



2021年秋季 《机器学习概论》课程

# 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](mailto: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

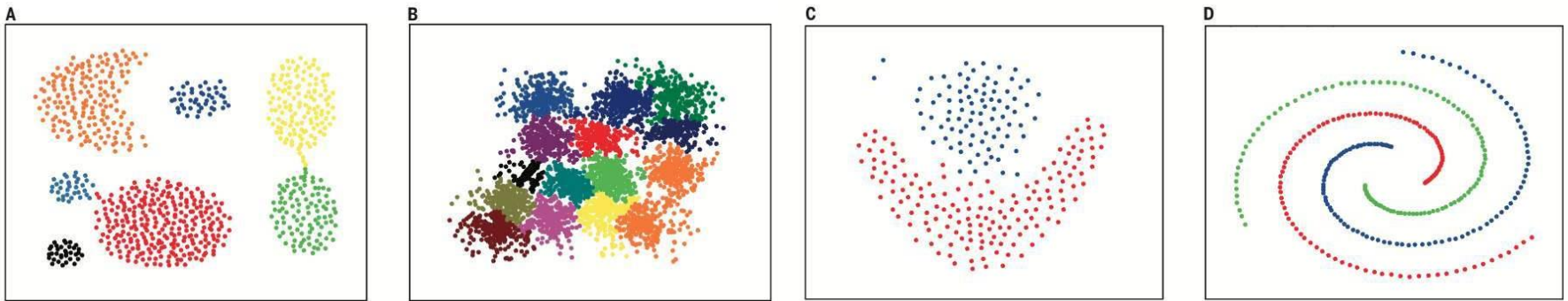
聚类：将数据集中的样本划分为若干个通常是不相交的子集

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  $D = \{x_1, \dots, x_m\}$  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  $\{C_l | l = 1, 2, \dots, k\}$ ，其中  $C_{l'} \cap C_l = \emptyset, \forall l' \neq l$  且  $D = \bigcup_{i=1}^k C_l$
- 用  $\lambda_j \in \{1, 2, \dots, k\}$  表示样本  $x_j$  的簇标记 (cluster label)，即  $x_j \in C_{\lambda_j}$



# 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  $D = \{x_1, \dots, x_m\}$ , 通过聚类给出的簇划分为  $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$
- 参考模型给出的簇划分为  $\mathcal{C}^* = \{C_1^*, C_2^*, \dots, C_k^*\}$
- 令  $\lambda$  与  $\lambda^*$  分别表示与  $\mathcal{C}$  和  $\mathcal{C}^*$  对应的簇标记向量

# 性能度量—外部指标

- 将样本两两配对

$$\begin{aligned} SS &= \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}, & a &= |SS| \\ SD &= \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}, & b &= |SS| \\ DS &= \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}, & c &= |SS| \\ DD &= \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}, & d &= |SS| \end{aligned}$$

- 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

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

$$avg(C) = \frac{1}{|C|} \sum_{i=1}^{|C|} dist(\mu, x_i)$$
$$d_{cen}(C_i, C_j) = dist(\mu_i, \mu_j)$$

## DBI越小越好



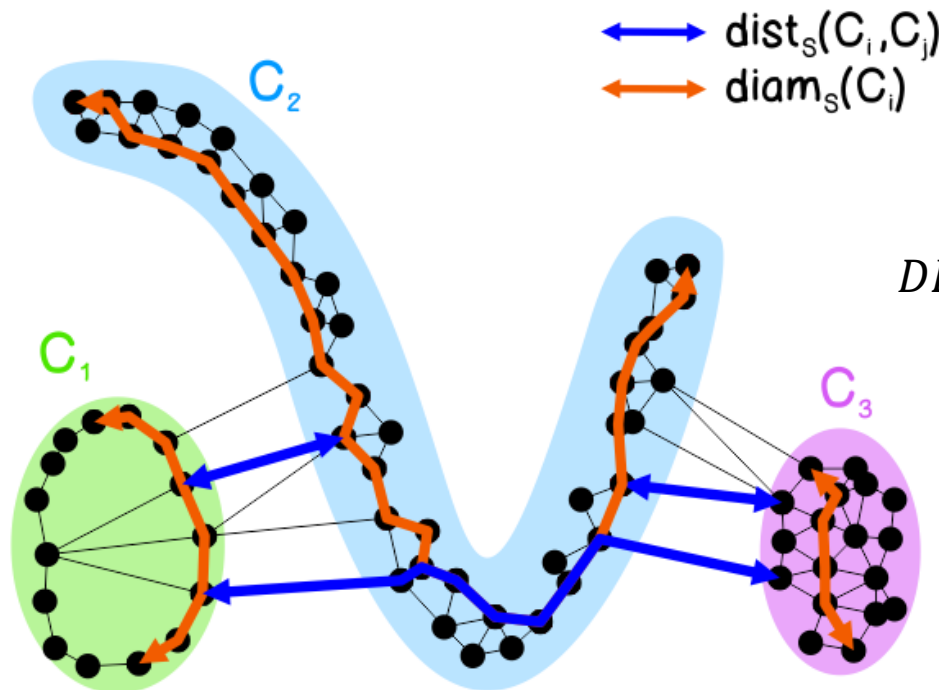
$$avg(C) = (|C - x_7| + |C - x_8| + |C - x_9|)/3$$



# 性能度量—内部指标

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

$$\text{dist}(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} \text{dist}(x_i, x_j) \quad \text{diam}(C) = \max_{1 \leq i < j \leq |C|} \text{dist}(x_i, x_j)$$



$$DI = \min_{1 \leq i \leq k} \left\{ \min_{j \neq i} \left( \frac{\text{dist}(C_i, C_j)}{\max_{1 \leq l \leq k} (\text{diam}(C_l))} \right) \right\}$$

**DI越大越好**

# 距离计算

- 距离度量的性质:

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

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

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

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

- 常用距离:

闵可夫斯基距离 (Minkowski distance) :

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

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

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

- 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

$$\text{dist}_{wmk}(\mathbf{x}_i, \mathbf{x}_j) = \left( w_1 |x_{i1} - y_{j1}|^p + \cdots + w_n |x_{in} - x_{jn}|^p \right)^{\frac{1}{p}}$$

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  $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ ;  
聚类簇数  $k$ .

过程:

初始化每个簇的均值向量

1: 从  $D$  中随机选择  $k$  个样本作为初始均值向量  $\{\mu_1, \mu_2, \dots, \mu_k\}$

2: **repeat**

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

将每个样本分配给最近的簇

4: **for**  $j = 1, \dots, m$  **do**

5: 计算样本  $\mathbf{x}_j$  与各均值向量  $\mu_i$  ( $1 \leq i \leq k$ ) 的距离:  $d_{ji} = \|\mathbf{x}_j - \mu_i\|_2$ ;

6: 根据距离最近的均值向量确定  $\mathbf{x}_j$  的簇标记:  $\lambda_j = \arg \min_{i \in \{1, 2, \dots, k\}} d_{ji}$ ;

7: 将样本  $\mathbf{x}_j$  划入相应的簇:  $C_{\lambda_j} = C_{\lambda_j} \cup \{\mathbf{x}_j\}$ ;

8: **end for**

9: **for**  $i = 1, \dots, k$  **do**

10: 计算新均值向量:  $\mu'_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$ ;

计算每个簇的均值向量

11: **if**  $\mu'_i \neq \mu_i$  **then**

12: 将当前均值向量  $\mu_i$  更新为  $\mu'_i$

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

18: **return** 簇划分结果

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

# 原型聚类—K均值

- 给定数据集  $D = \{x_1, \dots, x_m\}$ , K均值算法针对聚类所得簇划分  $C = \{C_1, C_2, \dots, C_k\}$  最小化平方误差

$$E = \sum_{c=1}^k \sum_{x \in C_c} \|x - \mu_c\|_2^2 \quad \mu_c = \frac{1}{|C_c|} \sum_{x \in C_c} x$$

$E$  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。

$$E(T, \mu) = \sum_{i=1}^m \|x_i - t_i^\top \mu\|_2^2 = \|X - T\mu\|_F^2 \quad T = \begin{bmatrix} t_1^\top \\ t_2^\top \\ \vdots \\ t_m^\top \end{bmatrix}$$

$$t_i^\top = \begin{matrix} 0, & \dots, & 1, & \dots, & 0 \\ 1 & & c_i & & k \end{matrix}$$

$$\mu = [\mu_1, \mu_2, \dots, \mu_k] \in \mathbb{R}^{k \times d}$$



# 原型聚类—K均值

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

## 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;  $T^{(t)} \leftarrow \min_T E(T, \mu^{(t-1)})$
2. 计算每个簇的均值向量;  $\mu^{(t)} \leftarrow \min_{\mu} E(T^{(t)}, \mu)$

**until** 当前均值向量均未更新

# 原型聚类—K均值

- 考虑  $\mathbf{T}^{(t)} \leftarrow \min_{\mathbf{T}} E(\mathbf{T}, \boldsymbol{\mu}^{(t-1)})$
- 由于样本间互不依赖，考虑求解第  $i$  个样本的  $\mathbf{t}_i$

$$\min_{\mathbf{t}_i} \|\mathbf{x}_i - \mathbf{t}_i^\top \boldsymbol{\mu}\|_2^2 = \left\| \mathbf{x}_i - \sum_c t_{ic} \boldsymbol{\mu}_c \right\|_2^2 \iff \min_c \|\mathbf{x}_i - \boldsymbol{\mu}_c\|$$

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

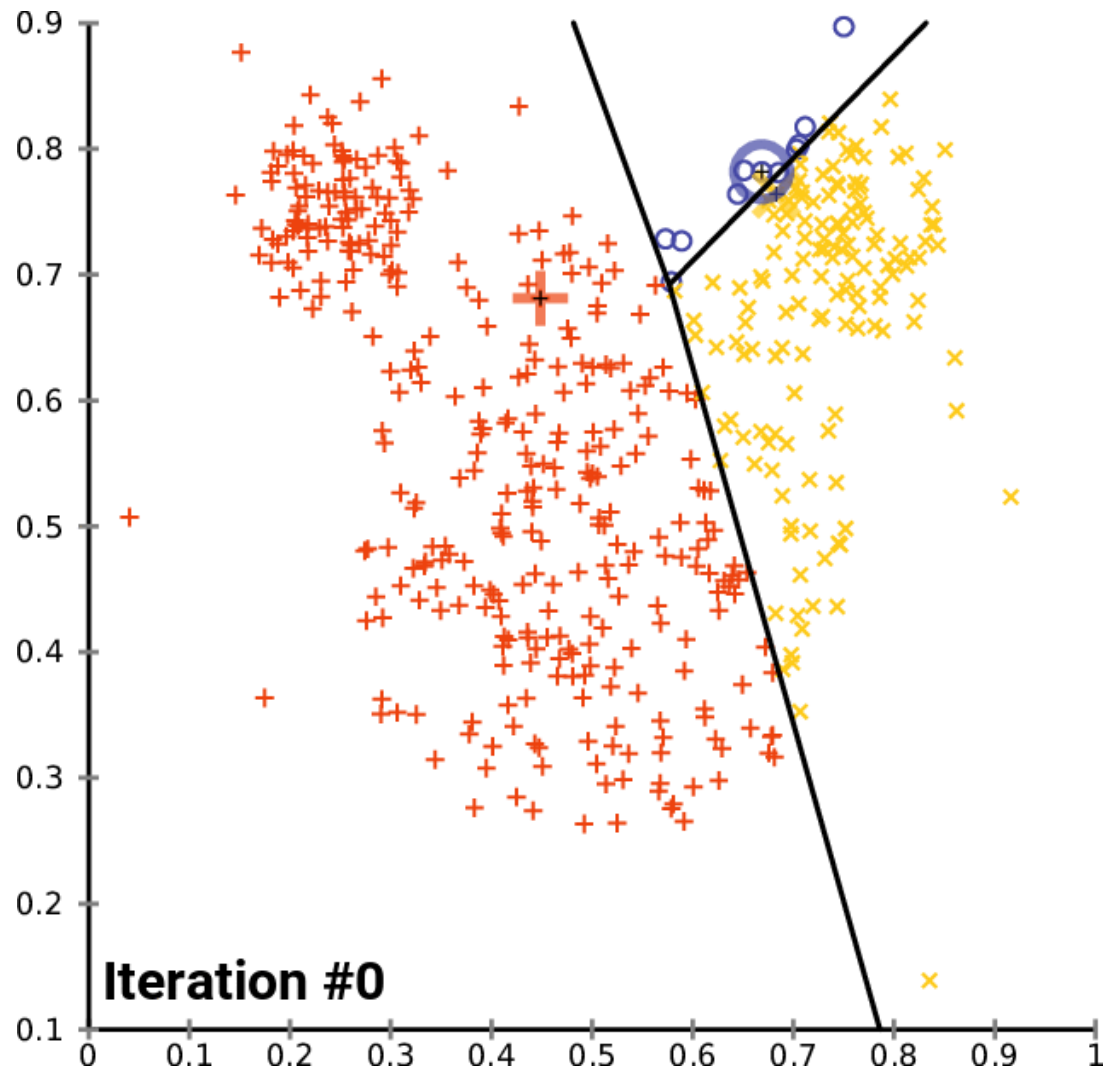
$$\mu = (T^T T)^{-1} T^T X$$

$$T^T T = [t_1, t_2, \dots, t_m] \begin{bmatrix} t_1^T \\ t_2^T \\ \vdots \\ t_m^T \end{bmatrix} = \begin{bmatrix} \tilde{t}_1^T \\ \tilde{t}_2^T \\ \vdots \\ \tilde{t}_k^T \end{bmatrix} [\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_k] = \begin{bmatrix} \tilde{t}_1^T \tilde{t}_1 & \tilde{t}_1^T \tilde{t}_2 & \dots & \tilde{t}_1^T \tilde{t}_k \\ \tilde{t}_2^T \tilde{t}_1 & \tilde{t}_2^T \tilde{t}_2 & \dots & \tilde{t}_2^T \tilde{t}_k \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{t}_k^T \tilde{t}_1 & \tilde{t}_k^T \tilde{t}_2 & \dots & \tilde{t}_k^T \tilde{t}_k \end{bmatrix}$$

$$T^T X = \begin{bmatrix} \tilde{t}_1^T X \\ \tilde{t}_2^T X \\ \vdots \\ \tilde{t}_k^T X \end{bmatrix} = \begin{bmatrix} \sum_{x \in C_1} x \\ \sum_{x \in C_2} x \\ \vdots \\ \sum_{x \in C_k} x \end{bmatrix} = \begin{bmatrix} |c_1| & 0 & \dots & 0 \\ 0 & |c_2| & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & |c_k| \end{bmatrix}$$

$$\mu = (T^T T)^{-1} T^T X = \begin{bmatrix} \frac{1}{|c_1|} \sum_{x \in C_1} x \\ \frac{1}{|c_2|} \sum_{x \in C_2} x \\ \vdots \\ \frac{1}{|c_k|} \sum_{x \in C_k} x \end{bmatrix}$$

# 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 $x_i$ 。

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 $x_1 = (0.697, 0.460)$ ，它与当前均值向量 $\mu_1, \mu_2, \mu_3$ 的距离分别为0.369, 0.506, 0.166，因此 $x_1$ 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

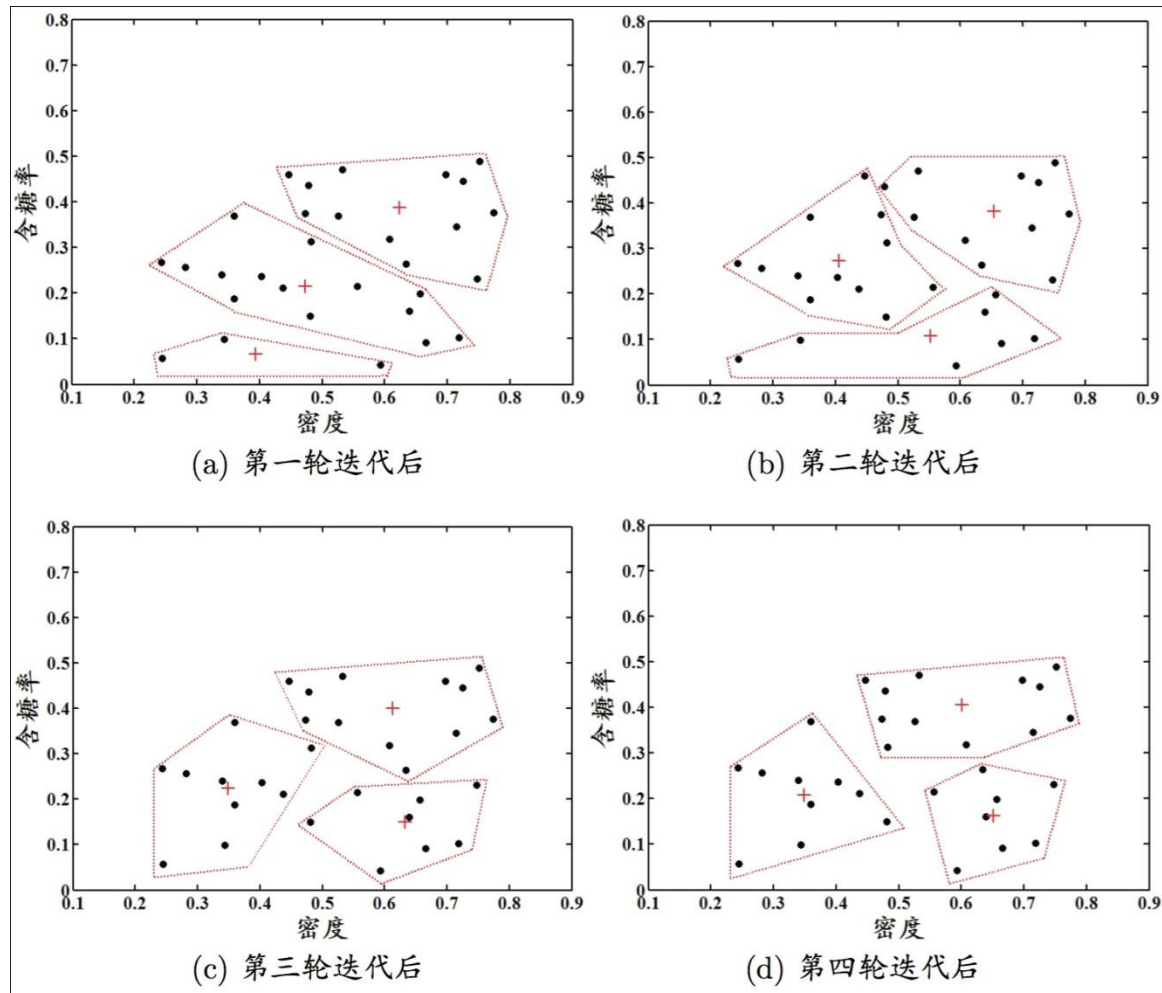
$$\begin{aligned} C_1 &= \{x_5, x_6, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}, x_{20}, x_{23}\} \\ C_2 &= \{x_{11}, x_{12}, x_{16}\} \\ C_3 &= \{x_1, x_2, x_3, x_4, x_{21}, x_{22}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}\} \end{aligned}$$

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



# 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  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ ,  $y_i \in \mathcal{Y}$ , LVQ的目标是学得一组  $d$  维原型向量  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q$ , 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  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ ;  
原型向量个数  $q$ , 各原型向量预设的类别标记  $\{t_1, t_2, \dots, t_q\}$ ;  
学习率  $\eta \in (0, 1)$ .

**过程：**

- 1: 初始化一组原型向量  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q\}$
- 2: **repeat**
- 3:   从样本集  $D$  随机选取样本  $(\mathbf{x}_j, y_j)$ ;
- 4:   计算样本  $\mathbf{x}_j$  与  $\mathbf{p}_i$  ( $1 \leq i \leq q$ ) 的距离:  $d_{ji} = \|\mathbf{x}_j - \mathbf{p}_i\|_2$ ;
- 5:   找出与  $\mathbf{x}_j$  距离最近的原型向量;  $i^* = \arg \min_{i \in \{1, 2, \dots, q\}} d_{ji}$ ;
- 6:   **if**  $y_j = t_{i^*}$  **then**
- 7:      $\mathbf{p}' = \mathbf{p}_{i^*} + \eta \cdot (\mathbf{x}_j - \mathbf{p}_{i^*})$
- 8:   **else**
- 9:      $\mathbf{p}' = \mathbf{p}_{i^*} - \eta \cdot (\mathbf{x}_j - \mathbf{p}_{i^*})$
- 10:   **end if**
- 11:   将原型向量  $\mathbf{p}_{i^*}$  更新为  $\mathbf{p}'$
- 12: **until** 满足停止条件
- 13: **return** 当前原型向量

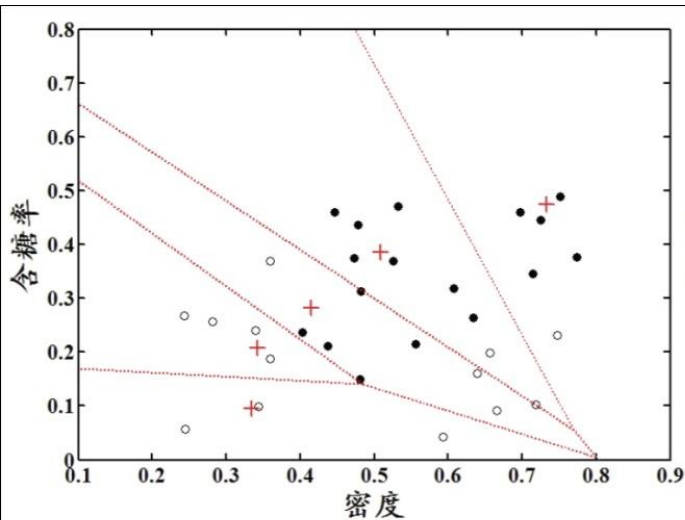
**输出：** 原型向量  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q\}$

根据两者的类别标记是否一致来对原型向量进行更新

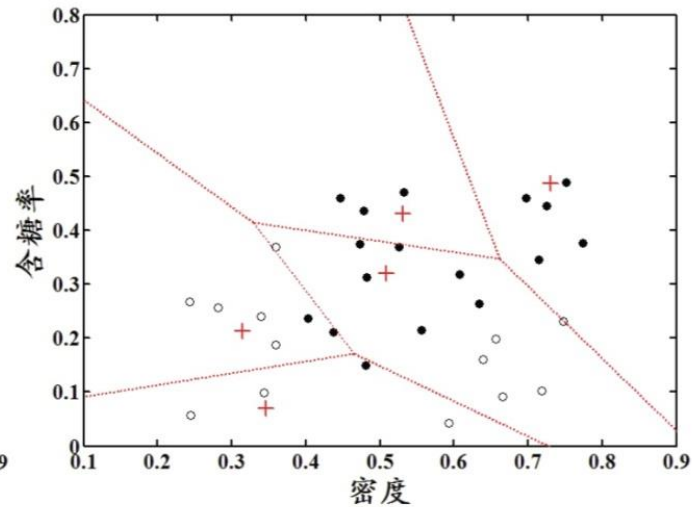
对样本  $\mathbf{x}_j$ , 若最近的原型向量  $\mathbf{p}_{i^*}$  和  $\mathbf{x}_j$  的类别标记相同, 则令  $\mathbf{p}_{i^*}$  向  $\mathbf{x}_j$  的方向靠拢

$$\begin{aligned}\|\mathbf{p}' - \mathbf{x}_j\| &= \|\mathbf{p}_{i^*} + \eta(\mathbf{x}_j - \mathbf{p}_{i^*}) - \mathbf{x}_j\| \\ &= (1 - \eta)\|\mathbf{p}_{i^*} - \mathbf{x}_j\| \\ &< \|\mathbf{p}_{i^*} - \mathbf{x}_j\|\end{aligned}$$

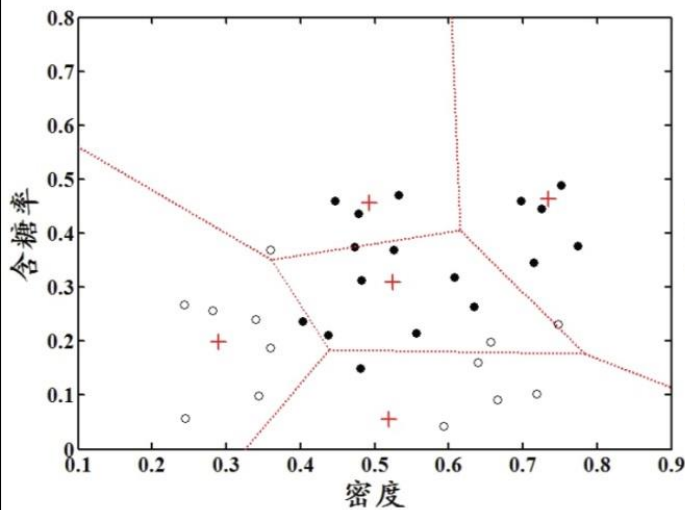
# 原型聚类—学习向量量化



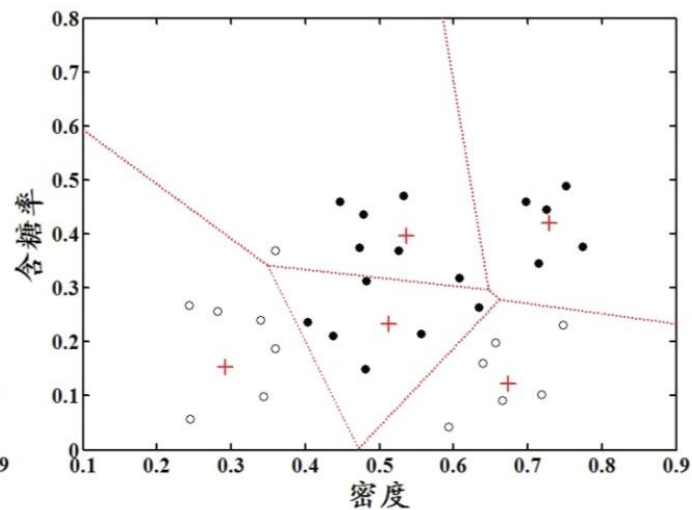
(a) 50轮迭代后



(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 $\mathbf{x}$ ，若 $\mathbf{x}$ 服从多元高斯分布，其概率密度函数为

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

- 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 $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 。

# 原型聚类 – 高斯混合聚类

- 高斯混合分布的定义

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

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 $\alpha_c$ 为相应的“混合系数”， $\sum_c \alpha_c = 1$ 。

假设样本的生成过程由高斯混合分布给出：

- (1) 根据 $\alpha_1, \dots, \alpha_k$ 定义的先验分布选择高斯混合成分， $\alpha_i$ 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

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

- 模型求解：最大化（对数）似然

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

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

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

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

$$\Rightarrow \boldsymbol{\mu}_c = \frac{\sum_{i=1}^m \gamma_{ic} \mathbf{x}_i}{\sum_{i=1}^m \gamma_{ic}}$$

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

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

- 模型求解：最大化（对数）似然

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

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

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

# 原型聚类 – 高斯混合聚类

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

- 模型求解：最大化（对数）似然

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

- 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 $\alpha_c$ 的导数，令其等于0

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

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

# 原型聚类 – 高斯混合聚类

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

- 模型求解：最大化（对数）似然

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

- 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 $\alpha_c$ 的导数，令其等于0

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

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



# 原型聚类 – 高斯混合聚类

假设样本的生成过程由高斯混合分布给出：

- (1) 根据 $\alpha_1, \dots, \alpha_k$ 定义的先验分布选择高斯混合成分， $\alpha_i$ 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

- 数据集 $D = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ 由上述过程生成
- 令随机变量 $z_i \in \{1, \dots, k\}$ 表示生成样本的高斯混合成分，则 $z_j$ 的先验概率 $P(z_i = c) = \alpha_c$

$$P(\mathbf{x}_i, z_i) = P(\mathbf{x}_i | z_i) P(z_i) = \prod_{c=1}^k [P(\mathbf{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \alpha_c]^{\mathbb{I}(z_i=c)}$$

$$\sum_{z_i} P(\mathbf{x}_i, z_i) = \sum_{z_i} \prod_{c=1}^k [P(\mathbf{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \alpha_c]^{\mathbb{I}(z_i=c)} = \sum_{c=1}^k P(\mathbf{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \alpha_c$$

# EM算法

E

基于 $\Theta^t$ 推断隐变量 $Z$ 的分布 $P(Z | X, \Theta^t)$ ，并计算对数似然 $LL(\Theta | X, Z)$ 关于 $Z$ 的期望；

$$\mathcal{L}(P(Z|X, \Theta), \Theta) = \mathbb{E}_{Z \sim P(Z|X, \Theta^t)} LL(\Theta | X, Z) = Q(\Theta | \Theta^t)$$

$$P(z_i = c | x_i) = \frac{P(x_i, z_i = c)}{\sum_{z_i} P(x_i, z_i)} = \frac{\alpha_c P(x_i | \mu_c, \Sigma_c)}{\sum_{c'} \alpha_{c'} P(x_i | \mu_{c'}, \Sigma_{c'})} = \gamma_{ic}$$

$$\begin{aligned} \mathbb{E}_{Z \sim P(Z|X, \Theta^t)} LL(\Theta | X, Z) &= \mathbb{E}_{Z \sim P(Z|X, \Theta^t)} \log \prod_{i=1}^m \prod_{c=1}^k [P(x_i | \mu_c, \Sigma_c) \alpha_c]^{\mathbb{I}(z_i=c)} \\ &= \mathbb{E}_{Z \sim P(Z|X, \Theta^t)} \sum_i^m \sum_{c=1}^k \mathbb{I}(z_i = c) \log P(x_i | \mu_c, \Sigma_c) \alpha_c \\ &= \sum_i^m \sum_{c=1}^k \gamma_{ic} [\log P(x_i | \mu_c, \Sigma_c) + \log \alpha_c] \end{aligned}$$

M

寻找参数最大化期望似然；  $\Theta^{t+1} = \arg \max_{\Theta} Q(\Theta | \Theta^t)$

# 原型聚类 – 高斯混合聚类

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

**过程:**

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

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

# 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( $\epsilon$ , MinPts)来刻画样本分布的紧密程度。
- 基本概念：
  - $\epsilon$ 邻域：对样本 $x_i \in D$ ，其 $\epsilon$ 邻域包含样本集 $D$ 中与 $x_i$ 的距离不大于 $\epsilon$ 的样本；
  - 核心对象：若样本 $x_i$ 的 $\epsilon$ 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 $x_j$ 位于样本 $x_i$ 的 $\epsilon$ 邻域中，且 $x_i$ 是一个核心对象，则称样本 $x_j$ 由 $x_i$ 密度直达；
  - 密度可达：对样本 $x_i$ 与 $x_j$ ，若存在样本序列 $p_1, p_2, \dots, p_n$ ，其中 $p_1 = x_i, p_n = x_j$ 且 $p_{i+1}$ 由 $p_i$ 密度直达，则该两样本密度可达；
  - 密度相连：对样本 $x_i$ 与 $x_j$ ，若存在样本 $x_k$ 使得两样本均由 $x_k$ 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

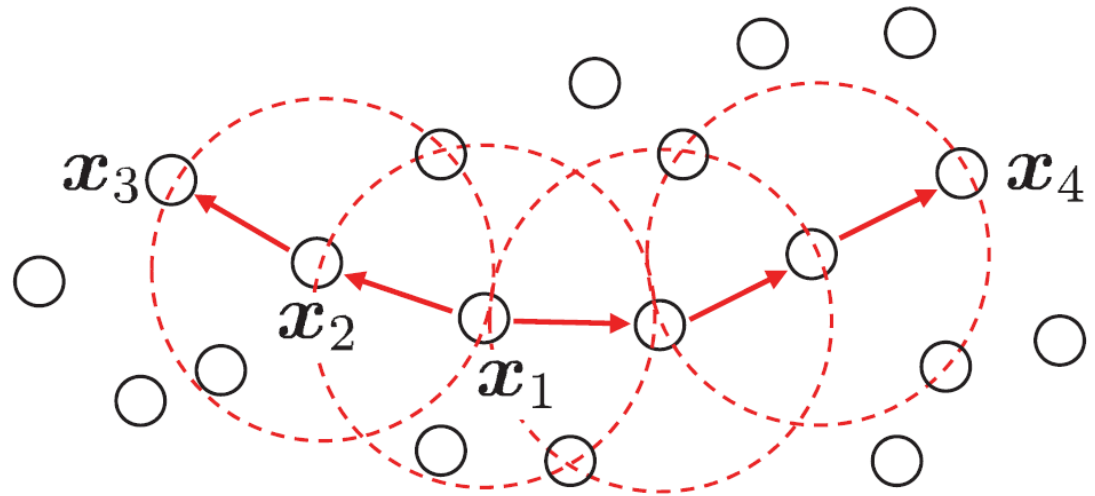
虚线显示出  $\epsilon$  领域。

$x_1$  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。



# 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： $x_i \in C, x_j \in C \Rightarrow x_i$ 与 $x_j$ 密度相连

最大性： $x_i \in C, x_i$ 与 $x_j$ 密度可达 $\Rightarrow x_j \in C$

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

# 密度聚类

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

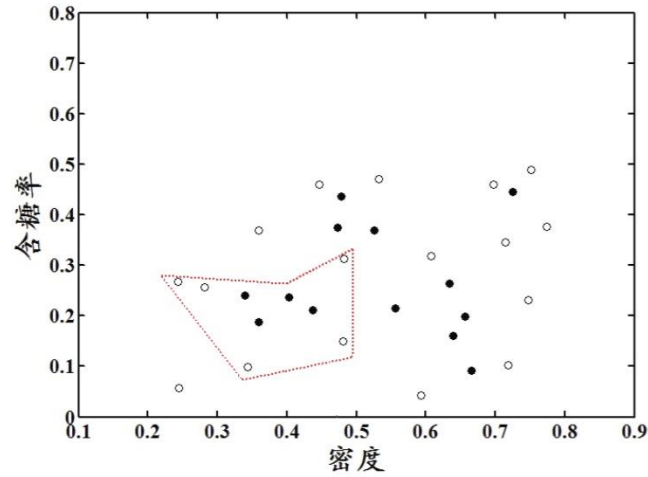
过程:

```
1: 初始化核心对象集合:  $\Omega = \emptyset$ 
2: for  $j = 1, \dots, m$  do
3:   确定样本  $\mathbf{x}_j$  的  $\epsilon$ -邻域  $N_\epsilon(\mathbf{x}_j)$ ;
4:   if  $|N_\epsilon(\mathbf{x}_j)| \geq MinPts$  then
5:     将样本  $\mathbf{x}_j$  加入核心对象集合:  $\Omega = \Omega \cup \{\mathbf{x}_j\}$ 
6:   end if
7: end for
8: 初始化聚类簇数:  $k = 0$ 
9: 初始化未访问样本集合:  $\Gamma = D$ 
10: while  $\Omega \neq \emptyset$  do
11:   记录当前未访问样本集合:  $\Gamma_{old} = \Gamma$ ;
12:   随机选取一个核心对象  $\mathbf{o} \in \Omega$ , 初始化队列  $Q = \langle \mathbf{o} \rangle$ ;
13:    $\Gamma = \Gamma \setminus \{\mathbf{o}\}$ ;
14:   while  $Q \neq \emptyset$  do
15:     取出队列  $Q$  中的首个样本  $\mathbf{q}$ ;
16:     if  $|N_\epsilon(\mathbf{q})| \geq MinPts$  then
17:       令  $\Delta = N_\epsilon(\mathbf{q}) \cap \Gamma$ ;
18:       将  $\Delta$  中的样本加入队列  $Q$ ;
19:        $\Gamma = \Gamma \setminus \Delta$ ;
20:     end if
21:   end while
22:    $k = k + 1$ , 生成聚类簇  $C_k = \Gamma_{old} \setminus \Gamma$ ;
23:    $\Omega = \Omega \setminus C_k$ 
24: end while
25: return 簇划分结果
```

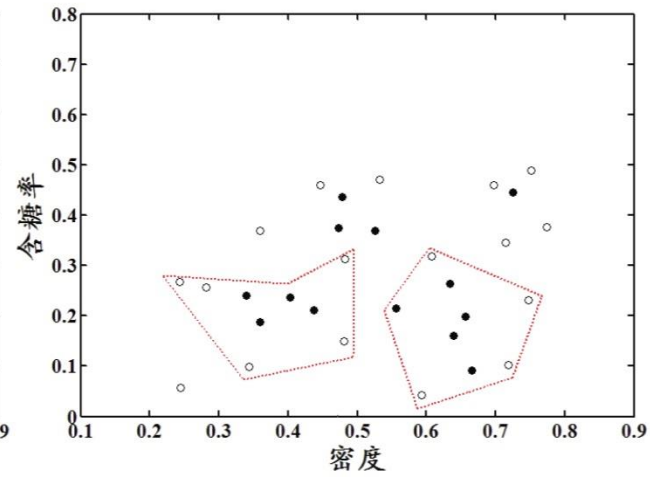
输出: 簇划分  $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$



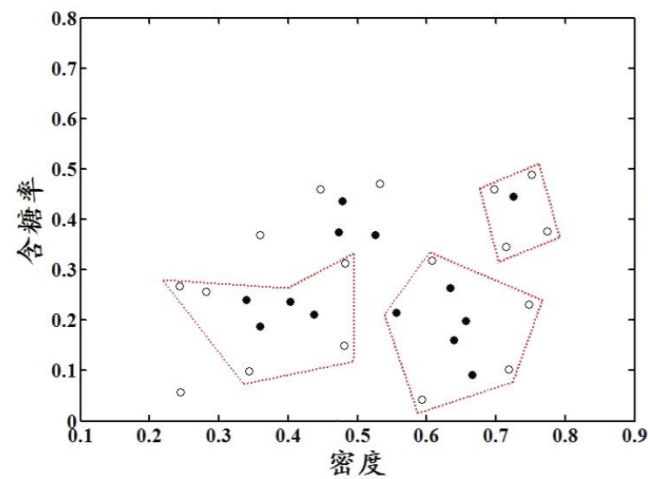
# 密度聚类



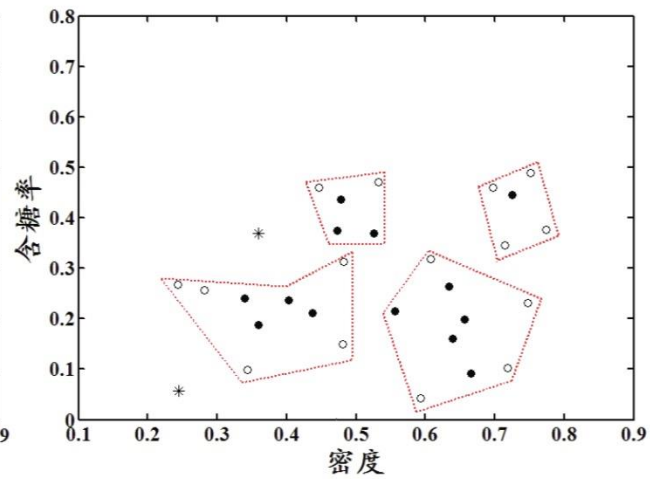
(a) 生成聚类簇 $C_1$



(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。

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

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

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

# 层次聚类

**输入:** 样本集  $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ ;  
聚类簇距离度量函数  $d \in \{d_{\min}, d_{\max}, d_{\text{avg}}\}$ ;  
聚类簇数  $k$ .

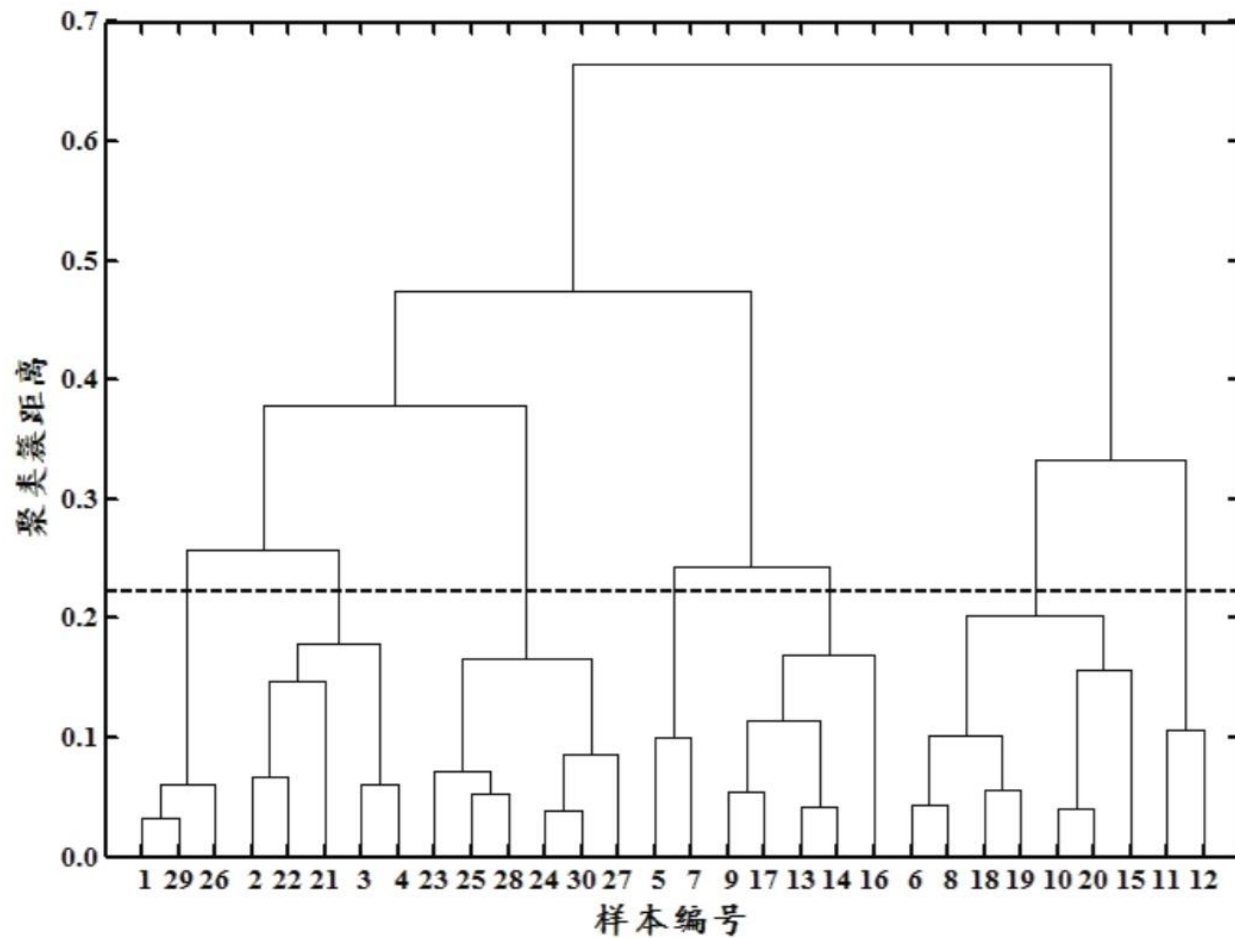
**过程:**

```
1: for  $j = 1, \dots, m$  do
2:    $C_j = \{\mathbf{x}_j\}$ 
3: end for
4: for  $i = 1, \dots, m$  do
5:   for  $j = i, \dots, m$  do
6:      $M(i, j) = d(C_i, C_j)$ ;
7:      $M(j, i) = M(i, j)$ 
8:   end for
9: end for
10: 设置当前聚类簇个数:  $q = m$ 
11: while  $q > k$  do
12:   找出距离最近的两个聚类簇  $(C_{i^*}, C_{j^*})$ ;
13:   合并  $(C_{i^*}, C_{j^*})$ :  $C_{i^*} = C_{i^*} \cup C_{j^*}$ ;
14:   for  $j = j^* + 1, \dots, q$  do
15:     将聚类簇  $C_j$  重编号为  $C_{j-1}$ 
16:   end for
17:   删除距离矩阵  $M$  的第  $j^*$  行与第  $j^*$  列;
18:   for  $j = 1, \dots, q - 1$  do
19:      $M(i^*, j) = d(C_{i^*}, C_j)$ ;
20:      $M(j, i^*) = M(i^*, j)$ 
21:   end for
22:    $q = q - 1$ 
23: end while
24: return 簇划分结果
```

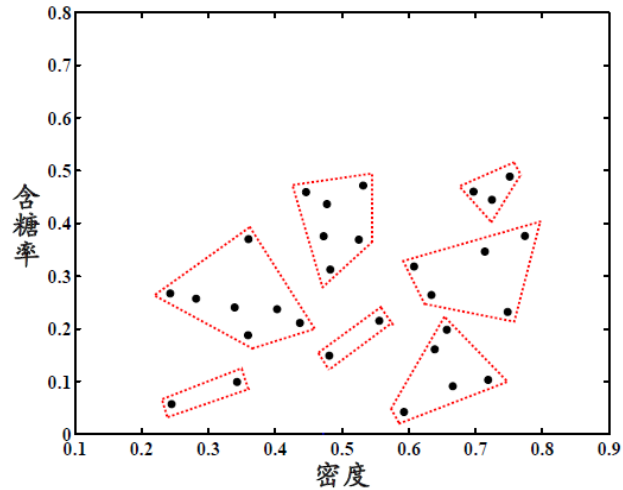
**输出:** 簇划分  $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$

# 层次聚类

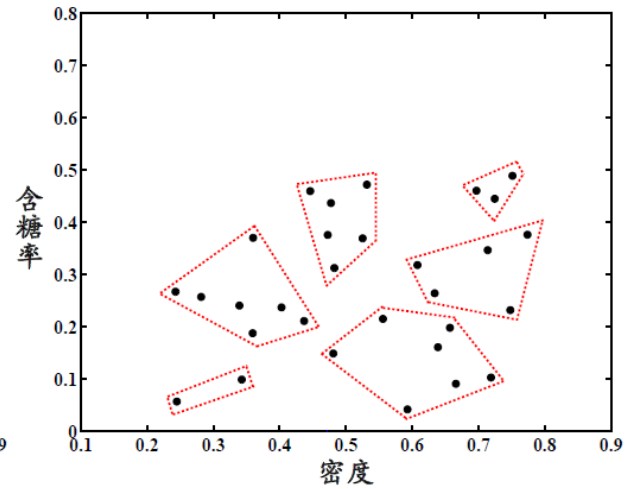
- AGNES算法树状图：



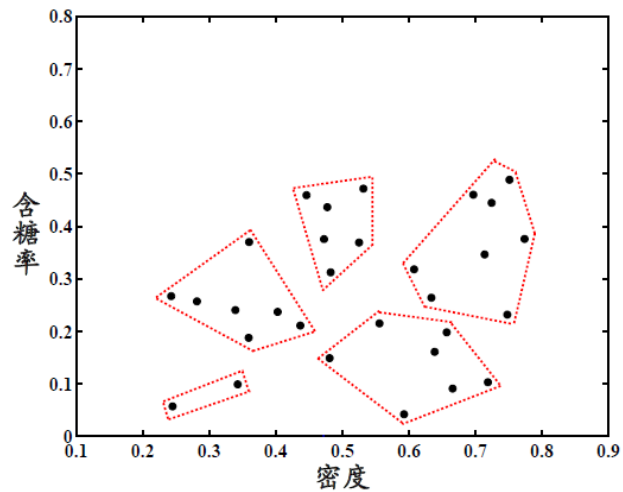
# 层次聚类



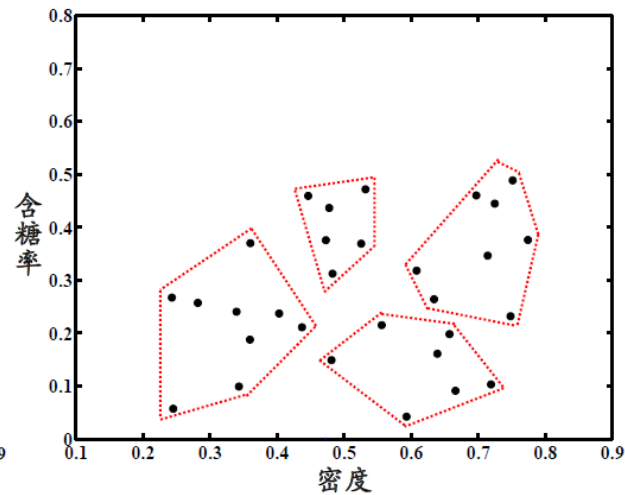
(a) 聚类簇数  $k = 7$



(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 $x, y$ ，其余弦距离为 $1 - \frac{x^\top y}{|x||y|}$ ，证明余弦距离不满足传递性，而余弦夹角 $\arccos\left(\frac{x^\top y}{|x||y|}\right)$ 满足
- 直接应用EM算法，推导高斯混合聚类均值、方差、和混合系数的最优值。