本平均距离

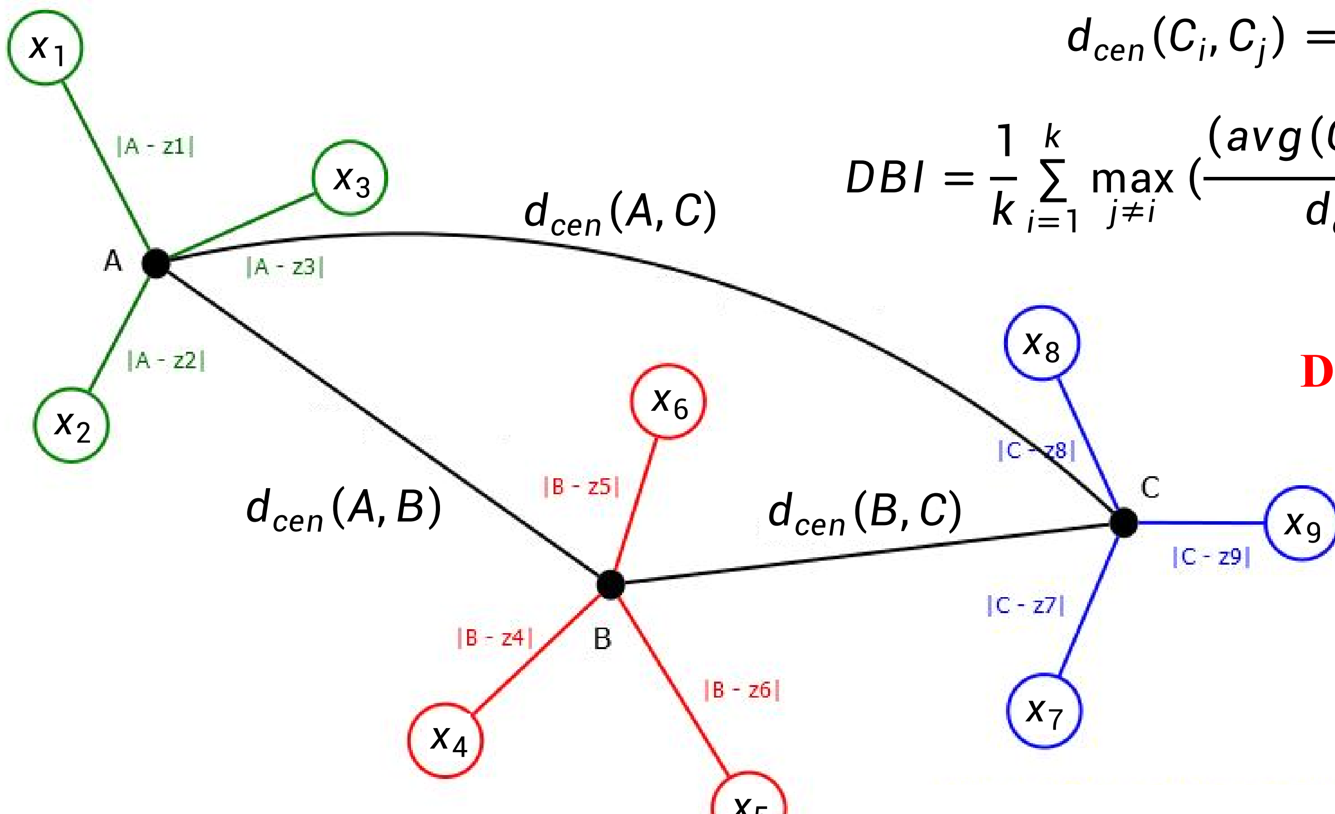
$$avg(C) = \frac{1}{|C|}\sum_{i=1}^{|C|} dist(\boldsymbol{\mu}, \boldsymbol{x}_i)$$

簇中心点的距离

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

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

$avg(A) = (|A - x_1| + |A - x_2| + |A - x_3|)/3$



DBI越小越好

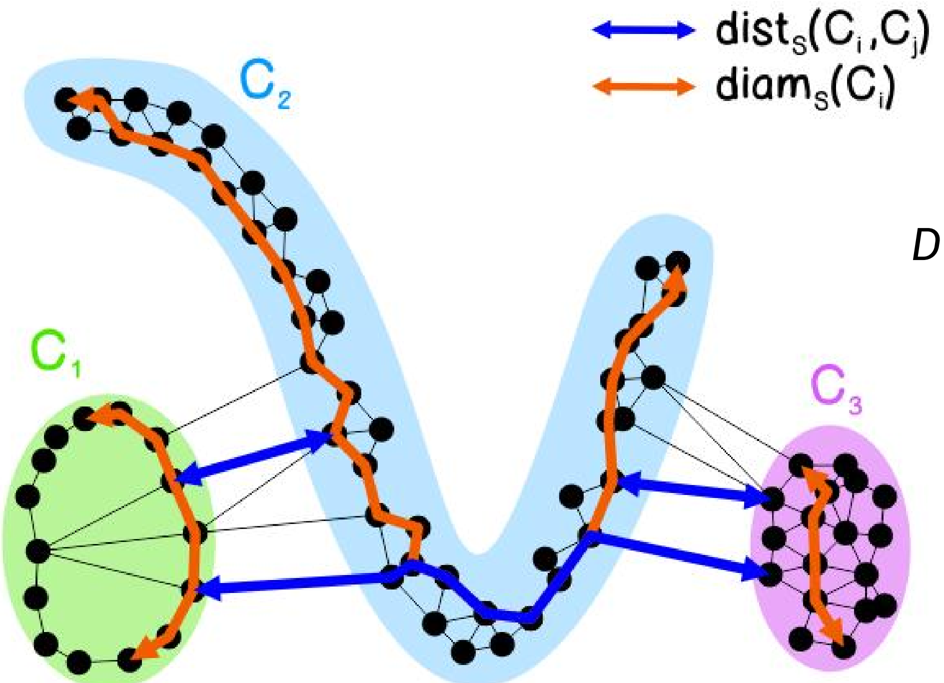
$avg(B) = (|B - x_4| + |B - x_5| + |B - x_6|)/3$

$avg(C) = (|C - x_7| + |C - x_8| + |C - x_9|)/3$

性能度量—内部指标

- 考虑聚类结果的簇划分$\mathbb{C}=\{C_1,C_2,\dots,C_k\}$

$$dist(C_i,C_j)=\min_{\boldsymbol{x}_i\in C_i,\boldsymbol{x}_j\in C_j} dist(\boldsymbol{x}_i,\boldsymbol{x}_j) \qquad \mathrm{diam}(C)=\max_{1\le i<j\le|C|} dist(\boldsymbol{x}_i,\boldsymbol{x}_j)$$



$$DI=\min_{1\le i\le k}\{\min_{j\ne i}(\frac{dist(C_i,C_j)}{\max_{1\le l\le k}(\mathrm{diam}(C_l))})\}$$

DI越大越好

距离计算

- 距离度量的性质：

非负性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$

同一性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$当且仅当$\boldsymbol{x}_i = \boldsymbol{x}_j$

对称性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = dist(\boldsymbol{x}_j, \boldsymbol{x}_i)$

传递性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq dist(\boldsymbol{x}_i, \boldsymbol{x}_k) + dist(\boldsymbol{x}_k, \boldsymbol{x}_j)$

- 常用距离：

闵可夫斯基距离（Minkowski distance）：

$$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{u}^{n} |x_{iu} - x_{ju}|^p\right)^{1/p}$$

主要应用连续属性上

$p = 2$：欧式距离

$p = 1$：曼哈顿距离，也称街区距离

距离计算

- 处理离散属性
  - 如果属性取值可比较，比如定义域{少年、中年、老年}。那么可以用{1,2,3}数值来表示，而1与2比较接近，与3比较远，可以直接计算距离。
  - 如果属性取值不可比，比如定义域为{飞机、火车、轮船}。那么无法计算距离
- 采样VDM (Value Difference Metric) 来处理无序属性
  - 令$m_{u,a}$表示属性u上取值为a的样本数，$m_{u,a,i}$表示在第i个样本簇中在属性u上取值为a的样本数。则属性u在两个离散值a和b的VDM距离为

$$VDM_p(a,b)=\sum_{i=1}^{k}\left|\frac{m_{u,a,i}}{m_{u,a}}-\frac{m_{u,b,i}}{m_{u,b}}\right|^p$$

距离计算

- 处理混合属性

$$\text{MinkovDM}_p = (\sum_{u=1}^{n_c} |x_{iu} - x_{ju}|^p + \sum_{u=n_c+1}^{n} VDM_p(x_{iu}, x_{ju}))$$

- 加权距离（样本中不同属性的重要性不同时）：
  - 加权闵可夫斯基距离

$$dist_{wmk}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (w_1|x_{i1} - y_{j1}|^p + \ldots + w_n|x_{in} - x_{jn}|^p)^{\frac{1}{p}}$$

满足$w_i \geq 0, \sum_i^n w_i = 1$

原型聚类

- 原型聚类

  也称为“基于原型的聚类”(prototype-based clustering)，此类算法假设聚类结构能通过一组原型刻画。

- 算法过程：

  通常情况下，算法先对原型进行初始化，再对原型进行迭代更新求解。

- 介绍几种著名的原型聚类算法

  k均值算法、学习向量量化算法、高斯混合聚类算法。

原型聚类—K均值

算法流程（迭代优化）：

初始化每个簇的均值向量

repeat

1. 将每个样本分配给最近的簇;
2. 计算每个簇的均值向量

until 当前均值向量均未更新

原型聚类—K均值

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

过程：

从D中随机选择k个样本作为初始均值向量$\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$

初始化每个簇的均值向量

repeat
　令$C_i = \emptyset\ (1 \leq i \leq k)$

将每个样本分配给最近的簇

　for $j = 1, \ldots, m$ **do**
　　计算样本$\boldsymbol{x}_j$与各均值向量$\boldsymbol{\mu}_i\ (1 \leq i \leq k)$的距离：$d_{ji} = ||\boldsymbol{x}_j - \boldsymbol{\mu}_i||_2$;
　　根据距离最近的均值向量确定$\boldsymbol{x}_j$的簇标记：$\lambda_j = \arg\min_{i \in \{1,2,\ldots,k\}} d_{ji}$;
　　将样本$\boldsymbol{x}_j$划入相应的簇：$C_{\lambda_j} = C_{\lambda_j} \bigcup \{\boldsymbol{x}_j\}$;
　end for
　for $i = 1, \ldots, k$ **do**
　　计算新均值向量：$\boldsymbol{\mu}'_i = \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x}$;

计算每个簇的均值向量

　　if $\boldsymbol{\mu}'_i \neq \boldsymbol{\mu}_i$ **then**
　　　将当前均值向量$\boldsymbol{\mu}_i$更新为$\boldsymbol{\mu}'_i$
　　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
return 簇划分结果

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

原型聚类—K均值

- 给定数据集 $D=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m\}$，K均值算法针对聚类所得簇划分$\mathbb{C}=\{C_1,C_2,\ldots,C_k\}$最小化平方误差

$$E=\sum_{c=1}^{k}\sum_{\boldsymbol{x}\in C_c}\|\boldsymbol{x}-\boldsymbol{\mu}_c\|_2^2 \qquad \boldsymbol{\mu}_c=\frac{1}{|C_c|}\sum_{\boldsymbol{x}\in C_c}\boldsymbol{x}$$

E值在一定程度上刻画了簇内样本围绕簇均值向量的紧密程度，E值越小，则簇内样本相似度越高。

$$E(\boldsymbol{T},\boldsymbol{\mu})=\sum_{i=1}^{m}\|\boldsymbol{x}_i-\boldsymbol{t}_i^\top\boldsymbol{\mu}\|_2^2=\|\boldsymbol{X}-\boldsymbol{T}\boldsymbol{\mu}\|_F^2 \qquad \boldsymbol{T}=\begin{bmatrix}\boldsymbol{t}_1^\top\\ \boldsymbol{t}_2^\top\\ \vdots\\ \boldsymbol{t}_m^\top\end{bmatrix}$$

$$\boldsymbol{t}_i^\top=[\underset{1}{0},\ldots,\underset{c_i}{1},\ldots,\underset{\mathrm{k}}{0}] \qquad \boldsymbol{\mu}=[\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_k]\in\mathbb{R}^{k\times d}$$

原型聚类—K均值

- 优化问题：$\min\limits_{\mathbf{T},\boldsymbol{\mu}} E(\boldsymbol{T},\boldsymbol{\mu}) = \|\boldsymbol{X} - \boldsymbol{T}\boldsymbol{\mu}\|_F^2$

算法流程（迭代优化）：

初始化每个簇的均值向量

repeat

1. 将每个样本分配给最近的簇； $\boldsymbol{T}^{(t)} \leftarrow \min\limits_{\boldsymbol{T}} E(\boldsymbol{T},\boldsymbol{\mu}^{(t-1)})$
2. 计算每个簇的均值向量； $\boldsymbol{\mu}^{(t)} \leftarrow \min\limits_{\boldsymbol{\mu}} E(\boldsymbol{T}^{(t)},\boldsymbol{\mu})$

until 当前均值向量均未更新

原型聚类—K均值

- 考虑$\boldsymbol{T}^{(t)} \leftarrow \min_{\boldsymbol{T}} E(\boldsymbol{T}, \boldsymbol{\mu}^{(t-1)})$

- 由于样本间互不依赖，考虑求解第i个样本的$\boldsymbol{t}_i$

$$\min_{\boldsymbol{t}_i} \|\boldsymbol{x}_i - \boldsymbol{t}_i^{\top} \boldsymbol{\mu}\|_2^2 = \|\boldsymbol{x}_i - \sum_c t_{ic} \boldsymbol{\mu}_c\|_2^2 \Leftrightarrow \min_{c} \|\boldsymbol{x}_i - \boldsymbol{\mu}_c\|$$

将样本i划分给距离最近的簇

t_{ic}在所有的c中只能有一个地方取值为1，其余均为0，即$\sum_c t_{ic} = 1$

原型聚类—K均值

- 进一步考虑$\boldsymbol{\mu}^{(t)} \leftarrow \min_{\boldsymbol{\mu}} E(\boldsymbol{T}^{(t)}, \boldsymbol{\mu})$
- 求$E(\boldsymbol{T}, \boldsymbol{\mu})$关于$\boldsymbol{\mu}$的梯度，令其等于0，则

$$\boldsymbol{\mu} = (\boldsymbol{T}^\top \boldsymbol{T})^{-1} \boldsymbol{T}^\top \boldsymbol{X}$$

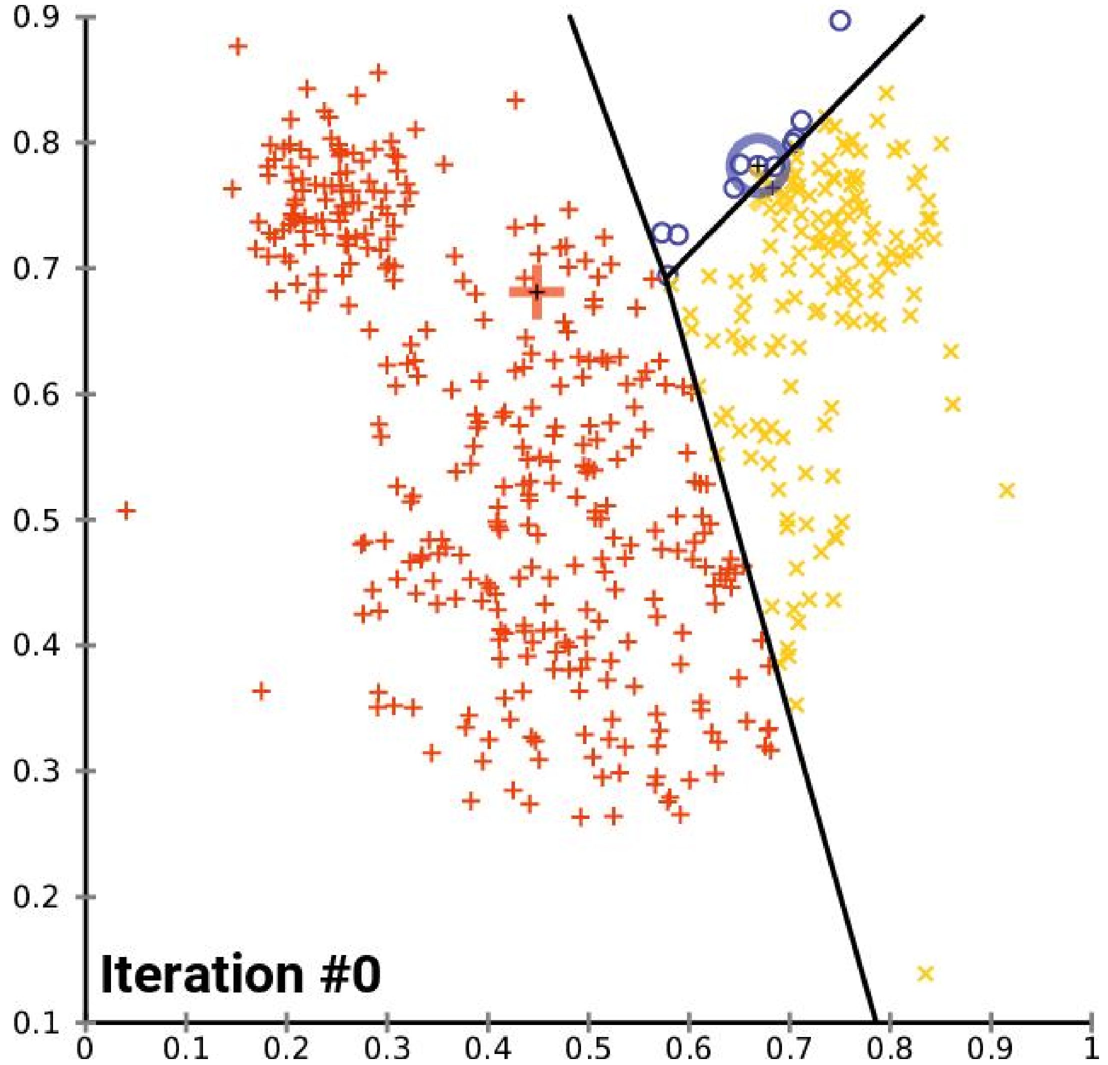
$$\boldsymbol{T}^\top \boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \dots, \boldsymbol{t}_m] \begin{bmatrix} \boldsymbol{t}_1^\top \\ \boldsymbol{t}_2^\top \\ \vdots \\ \boldsymbol{t}_m^\top \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{t}}_1^\top \\ \tilde{\boldsymbol{t}}_2^\top \\ \vdots \\ \tilde{\boldsymbol{t}}_k^\top \end{bmatrix} [\tilde{\boldsymbol{t}}_1, \tilde{\boldsymbol{t}}_2, \dots, \tilde{\boldsymbol{t}}_k] = \begin{bmatrix} \tilde{\boldsymbol{t}}_1^\top \tilde{\boldsymbol{t}}_1 & \tilde{\boldsymbol{t}}_1^\top \tilde{\boldsymbol{t}}_2 & \cdots & \tilde{\boldsymbol{t}}_1^\top \tilde{\boldsymbol{t}}_k \\ \tilde{\boldsymbol{t}}_2^\top \tilde{\boldsymbol{t}}_1 & \tilde{\boldsymbol{t}}_2^\top \tilde{\boldsymbol{t}}_2 & \cdots & \tilde{\boldsymbol{t}}_2^\top \tilde{\boldsymbol{t}}_k \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{t}}_k^\top \tilde{\boldsymbol{t}}_1 & \tilde{\boldsymbol{t}}_k^\top \tilde{\boldsymbol{t}}_2 & \cdots & \tilde{\boldsymbol{t}}_k^\top \tilde{\boldsymbol{t}}_k \end{bmatrix}$$

$$= \begin{bmatrix} |c_1| & 0 & \cdots & 0 \\ 0 & |c_2| & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & |c_k| \end{bmatrix}$$

$$\boldsymbol{T}^\top \boldsymbol{X} = \begin{bmatrix} \tilde{\boldsymbol{t}}_1^\top \boldsymbol{X} \\ \tilde{\boldsymbol{t}}_2^\top \boldsymbol{X} \\ \vdots \\ \tilde{\boldsymbol{t}}_k^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} \sum_{\boldsymbol{x} \in C_1} \boldsymbol{x} \\ \sum_{\boldsymbol{x} \in C_2} \boldsymbol{x} \\ \vdots \\ \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x} \end{bmatrix}$$

$$\boldsymbol{\mu} = (\boldsymbol{T}^\top \boldsymbol{T})^{-1} \boldsymbol{T}^\top \boldsymbol{X} = \begin{bmatrix} \frac{1}{|c_1|} \sum_{\boldsymbol{x} \in C_1} \boldsymbol{x} \\ \frac{1}{|c_1|} \sum_{\boldsymbol{x} \in C_2} \boldsymbol{x} \\ \vdots \\ \frac{1}{|c_1|} \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x} \end{bmatrix}$$

原型聚类—K均值



k均值算法实例

- 接下来以表9-1的西瓜数据集4.0为例，来演示k均值算法的学习过程。将编号为i的样本称为$\boldsymbol{x}_i$ 。

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均值算法实例

假定聚类簇数$k=3$，算法开始时，随机选择3个样本x_6, x_{12}, x_{27}作为初始均值向量，即$\mu_1 = (0.403,0.237), \mu_2 = (0.343,0.099), \mu_3 = (0.533,0.472)$

考察样本$\boldsymbol{x}_1 = (0.697,0.460)$，它与当前均值向量 μ_1，μ_2, μ_3的距离分别为$0.369, 0.506, 0.166$，因此 $\boldsymbol{x}_1$将被划入簇C_3中。类似的，对数据集中的所有样本考察一遍后，可得当前簇划分为

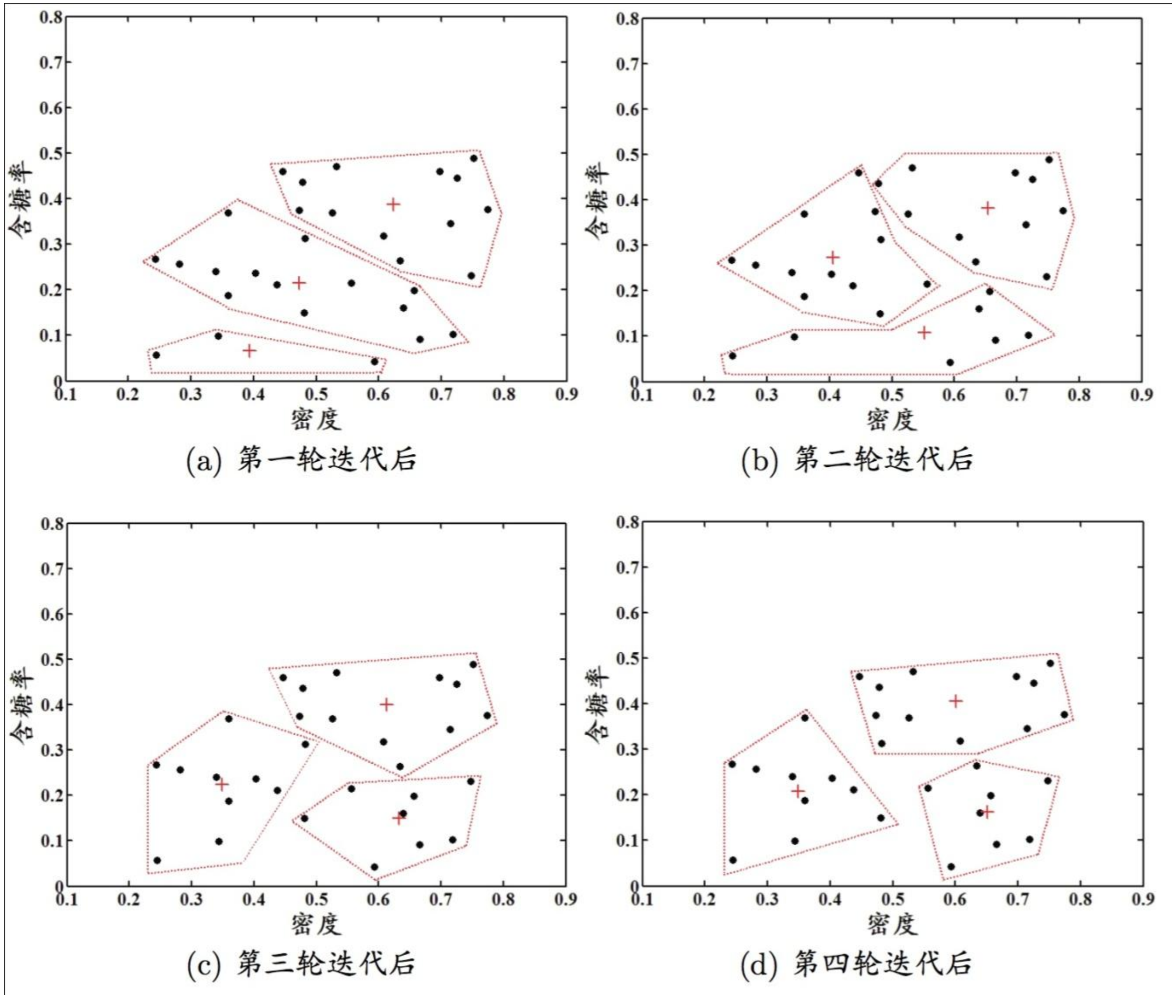
$$C_1 = \{x_5, x_6, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}, x_{20}, x_{23}\}$$
$$C_2 = \{x_{11}, x_{12}, x_{16}\}$$
$$C_3 = \{x_1, x_2, x_3, x_4, x_{21}, x_{22}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}\}$$

于是，可以从分别求得新的均值向量

$$\mu_1' = (0.473,0.214), \mu_2' = (0.394,0.066), \mu_3' = (0.623,0.388)$$

不断重复上述过程，如下图所示。

k均值算法实例



(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

原型聚类—学习向量量化

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

- 与一般聚类算法不同的是，LVQ假设数据样本带有类别标记，学习过程中利用样本的这些监督信息来辅助聚类

- 给定样本集$D=\{(\boldsymbol{x}_1,y_1),\cdots,(\boldsymbol{x}_m,y_m)\}$，$y_i\in\mathcal{Y}$，LVQ的目标是学得一组$d$维原型向量$\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_q$，每个原型向量代表一个聚类簇，簇标记$t_i\in\mathcal{Y}$

原型聚类—学习向量量化

输入： 样本集$D=\{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_m,y_m)\}$;
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; 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
return 当前原型向量
```

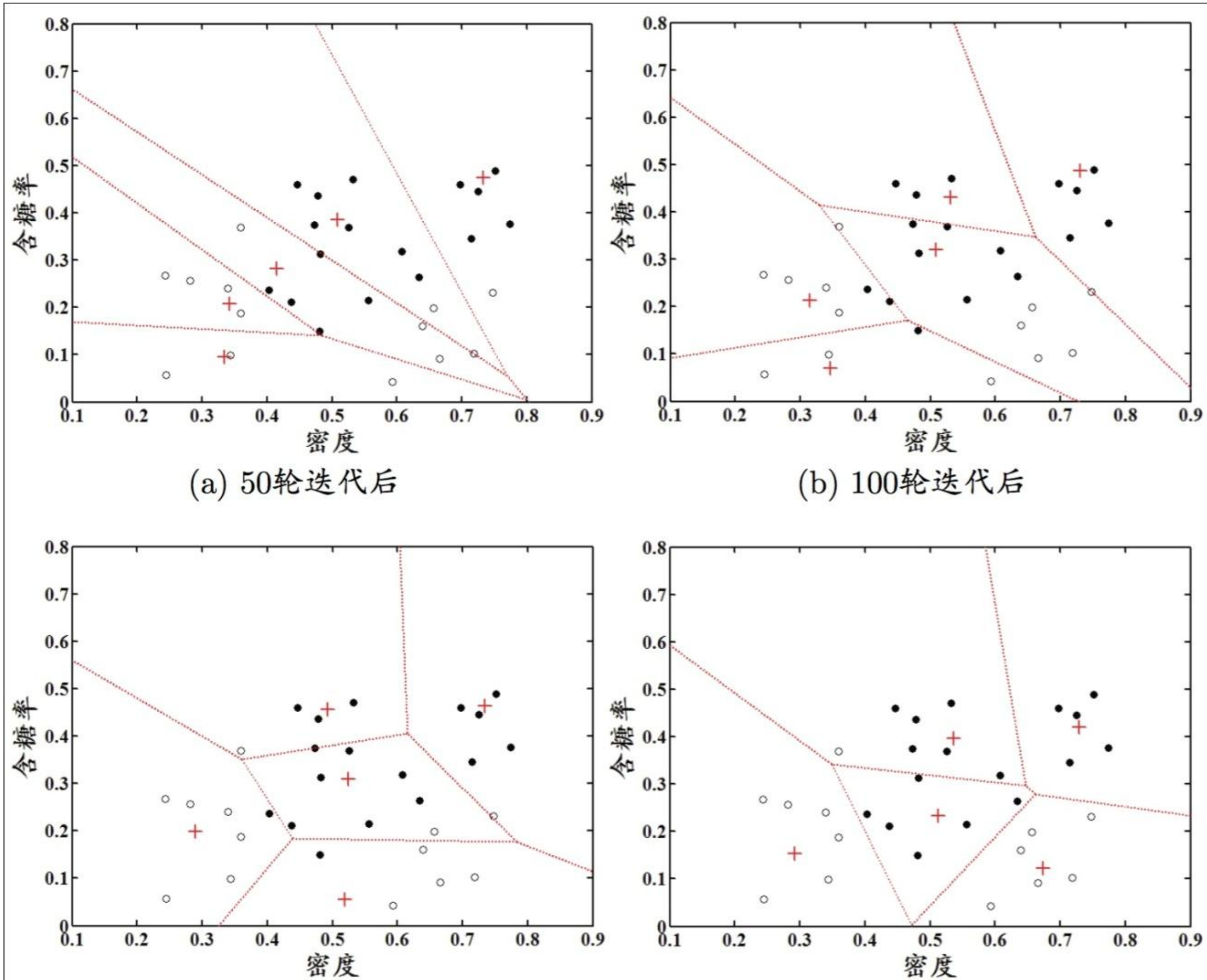
输出： 原型向量$\{\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_q\}$

根据两者的类别标记是否一致来对原型向量进行更新

对样本$\boldsymbol{x}_j$，若最近的原型向量$\boldsymbol{p}_{i^*}$和$\boldsymbol{x}_j$的类别标记相同，则令$\boldsymbol{p}_{i^*}$向$\boldsymbol{x}_j$的方向靠拢

$$\|\boldsymbol{p}' - \boldsymbol{x}_j\| = \|\boldsymbol{p}_{i^*} + \eta(\boldsymbol{x}_j - \boldsymbol{p}_{i^*}) - \boldsymbol{x}_j\|$$
$$= (1-\eta)\|\boldsymbol{p}_{i^*} - \boldsymbol{x}_j\|$$
$$< \|\boldsymbol{p}_{i^*} - \boldsymbol{x}_j\|$$

原型聚类—学习向量量化



(a) 50轮迭代后　(b) 100轮迭代后

(c) 200轮迭代后　(d) 400轮迭代后

原型聚类 – 高斯混合聚类

- 与k均值、LVQ用原型向量来刻画聚类结构不同，高斯混合聚类（Mixture-of-Gaussian）采用概率模型来表达聚类原型：

- 对d维样本空间中的随机向量$\boldsymbol{x}$，若$\boldsymbol{x}$服从多元高斯分布，其概率密度函数为

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 其中$\boldsymbol{\mu}$是d维均值向量，$\boldsymbol{\Sigma}$是$d \times d$的协方差矩阵。也可将概率密度函数记作$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$。

原型聚类 – 高斯混合聚类

- 高斯混合分布的定义

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} a_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

该分布由k个混合分布组成，每个分布对应一个高斯分布。其中，$\boldsymbol{\mu}_c$与$\boldsymbol{\Sigma}_c$是第c个高斯混合成分的参数。而a_c为相应的“混合系数”，$\sum_c a_c = 1$。

假设样本的生成过程由高斯混合分布给出：

(1) 根据$a_1, \cdots, a_k$定义的先验分布选择高斯混合成分，a_i为选择第i个成分的概率；

(2) 根据被选择的混合成分的概率密度函数进行采样，从而生成相应的样本

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \qquad p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 求$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$关于$\boldsymbol{\mu_c}$的偏导数

$$\frac{\partial LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu_c}} = \sum_{i=1}^{m} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu_c}) = 0$$

$$\Rightarrow \quad \sum_{i=1}^{m} \gamma_{ic} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu_c}) = 0$$

$$\Rightarrow \quad \boldsymbol{\mu_c} = \frac{\sum_{i=1}^{m} \gamma_{ic} \boldsymbol{x}_i}{\sum_{i=1}^{m} \gamma_{ic}}$$

各混合成分的均值可通过样本加权平均来估计

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} a_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{a}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} a_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 令$LL(\boldsymbol{a}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$关于$\boldsymbol{\Sigma}_{\boldsymbol{c}}$的偏导数等于0，可得

$$\boldsymbol{\Sigma}_c = \frac{\sum_i^m \gamma_{ic}(\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^\top}{\sum_i^m \gamma_{ic}}$$

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \qquad p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 考虑$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$的拉格朗日形式

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda\left(\sum_{c} \alpha_c - 1\right)$$

- 求其关于α_c的导数，令其等于0

$$\frac{\partial LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \alpha_c} = \sum_{i=1}^{m} \frac{p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} + \lambda = 0 \Rightarrow \sum_{i=1}^{m} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} = -\lambda \alpha_c$$

$$\Rightarrow \sum_{c=1}^{k} \sum_{i=1}^{m} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} = \sum_{c=1}^{k} -\lambda \alpha_c \Rightarrow \lambda = -m$$

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \qquad p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 考虑$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$的拉格朗日形式

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda\left(\sum_{c} \alpha_c - 1\right)$$

- 求其关于α_c的导数，令其等于0

$$\frac{\partial LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \alpha_c} = \sum_{i=1}^{m} \frac{p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} + \lambda = 0 \Rightarrow \sum_{i=1}^{m} \frac{1}{\alpha_c} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} = m$$

$$\Rightarrow \sum_{i=1}^{m} \frac{1}{\alpha_c} \gamma_{ic} = m \Rightarrow \alpha_c = \frac{1}{m} \sum_{i=1}^{m} \gamma_{ic}$$

原型聚类 – 高斯混合聚类

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.

过程：

初始化高斯混合分布的模型参数$\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$
repeat
  for $j = 1, \ldots, m$ **do**
    根据(9.30)计算$\boldsymbol{x}_j$由各混合成分生成的后验概率，即
    $\gamma_{ji} = p_{\mathcal{M}}(z_j = i \mid \boldsymbol{x}_j)\ (1 \leq i \leq k)$
  end for
  for $i = 1, \ldots, k$ **do**
    计算新均值向量：$\boldsymbol{\mu}_i' = \frac{\sum_{j=1}^{m} \gamma_{ji} \boldsymbol{x}_j}{\sum_{j=1}^{m} \gamma_{ji}}$;
    计算新协方差矩阵：$\boldsymbol{\Sigma}_i' = \frac{\sum_{j=1}^{m} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i')(\boldsymbol{x}_j - \boldsymbol{\mu}_i')^{\top}}{\sum_{j=1}^{m} \gamma_{ji}}$;
    计算新混合系数：$\alpha_i' = \frac{\sum_{j=1}^{m} \gamma_{ji}}{m}$;
  end for
  将模型参数$\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$更新为$\{(\alpha_i', \boldsymbol{\mu}_i', \boldsymbol{\Sigma}_i') \mid 1 \leq i \leq k\}$
until 满足停止条件
$C_i = \emptyset\ (1 \leq i \leq k)$
for $j = 1, \ldots, m$ **do**
  根据(9.31)确定$\boldsymbol{x}_j$的簇标记λ_j;
  将$\boldsymbol{x}_j$划入相应的簇：$C_{\lambda_j} = C_{\lambda_j} \bigcup \{\boldsymbol{x}_j\}$
end for
return 簇划分结果

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

密度聚类

- 密度聚类也称为“基于密度的聚类”(density-based clustering)。
- 此类算法假设聚类结构能通过样本分布的紧密程度来确定。
- 通常情况下，密度聚类算法从样本密度的角度来考察样本之间的可连接性，并基于可连接样本不断扩展聚类簇来获得最终的聚类结果

接下来介绍DBSCAN这一密度聚类算法。

密度聚类

- DBSCAN算法：基于一组“邻域”参数$(\epsilon, \mathrm{MinPts})$来刻画样本分布的紧密程度。
- 基本概念：

- ϵ邻域：对样本$\boldsymbol{x}_i \in D$，其ϵ邻域包含样本集D中与$\boldsymbol{x}_i$的距离不大于ϵ的样本；

- 核心对象：若样本$\boldsymbol{x}_i$的ϵ邻域至少包含MinPts个样本，则该样本点为核心对象；

- 密度直达：若样本$\boldsymbol{x}_j$位于样本$\boldsymbol{x}_i$的ϵ邻域中，且$\boldsymbol{x}_i$是一个核心对象，则称样本$\boldsymbol{x}_j$由$\boldsymbol{x}_i$密度直达；

- 密度可达：对样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$，若存在样本序列$\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n$，其中$\boldsymbol{p}_1 = \boldsymbol{x}_i, \boldsymbol{p}_n = \boldsymbol{x}_j$且$\boldsymbol{p}_{i+1}$由$\boldsymbol{p}_i$密度直达，则该两样本密度可达；

- 密度相连：对样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$，若存在样本$\boldsymbol{x}_k$使得两样本均由$\boldsymbol{x}_k$密度可达，则称该两样本密度相连。

密度聚类

- 一个例子

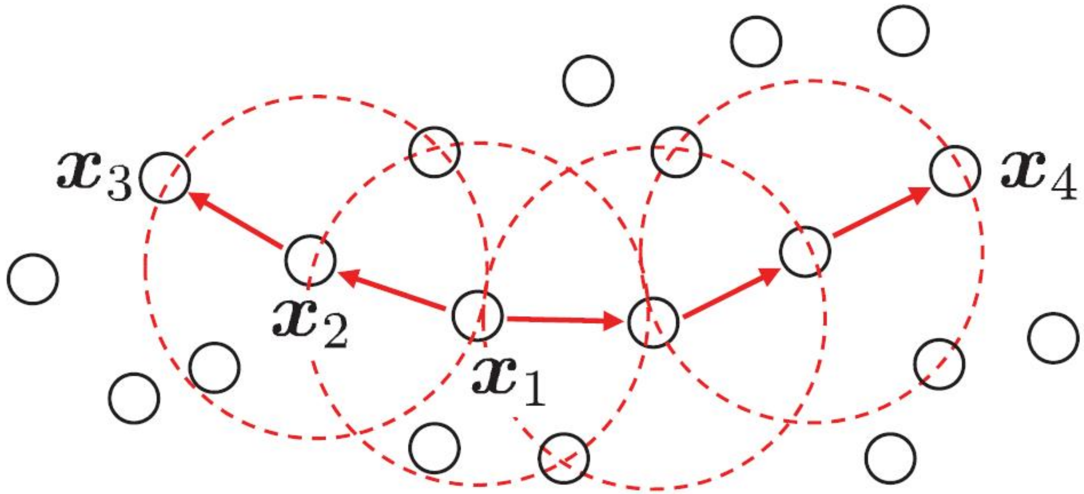
  令$MinPts = 3$，则虚线显示出ϵ领域。

  $\boldsymbol{x}_1$是核心对象。

  $\boldsymbol{x}_2$由$\boldsymbol{x}_1$密度直达。

  $\boldsymbol{x}_3$由$\boldsymbol{x}_1$密度可达。

  $\boldsymbol{x}_3$与$\boldsymbol{x}_4$密度相连。



密度聚类

- 对“簇”的定义

  由密度可达关系导出的最大密度相连样本集合。

- 对“簇”的形式化描述

  给定领域参数，簇是满足以下性质的非空样本子集：

  连接性：$\boldsymbol{x}_i \in C, \boldsymbol{x}_j \in C \Rightarrow \boldsymbol{x}_i$与$\boldsymbol{x}_j$密度相连

  最大性：$\boldsymbol{x}_i \in C$，$\boldsymbol{x}_i$与$\boldsymbol{x}_j$密度可达$\Rightarrow \boldsymbol{x}_j \in C$

实际上，若$\boldsymbol{x}$为核心对象，由$\boldsymbol{x}$密度可达的所有样本组成的集合记为$X = \{\boldsymbol{x}' \in D \mid \boldsymbol{x}'$由$\boldsymbol{x}$密度可达$\}$，则$X$为满足连接性与最大性的簇。

密度聚类

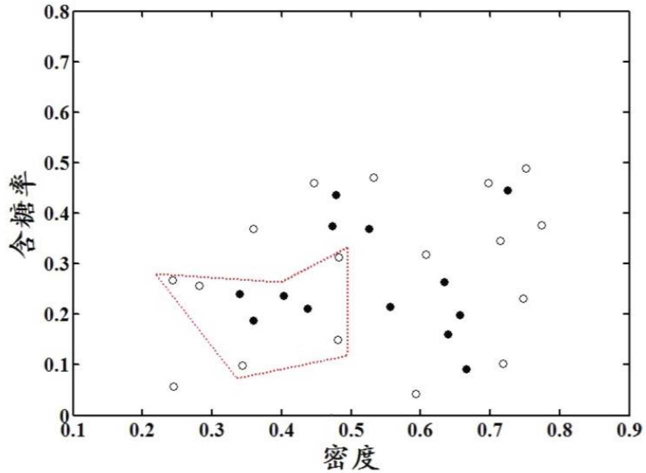
输入: 样本集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
邻域参数$(\epsilon, MinPts)$.

过程:

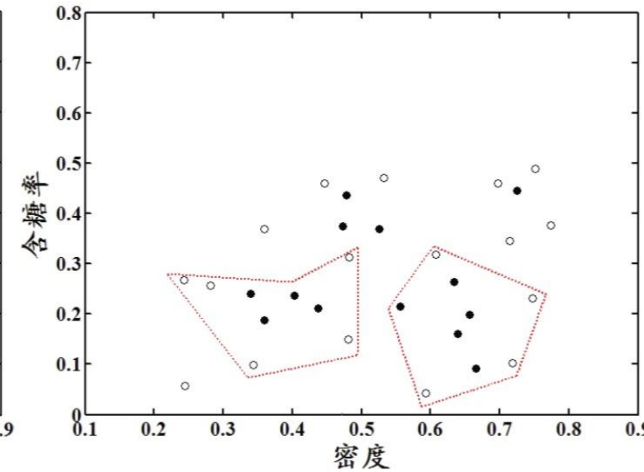
初始化核心对象集合：$\Omega = \emptyset$
for $j = 1, \ldots, m$ **do**
  确定样本$\boldsymbol{x}_j$的ϵ-邻域$N_\epsilon(\boldsymbol{x}_j)$;
  if $|N_\epsilon(\boldsymbol{x}_j)| \geq MinPts$ **then**
    将样本$\boldsymbol{x}_j$加入核心对象集合：$\Omega = \Omega \bigcup \{\boldsymbol{x}_j\}$
  end if
end for
初始化聚类簇数：$k = 0$
初始化未访问样本集合：$\Gamma = D$
while $\Omega \neq \emptyset$ **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 $Q \neq \emptyset$ **do**
    取出队列Q中的首个样本$\boldsymbol{q}$;
    if $|N_\epsilon(\boldsymbol{q})| \geq MinPts$ **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
return 簇划分结果

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

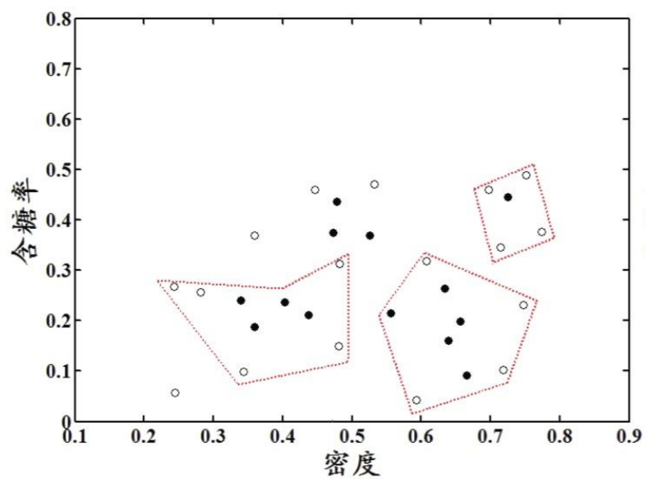
密度聚类



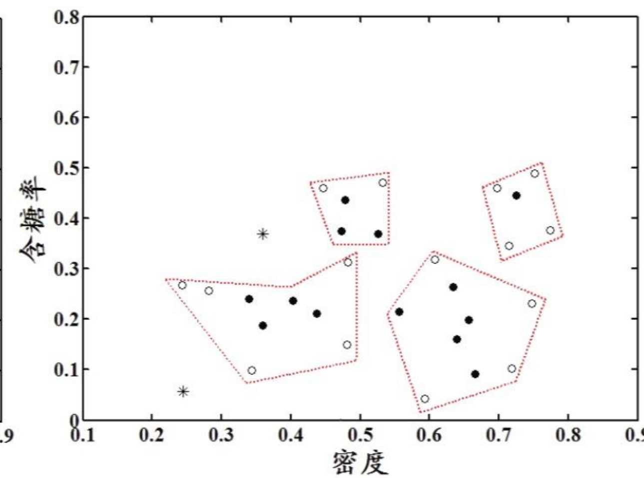
(a) 生成聚类簇C_1



(b) 生成聚类簇C_2



(c) 生成聚类簇C_3



(d) 生成聚类簇C_4

层次聚类

- 层次聚类试图在不同层次对数据集进行划分，从而形成树形的聚类结构。
- 数据集划分既可采用“自底向上”的聚合策略，也可采用“自顶向下”的分拆策略。

层次聚类

- AGNES算法（自底向上的层次聚类算法）

1 将样本中的每一个样本看做一个初始聚类簇

未到预设
聚类簇数

2 在算法运行的每一步中找出距离最近的两个聚类簇进行合并

层次聚类

- 这里两个聚类簇C_i和C_j的距离，可以有3种度量方式。

最小距离：$d_{\min}(C_i, C_j) = \min_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j} dist(\boldsymbol{x}, \boldsymbol{y})$

最大距离：$d_{\max}(C_i, C_j) = \max_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j} dist(\boldsymbol{x}, \boldsymbol{y})$

平均距离：$d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\boldsymbol{x} \in C_i} \sum_{\boldsymbol{y} \in C_j} dist(\boldsymbol{x}, \boldsymbol{y})$

层次聚类

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$d \in \{d_{\min}, d_{\max}, d_{\text{avg}}\}$;
聚类簇数k.

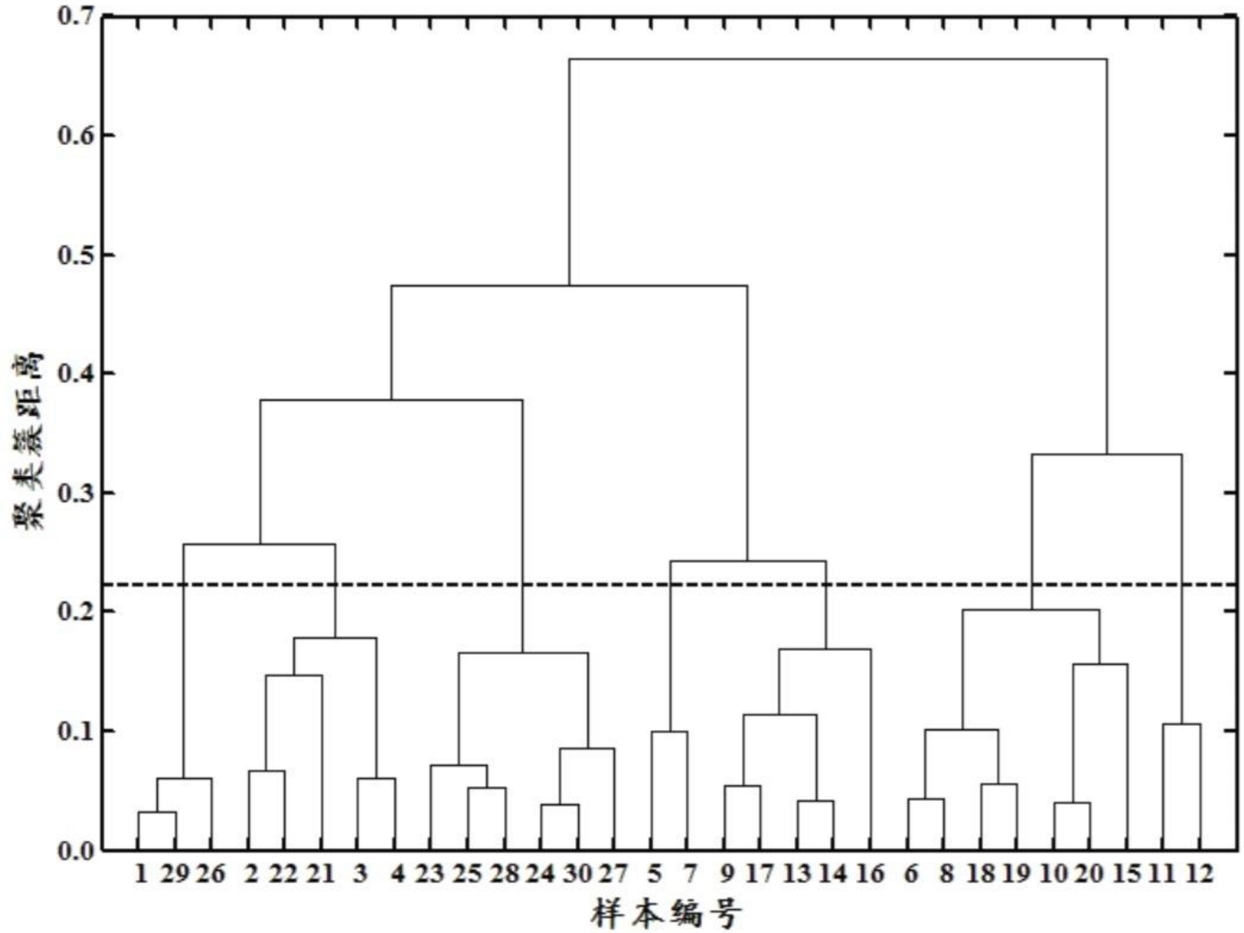
过程:

for $j = 1, \ldots, m$ **do**
    $C_j = \{\boldsymbol{x}_j\}$
end for
for $i = 1, \ldots, m$ **do**
    for $j = i, \ldots, m$ **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^*}, C_{j^*});
    合并(C_{i^*}, C_{j^*})：$C_{i^*} = C_{i^*} \bigcup C_{j^*}$;
    for $j = j^* + 1, \ldots, q$ **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 $j = 1, \ldots, q-1$ **do**
        $M(i^*, j) = d(C_{i^*}, C_j)$;
        $M(j, i^*) = M(i^*, j)$
    end for
    $q = q - 1$
end while
return 簇划分结果

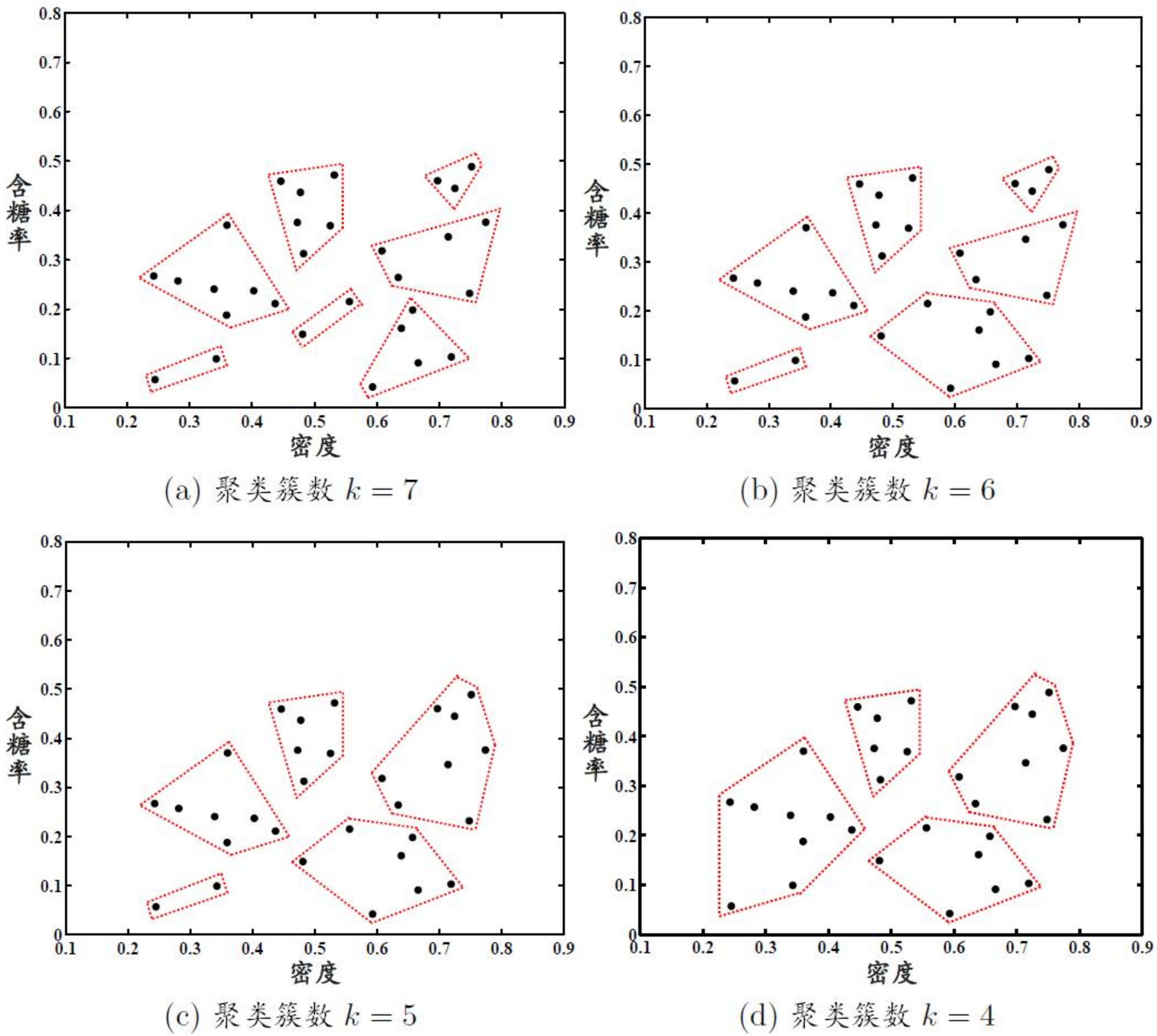
输出: 簇划分$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

层次聚类

- AGNES算法树状图：



层次聚类



(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 $k = 4$

作业

- 给定任意的两个相同长度向量$\boldsymbol{x}, \boldsymbol{y}$，其余弦距离为$1-\frac{\boldsymbol{x}^\top \boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|}$，证明余弦距离不满足传递性，而余弦夹角$\arccos\left(\frac{\boldsymbol{x}^\top \boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|}\right)$满足

- 证明k-means算法的收敛性

- 在k-means算法中替换欧式距离为其他任意的度量，请问“聚类簇”中心如何计算？