第九章：聚类

大纲

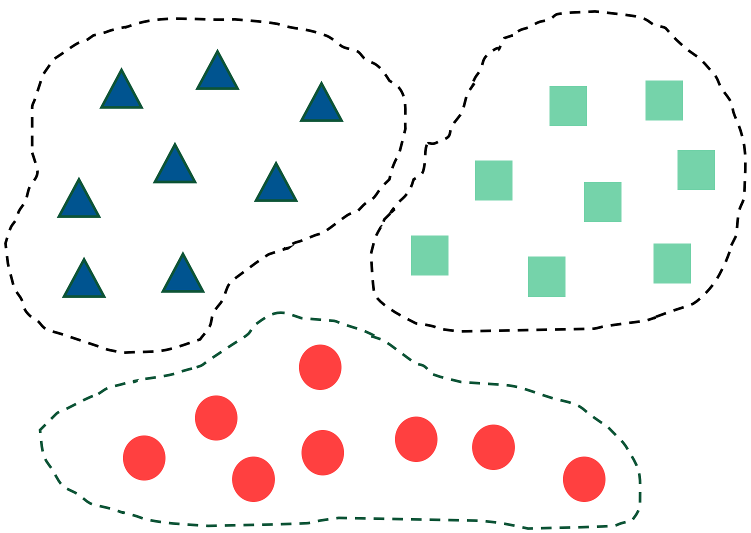
- 聚类任务
- 性能度量
- 距离计算
- 原型聚类
- 密度聚类
- 层次聚类

大纲

- 聚类任务
- 性能度量
- 距离计算
- 原型聚类
- 密度聚类
- 层次聚类

聚类任务

- 聚类在“无监督学习”任务中研究最多、应用最广.
- 聚类目标：将数据集中的样本划分为若干个通常不相交的子集（“**簇**”，cluster）.
- 聚类既可以作为一个单独过程（用于找寻数据内在的分布结构），也可作为分类等其他学习任务的前驱过程.



聚类任务

形式化描述

假定样本集 $D=\{x_1,x_2,\cdots,x_m\}$ 包含m个无标记样本,每个样本 $x_i=(x_{i1};x_{i2};\cdots;x_{in})$ 是一个n维的特征向量，聚类算法将样本集 D 划分成 k 个不相交的簇$\{C_l|l=1,2,...,k\}$ 。其中$C_{l'}\bigcap_{l'\neq l}C_l=\phi$，且 $D=\bigcup_{l=1}^{k}C_l$。

相应地，用$\lambda_j\in\{1,2,\cdots,k\}$表示样本 x_j的 “簇标记” （即cluster label）,即$x_j\in C_{\lambda_j}$。于是，聚类的结果可用包含 m 个元素的簇标记向量 $\lambda=\{\lambda_1;\lambda_2;\cdots;\lambda_m\}$ 表示。

大纲

- 聚类任务
- 性能度量
- 距离计算
- 原型聚类
- 密度聚类
- 层次聚类

性能度量

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

我们希望“**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

对数据集 $D=\{x_1,x_2,\ldots,x_m\}$，假定通过**聚类**得到的簇划分为 $C=\{C_1,C_2,\ldots,C_k\}$，**参考模型**给出的簇划分为 $C^*=\{C_1^*,C_2^*,\ldots,C_s^*\}$.相应地，令 λ 与 λ^* 分别表示与 C 和 C^* 对应的簇标记向量.

我们将样本两两配对考虑，定义

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

性能度量 - 外部指标

- 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）

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

[0,1]区间内，
越大越好.

性能度量 – 内部指标

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

  簇C内样本间的平均距离

$$avg(C)=\frac{2}{|C|(|C|-1)}\sum_{1\le i\le j\le |C|} dist(x_i, x_j)$$

簇C内样本间的最远距离

$$diam(C)=max_{1\le i\le j\le |C|} dist(x_i, x_j)$$

簇C_i与簇C_j最近样本间的距离

$$d_{min}(C)=min_{x_i\in C_i, x_j\in C_j} dist(x_i, x_j)$$

簇C_i与簇C_j中心点间的距离

$$d_{cen}(C)=dist(\mu_i, \mu_j)$$

性能度量 – 内部指标

- DB指数（Davies-Bouldin Index, DBI）

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

越小越好.

- Dunn指数（Dunn Index, DI）

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

越大越好.

大纲

- 聚类任务
- 性能度量
- 距离计算
- 原型聚类
- 密度聚类
- 层次聚类

距离计算

- 距离度量的性质：
  - 非负性：$dist(x_i, x_j) \geq 0$
  - 同一性：$dist(x_i, x_j) = 0$ 当且仅当 $x_i = x_j$
  - 对称性：$dist(x_i, x_j) = dist(x_j, x_i)$
  - 直递性：$dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$

距离计算

- 距离度量的性质：
  - 非负性： $dist(x_i, x_j) \geq 0$
  - 同一性： $dist(x_i, x_j) = 0$ 当且仅当 $x_i = x_j$
  - 对称性： $dist(x_i, x_j) = dist(x_j, x_i)$
  - 直递性： $dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$
- 常用距离：
  - 闵可夫斯基距离（Minkowski distance）：

$$dist(x_i, x_j) = \left(\sum_{u=1}^{n} |x_{iu} - x_{ju}|^p \right)^{\frac{1}{p}}$$

p=2: 欧氏距离（Euclidean distance）.

p=1：曼哈顿距离（Manhattan distance）.

距离计算

- 属性介绍
  - 连续属性 (continuous attribute)
    在定义域上有无穷多个可能的取值
  - 离散属性 (categorical attribute)
    在定义域上是有限个可能的取值

距离计算

- 属性介绍
  - **连续属性** (continuous attribute)
    在定义域上有无穷多个可能的取值
  - **离散属性** (categorical attribute)
    在定义域上是有限个可能的取值
  - **有序属性** (ordinal attribute)
    例如定义域为{1,2,3}的离散属性，“1”与“2”比较接近、与“3”比较远，称为“有序属性”。
  - **无序属性** (non-ordinal attribute)
    例如定义域为{飞机，火车，轮船}这样的离散属性，不能直接在属性值上进行计算，称为“无序属性”。

距离度量

- Value Difference Metric, VDM（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，k 为样本簇数，则**属性** u 上两个**离散值** a 与 b 之间的VDM距离为

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

属性u取值a的样本中，归入第i个簇的概率

距离度量

- MinkovDMp（处理混合属性）：

$$MinkovDM_p(x_i, x_j) = \left(\sum_{u=1}^{n_c} |x_{iu} - x_{ju}|^p + \sum_{u=n_c+1}^{n} VDM_p(x_{iu}, x_{ju}) \right)^{\frac{1}{p}}$$

- 加权距离（样本中不同属性的重要性不同时）：

$$dist(x_i, x_j) = (\omega_1 \cdot |x_{i1} - x_{j1}|^p + \cdots + \omega_n \cdot |x_{in} - x_{jn}|^p)^{\frac{1}{p}}$$

$$\omega_i \geq 0, \sum_{i=1}^{n} \omega_i = 1$$

距离度量

- 连续属性（有序属性）数据如下：
  - **样本x_1**：重量=3.0kg，含糖量=0.25
  - **样本x_2**：重量=3.5kg，含糖量=0.18

计算：

- **曼哈顿距离（p=1）**

$$\mathrm{dist}_{\mathrm{man}}(x_i, x_j) = \sum_{u=1}^{n} |x_{iu} - x = |3.0 - 3.5| + |0.25 - 0.18| = 0.5 + 0.07 = 0.57$$

- **欧氏距离（p=2）**

$$\mathrm{dist}_{\mathrm{ed}}(x_i, x_j) = \sqrt{\sum_{u=1}^{n} (x_{iu} - x_{ju})^2} = \sqrt{(3.0 - 3.5)^2 + (0.25 - 0.18)^2} = \sqrt{0.25 + 0.0049} \approx 0.505$$

距离度量

- 无序属性数据如下：
  - 假设“根蒂”为无序属性，取值集合为{蜷缩, 稍蜷, 硬挺}，统计分布如下：
    - 蜷缩：在好瓜中出现**4**次，坏瓜中出现**1**次
    - 稍蜷：在好瓜中出现**2**次，坏瓜中出现**3**次
    - 硬挺：在好瓜中出现**0**次，坏瓜中出现**2**次

计算VDM距离：

- 蜷缩在好瓜中的比例：**4/5**，坏瓜中：**1/5**
- 稍蜷在好瓜中的比例：**2/5**，坏瓜中：**3/5**

$$\mathrm{VDM}_p(a,b) = \sum_{i=1}^{k} \left| \frac{m_{u,a,i}}{m_{u,a}} - \frac{m_{u,b,i}}{m_{u,b}} \right|^p = \left| \frac{4}{5} - \frac{2}{5} \right|^2 + \left| \frac{1}{5} - \frac{3}{5} \right|^2 = 0.16 + 0.16 = 0.32$$

大纲

- 聚类任务
- 性能度量
- 距离计算
- 原型聚类
- 密度聚类
- 层次聚类

原型聚类

- 原型聚类

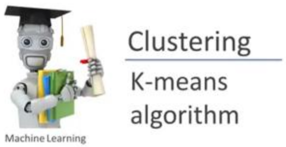
  也称为“基于原型的聚类” (prototype-based clustering), 此类算法假设聚类结构能通过一组原型刻画。

- 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。

- 接下来，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

给定数据集 $D=\{x_1,x_2,\cdots,x_m\}$， k均值算法针对聚类所得簇划分 $C=\{C_1,C_2,\cdots,C_k\}$ 最小化平方误差

$$E=\sum_{i=1}^{k}\sum_{x\in C_j}\|x-\mu_i\|_2^2$$

其中，μ_i是簇 C_i 的均值向量。

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。

原型聚类 – *k*均值算法

给定数据集 $D=\{x_1,x_2,\cdots,x_m\}$， k均值算法针对聚类所得簇划分 $C=\{C_1,C_2,\cdots,C_k\}$ 最小化平方误差

$$E=\sum_{i=1}^{k}\sum_{x\in C_j}\|x-\mu_i\|_2^2$$

其中，μ_i 是簇 C_i 的均值向量。

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。

- 算法流程（迭代优化）：

  初始化每个簇的均值向量

  repeat

  1. （更新）簇划分；
  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均值算法

- 算法伪代码：

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$\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_k\}$
repeat
  令$C_i=\emptyset\ (1\leq i\leq k)$
  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

1 E步

  for $i=1,\ldots,k$ **do**
    计算新均值向量：$\boldsymbol{\mu}'_i=\frac{1}{|C_i|}\sum_{\boldsymbol{x}\in C_i}\boldsymbol{x}$;
    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

2 M步

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均值算法

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

原型聚类 – k均值算法

- 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 μ_1, μ_2, μ_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

$$C_1=\{x_5, x_6, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}, x_{20}, x_{23}\}$$

$$C_2=\{x_{11}, x_{12}, x_{16}\}$$

$$C_3=\{x_1, x_2, x_3, x_4, x_{21}, x_{22}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}\}$$

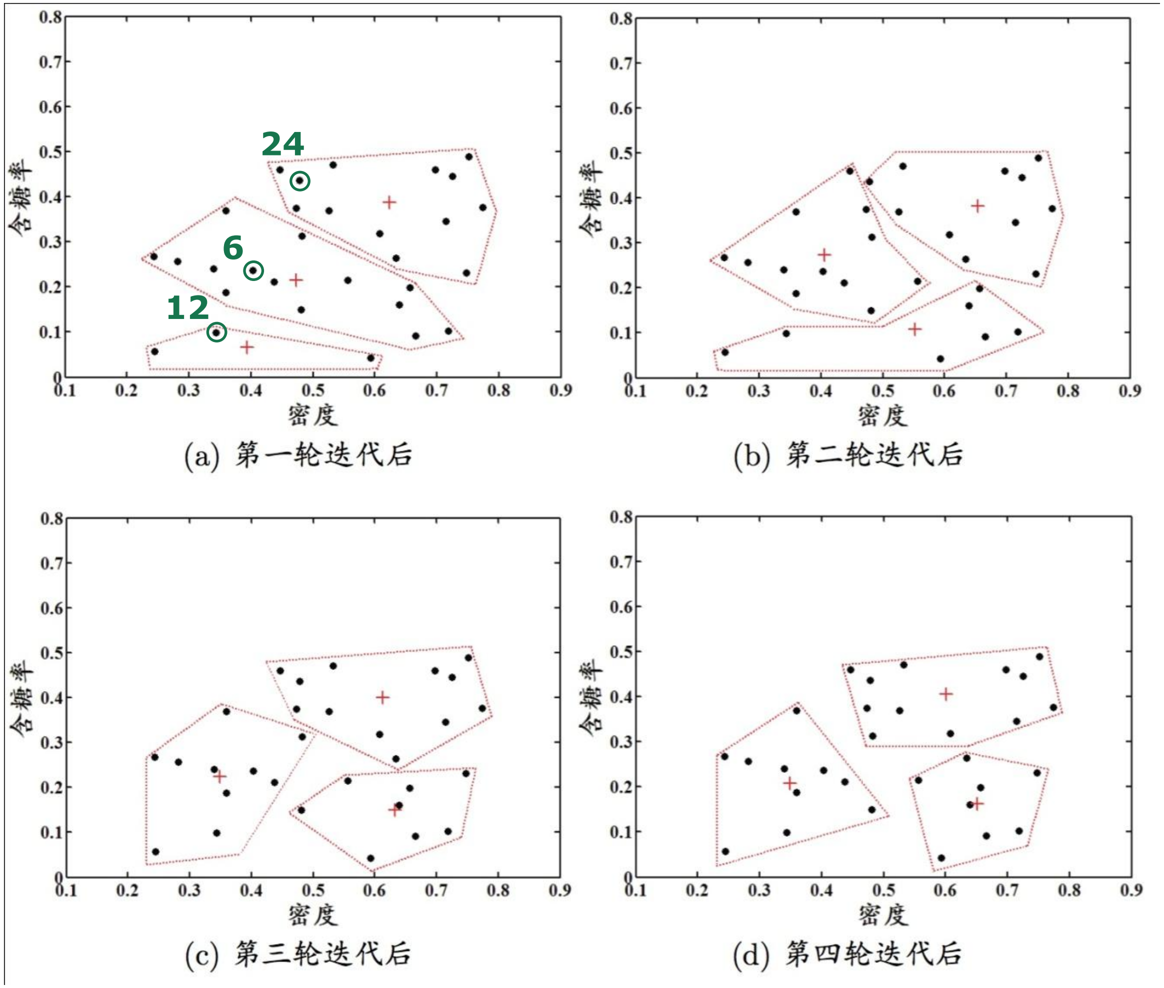
于是，可以从分别求得新的均值向量

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

不断重复上述过程，如下图所示。

原型聚类 - k均值算法

- 聚类结果：



原型聚类 – 学习向量量化

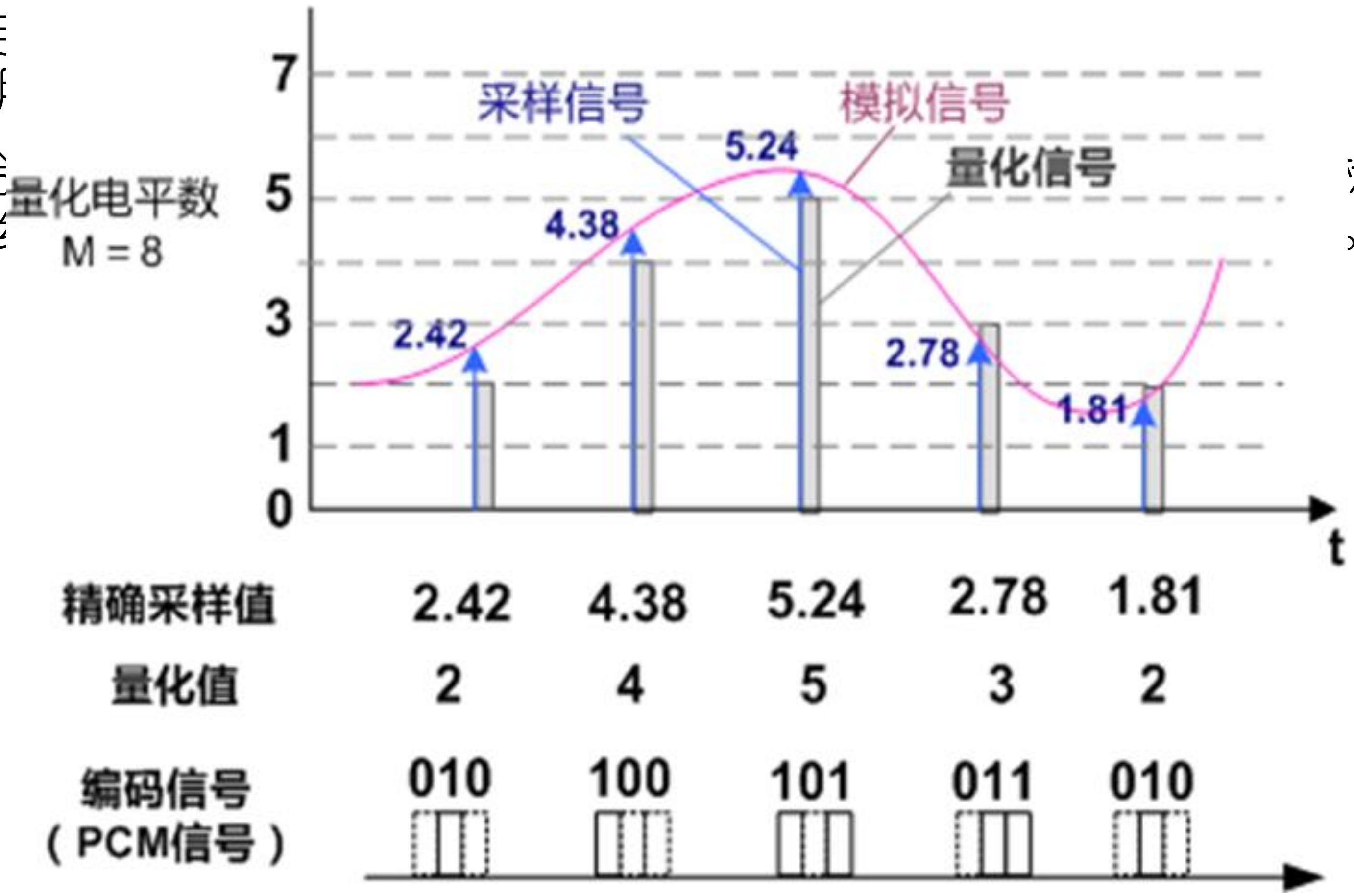
- 学习向量量化（Learning Vector Quantization, LVQ）

与 … 学习过程

中利 …

给 … 标是学得一

组 n … 。



原型聚类 – 学习向量量化

- 算法伪代码：

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$\{\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_q\}$
repeat
  从样本集D随机选取样本$(\boldsymbol{x}_j,y_j)$；
  计算样本$\boldsymbol{x}_j$与$\boldsymbol{p}_i$ $(1\leq i\leq q)$的距离：$d_{ji}=||\boldsymbol{x}_j-\boldsymbol{p}_i||_2$；
  找出与$\boldsymbol{x}_j$距离最近的原型向量；$i^*=\arg\min_{i\in\{1,2,\ldots,q\}} d_{ji}$；
  if $y_j=t_{i^*}$ **then**
    $\boldsymbol{p}'=\boldsymbol{p}_{i^*}+\eta\cdot(\boldsymbol{x}_j-\boldsymbol{p}_{i^*})$
  else
    $\boldsymbol{p}'=\boldsymbol{p}_{i^*}-\eta\cdot(\boldsymbol{x}_j-\boldsymbol{p}_{i^*})$
  end if
  将原型向量$\boldsymbol{p}_{i^*}$更新为$\boldsymbol{p}'$
until 满足停止条件
return 当前原型向量

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

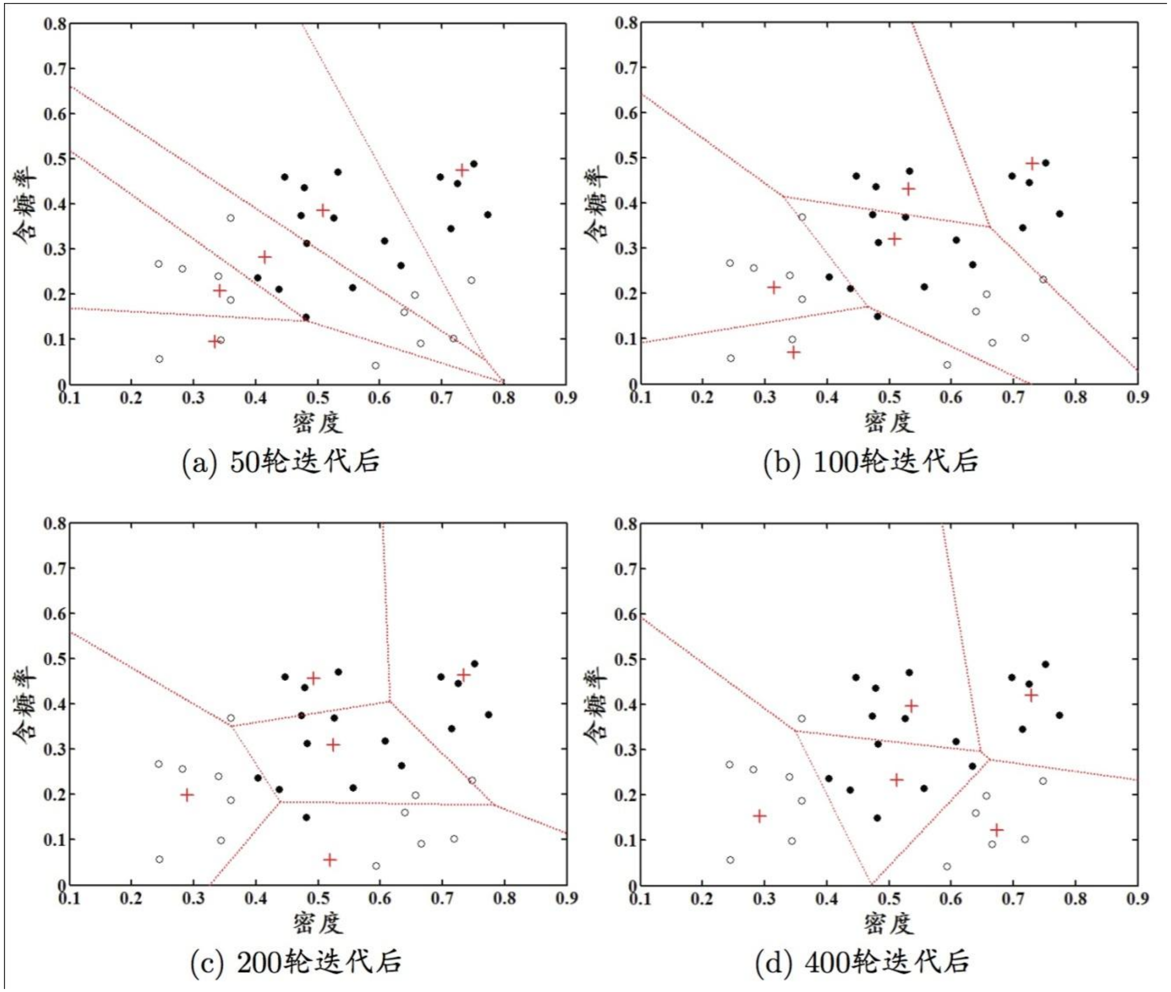
原型聚类 – 学习向量量化

- 聚类效果：

$\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4, \boldsymbol{p}_5$

$\boldsymbol{x}_1$与$\boldsymbol{p}_5$最近且同类，更新为：

$$\boldsymbol{p}' = \boldsymbol{p}_5 + \eta \cdot (\boldsymbol{x}_1 - \boldsymbol{p}_5)$$



(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：

- 多元高斯分布的定义

对 n 维样本空间中的随机向量 x，若 x 服从高斯分布，其概率密度函数为

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

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

原型聚类 – 高斯混合聚类

- 高斯混合分布的定义

$$p_M(x) = \sum_{i=1}^{k} \alpha_i p(x|\mu_i, \Sigma_i)$$

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，μ_i 与 Σ_i 是第 i 个高斯混合成分的参数。而 $\alpha_i > 0$ 为相应的“混合系数”，$\sum_{i=1}^{k} \alpha_i = 1$

i

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：

首先，根据 $\alpha_1, \alpha_2, \cdots, \alpha_k$ 定义的先验分布选择高斯混合成分，其中 α_i为选择第 i个混合成分的概率；

然后，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。

- 定义隐变量 $z_j \in \{1, 2, \cdots k\}$

$$P(z_j = i) = \alpha_i \qquad\qquad p_{\mathrm{M}}(z_j = i \mid x_j) = \gamma_{ji}$$

- 当高斯混合分布已知，确定样本的簇标记为

$$\lambda_j = \arg\max_{i=1,2,\cdots,k} \gamma_{ji}$$

原型聚类 – 高斯混合聚类

- 模型求解：最大化（对数）似然

$$
\begin{aligned}
LL(D) &= \ln\left(\prod_{j=1}^{m} p_M(x_j)\right) \\
&= \sum_{j=1}^{m} \ln\left(\sum_{i=1}^{k} \alpha_i \cdot p\left(x_j | \mu_i, \Sigma_i\right)\right)
\end{aligned}
$$

采用EM算法进行迭代求解。令：

$$
\frac{\partial LL(D)}{\partial \mu_i} = 0 \quad \Longrightarrow \quad \mu_i = \frac{\sum_{j=1}^{m} \gamma_{ji} x_j}{\sum_{j=1}^{m} \gamma_{ji}}
$$

高斯混合聚类 – 模型求解（续）

$$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i} = \frac{\partial LL(D)}{\partial p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)} \cdot \frac{\partial p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)}{\partial \boldsymbol{\mu}_i}$$

$$\begin{aligned}\frac{\partial LL(D)}{\partial p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \ _i\right)} &= \frac{\partial \sum_{j=1}^{m} \ln\left(\sum_{l=1}^{k} \alpha_l \cdot p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\right)\right)}{\partial p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)} \\ &= \sum_{j=1}^{m} \frac{\partial \ln\left(\sum_{l=1}^{k} \alpha_l \cdot p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\right)\right)}{\partial p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)} \\ &= \sum_{j=1}^{m} \frac{\alpha_i}{\sum_{l=1}^{k} \alpha_l \cdot p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\right)}\end{aligned}$$

$$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i} = \sum_{j=1}^{m} \frac{\alpha_i}{\sum_{l=1}^{k} \alpha_l \cdot p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_l, \ _l\right)} \cdot p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \cdot \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right) = 0$$

$$\sum_{j=1}^{m} \gamma_{ji}\left(\mathbf{X}_j - \boldsymbol{\mu}_i\right) = 0$$

$$\begin{aligned}\frac{\partial p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)}{\partial \boldsymbol{\mu}_i} &= \frac{\partial \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)^{\top} \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)\right)}{\partial \boldsymbol{\mu}_i} \\ &= \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \cdot \frac{\partial \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)^{\top} \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)\right)}{\partial \boldsymbol{\mu}_i} \\ &= \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)^{\top} \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)\right) \cdot -\frac{1}{2} \frac{\partial\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)^{\top} \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)}{\partial \boldsymbol{\mu}_i} \\ &= \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)^{\top} \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)\right) \cdot \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right) \\ &= p\left(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \cdot \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)\end{aligned}$$

高斯混合聚类 – 模型求解（续）

令：

$$\frac{\partial LL(D)}{\partial \Sigma_i} = 0 \quad \Longrightarrow \quad \Sigma_i = \frac{\sum_{j=1}^{m} \gamma_{ji}(x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{m} \gamma_{ji}}$$

高斯混合聚类 – 模型求解（续）

$$\frac{\partial LL(D)}{\partial \boldsymbol{\Sigma}_i} = \frac{\partial}{\partial \boldsymbol{\Sigma}_i}\left[\sum_{j=1}^{m} \ln\left(\sum_{i=1}^{k} \alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right)\right]$$

$$= \sum_{j=1}^{m} \frac{\partial}{\partial \boldsymbol{\Sigma}_i}\left[\ln\left(\sum_{i=1}^{k} \alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right)\right]$$

$$= \sum_{j=1}^{m} \frac{\alpha_i \cdot \boxed{\frac{\partial}{\partial \boldsymbol{\Sigma}_i}(p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i))}}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

$$= \frac{\partial}{\partial \boldsymbol{\Sigma}_i}\left[\frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right)\right]$$

$$= \frac{\partial}{\partial \boldsymbol{\Sigma}_i}\left\{\exp\left[\ln\left(\frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right)\right)\right]\right\}$$

$$= p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\cdot\frac{\partial}{\partial \boldsymbol{\Sigma}_i}\left[\ln\left(\frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right)\right)\right]$$

$$= p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\cdot\frac{\partial}{\partial \boldsymbol{\Sigma}_i}\left[\ln\frac{1}{(2\pi)^{\frac{n}{2}}} - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right]$$

$$= p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\cdot\left[-\frac{1}{2}\frac{\partial(\ln|\boldsymbol{\Sigma}_i|)}{\partial \boldsymbol{\Sigma}_i} - \frac{1}{2}\frac{\partial\left[(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right]}{\partial \boldsymbol{\Sigma}_i}\right]$$

$$\frac{\partial|\mathbf{X}|}{\partial \mathbf{X}} = |\mathbf{X}|\cdot(\mathbf{X}^{-1})^T, \quad \frac{\partial \boldsymbol{a}^T\mathbf{X}^{-1}\mathbf{B}}{\partial \mathbf{X}} = -\mathbf{X}^{-T}\boldsymbol{a}\boldsymbol{b}^T\mathbf{X}^{-T}$$

$$\Longrightarrow \quad \frac{\partial}{\partial \boldsymbol{\Sigma}_i}(p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)) = p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\cdot\left[-\frac{1}{2}\boldsymbol{\Sigma}_i^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}\right]$$

$$\frac{\partial LL(D)}{\partial \boldsymbol{\Sigma}_i} = \sum_{j=1}^{m}\frac{\alpha_i\cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k}\alpha_l\cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}\cdot\left[-\frac{1}{2}\boldsymbol{\Sigma}_i^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}\right]$$

高斯混合聚类 - 模型求解（续）

$$\frac{\partial LL(D)}{\partial \boldsymbol{\Sigma}_i} = \sum_{j=1}^{m} \gamma_{ji} \cdot \left[-\frac{1}{2}\boldsymbol{\Sigma}_i^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} \right] = 0$$

$$\sum_{j=1}^{m} \gamma_{ji} \cdot \left[-\boldsymbol{I} + (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} \right] = 0$$

$$\sum_{j=1}^{m} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} = \sum_{j=1}^{m} \gamma_{ji} \boldsymbol{I}$$

$$\sum_{j=1}^{m} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T = \sum_{j=1}^{m} \gamma_{ji} \boldsymbol{\Sigma}_i$$

$$\boldsymbol{\Sigma}_i^{-1} \cdot \sum_{j=1}^{m} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T = \sum_{j=1}^{m} \gamma_{ji}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{j=1}^{m} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{m} \gamma_{ji}}$$

高斯混合聚类 – 模型求解（续）

混合系数 α_i 的确定

对公式(9.37) 两边同时乘以α_i可得

$$\sum_{j=1}^{m} \frac{\alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} + \lambda\alpha_i = 0$$

$$\sum_{j=1}^{m} \frac{\alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} = -\lambda\alpha_i$$

两边对所有混合成分求和可得

$$\sum_{i=1}^{k}\sum_{j=1}^{m} \frac{\alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} = -\lambda\sum_{i=1}^{k}\alpha_i$$

$$\sum_{j=1}^{m}\sum_{i=1}^{k} \frac{\alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} = -\lambda\sum_{i=1}^{k}\alpha_i \quad \Rightarrow \quad m = -\lambda$$

$$\sum_{j=1}^{m} \frac{\alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} = -\lambda\alpha_i = m\alpha_i \qquad \alpha_i = \frac{1}{m}\sum_{j=1}^{m} \frac{\alpha_i \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \qquad \alpha_i = \frac{1}{m}\sum_{j=1}^{m}\gamma_{ji}$$

高斯混合聚类

- 算法伪代码：

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

1 E步

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：$\alpha'_i = \frac{\sum_{j=1}^m \gamma_{ji}}{m}$;

2 M步

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：$C_{\lambda_j} = C_{\lambda_j} \bigcup \{\boldsymbol{x}_j\}$

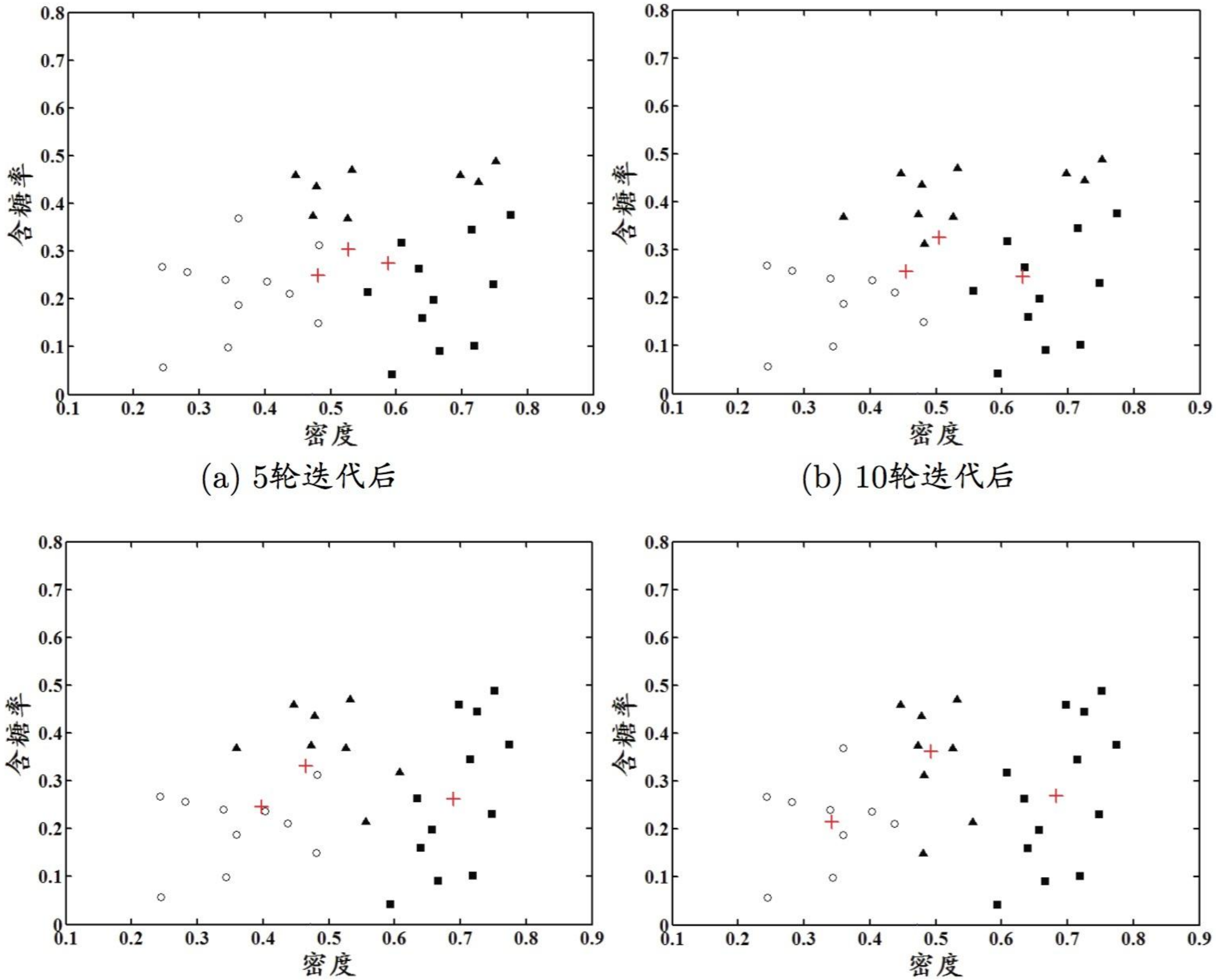
end for

return 簇划分结果

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

高斯混合聚类

- 聚类效果：



(a) 5轮迭代后 (b) 10轮迭代后

(c) 20轮迭代后 (d) 50轮迭代后

大纲

- 聚类任务
- 性能度量
- 距离计算
- 原型聚类
- 密度聚类
- 层次聚类

密度聚类

- 密度聚类的定义

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。

DBSCAN（ Density-Based Spatial Clustering of Applications with Noise ）是一种代表性的密度聚类算法。

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：
  - ϵ **邻域**：对样本 $x_j \in D$，其 ϵ 邻域包含样本集 D 中与 x_j 的距离不大于 ϵ 的样本；
  - **核心对象**：若样本 x_j 的 ϵ 邻域至少包含**MinPts**个样本，则该样本点为一个核心对象；
  - **密度直达**：若样本 x_j 位于样本 x_i 的 ϵ 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 $p_1, p_2, \cdots, p_n$，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

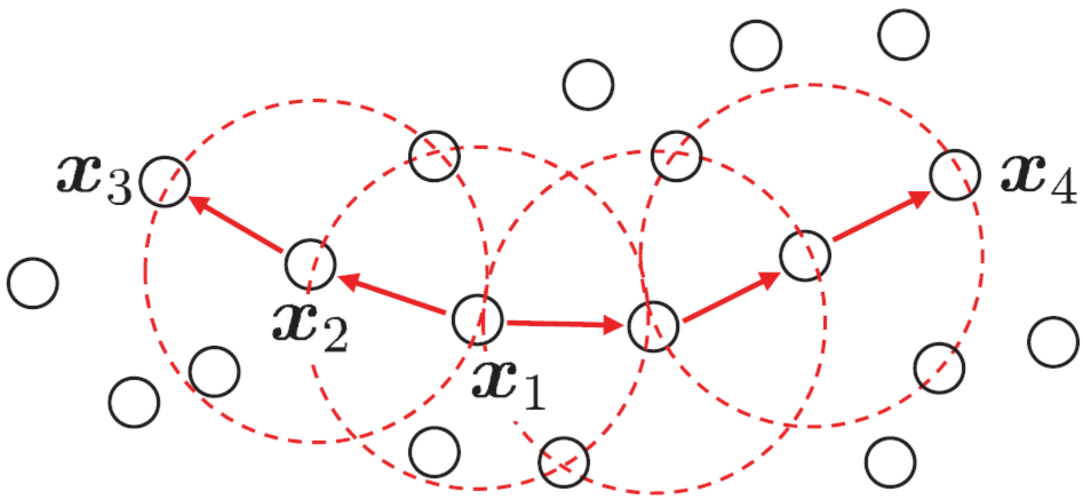
  虚线显示出ϵ邻域。

  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。



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$X = \{x' \in D \mid x'$由x密度可达$\}$，则X为满足连接性与最大性的簇。

密度聚类

- DBSCAN算法伪代码：

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

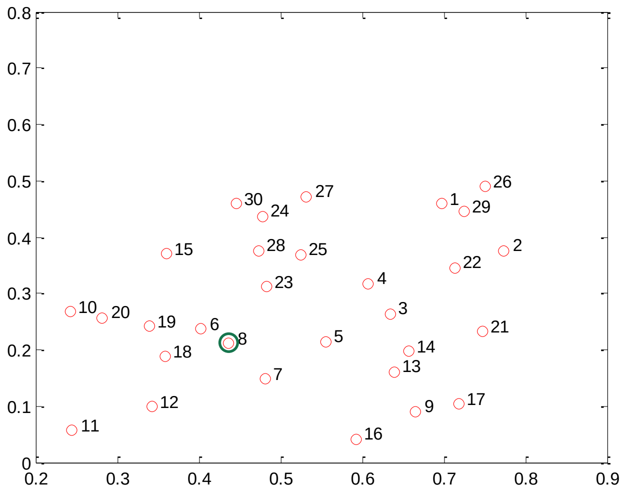
输出： 簇划分$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

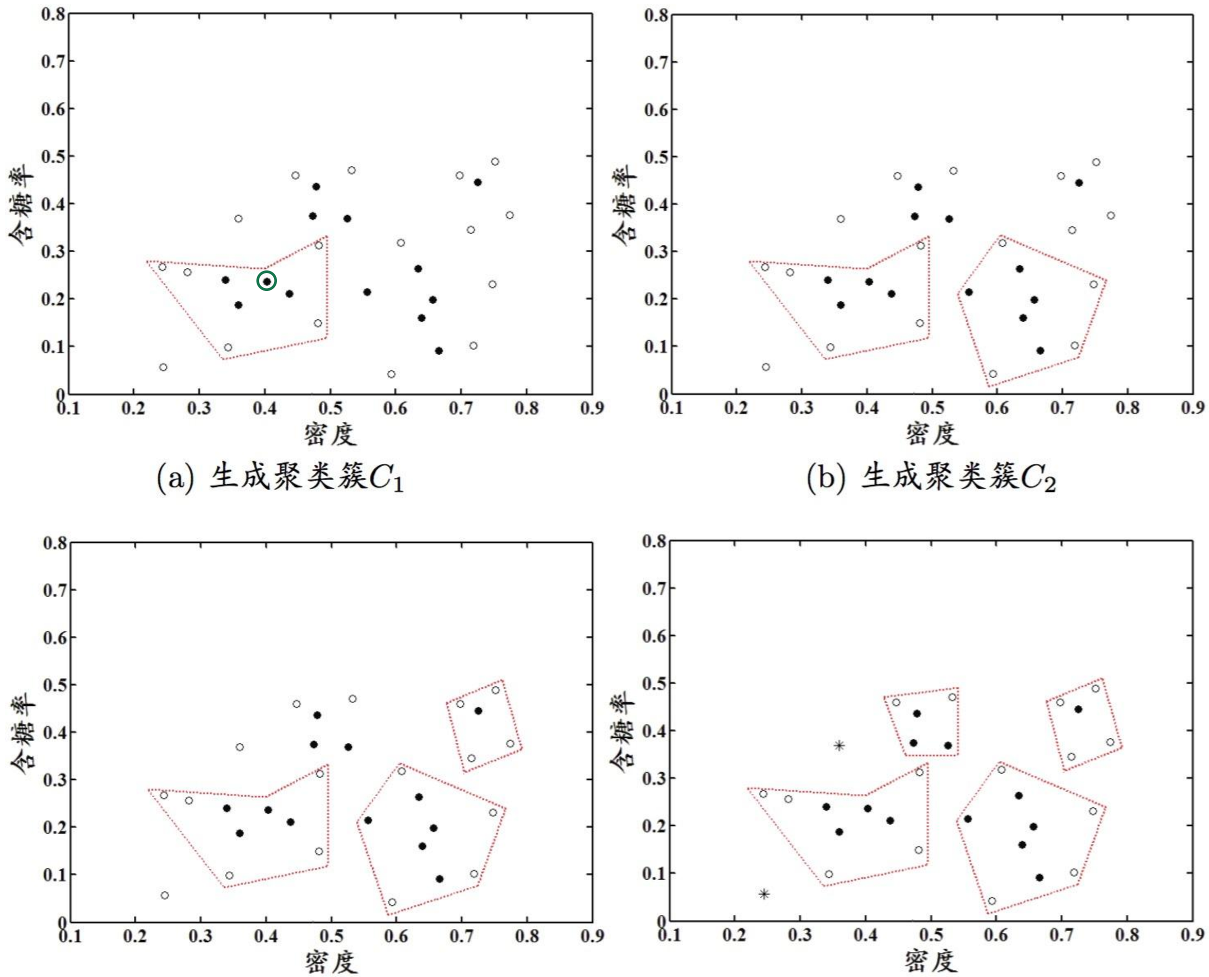
确定核心对象集合

生成某个核心对象的聚类簇

密度聚类

- 聚类效果：





(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

大纲

- 聚类任务
- 性能度量
- 距离计算
- 原型聚类
- 密度聚类
- 层次聚类

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。
- AGNES(AGglomerative NESting)算法（自底向上的层次聚类算法）

首先，将样本中的每一个样本看做一个初始聚类簇，然后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，该过程不断重复，直到达到预设的聚类簇的个数。

这里两个聚类簇 C_i 和 C_j 的距离，可以有3种度量方式。

层次聚类

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

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

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

层次聚类 – AGNES算法

- AGNES算法伪代码：

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$d \in \{d_{\min}, d_{\max}, d_{\mathrm{avg}}\}$;
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

1

初始化簇和距离矩阵

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

2

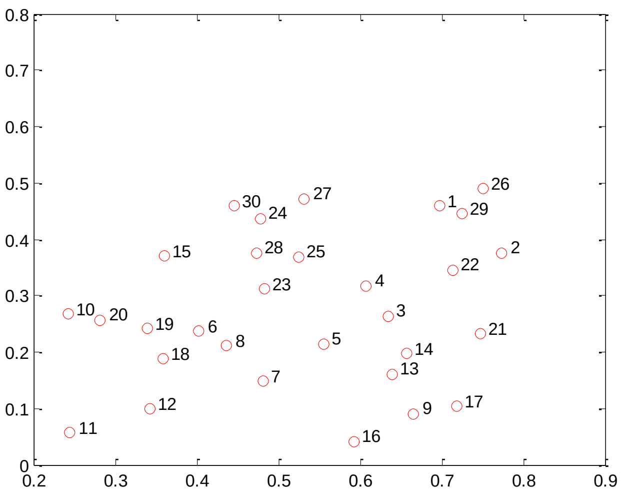
合并簇，更新距离矩阵

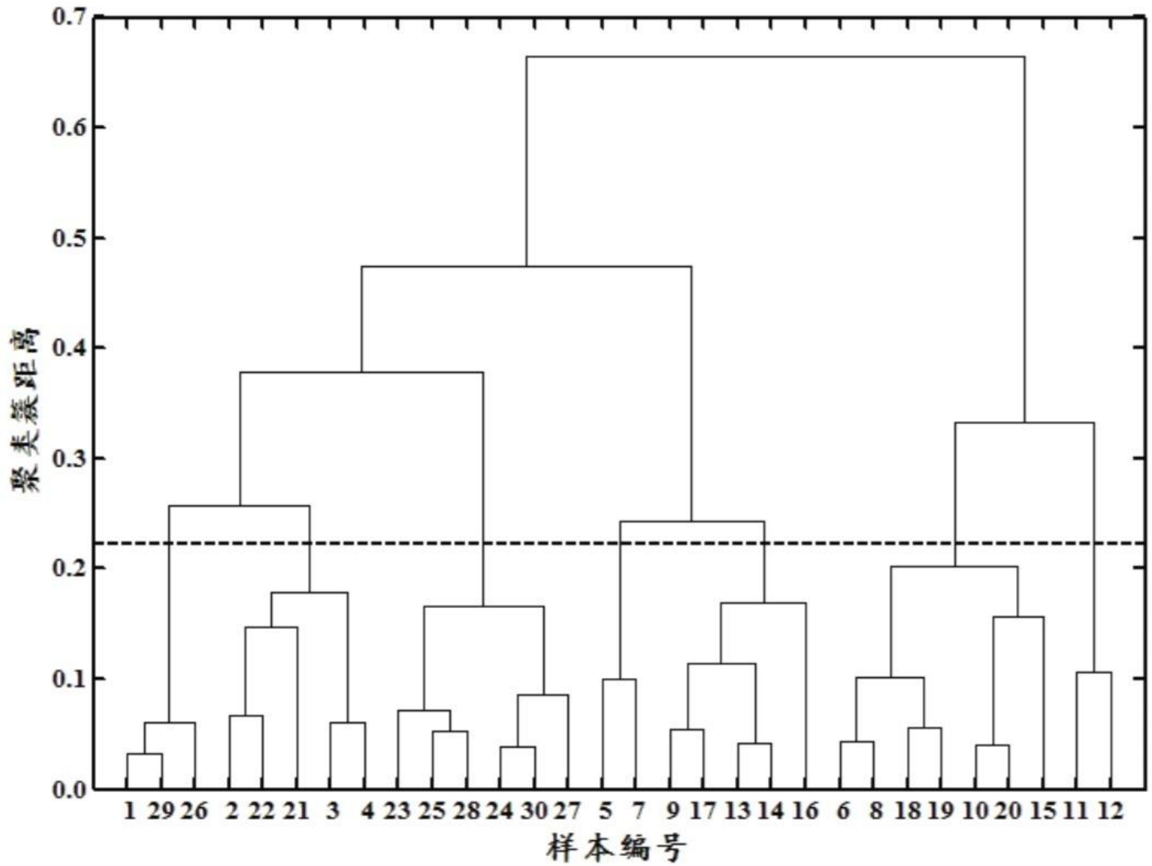
return 簇划分结果

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

层次聚类 – 树状图

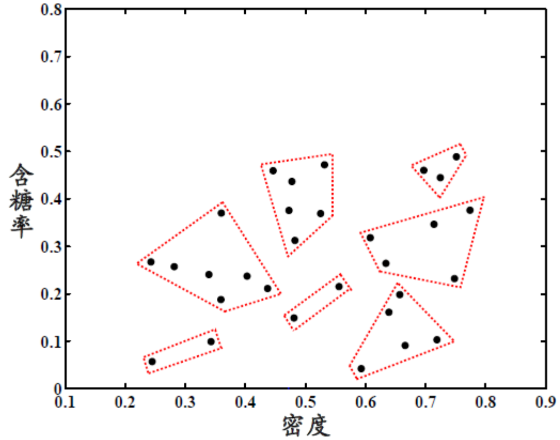
- AGNES算法树状图：



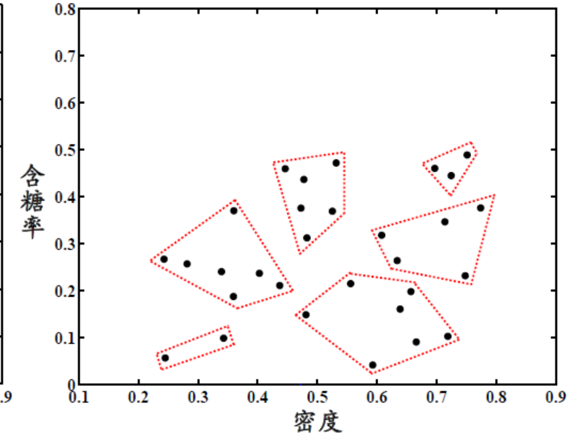


层次聚类

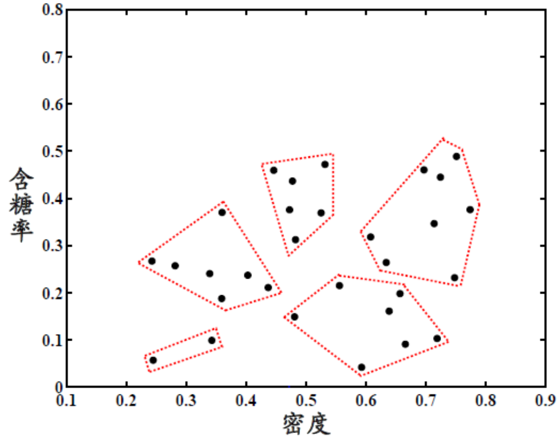
- AGNES算法聚类效果：



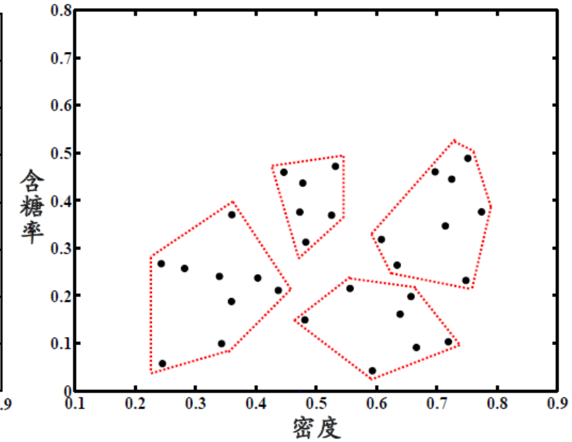
(a) 聚类簇数 $k = 7$



(b) 聚类簇数 $k = 6$



(c) 聚类簇数 $k = 5$



(d) 聚类簇数 $k = 4$

$$X = \begin{bmatrix} 0 & 0 & 1 & 5 & 5 \\ 2 & 0 & 0 & 0 & 2 \end{bmatrix}$$

（1）选择两个样本点作为类的中心。假设选择$m_1^{(0)}=x_1=(0,2)^T$，$m_2^{(0)}= x_2=(0,0)^T$ 。

（2）以$m_1^{(0)}$，$m_2^{(0)}$为类$G_1^{(0)}$，$G_2^{(0)}$的中心，计算$x_3=(1,0)^T$ ，$x_4=(5,0)^T$ ，$x_5=(5,2)^T$ 与$m_1^{(0)}= (0,2)^T$，$m_2^{(0)}= (0,0)^T$的欧氏距离平方。

对$x_3=(1,0)^T$，$d(x_3, m_1^{(0)})=5$，$d(x_3, m_2^{(0)})= 1$，将x_3分到类$G_2^{(0)}$。

对$x_4=(5,0)^T$，$d(x_4, m_1^{(0)})=29$，$d(x_4, m_2^{(0)})=25$，将x_4分到类$G_2^{(0)}$ 。

对$x_5=(5,2)^T$，$d(x_5, m_1^{(0)})=25$，$d(x_5, m_2^{(0)})=29$，将x_5分到类$G_1^{(0)}$。

（3）得到新的聚类$G_1^{(1)}=\{x_1, x_5\}$，$G_2^{(1)}=\{x_2, x_3, x_4\}$，计算类的中心$m_1^{(1)}$，$m_2^{(1)}$：

$$m_1^{(1)}= (2.5, 2)^T, \quad m_2^{(1)}= (2, 0)^T$$

（4）重复步骤（2）和步骤（3）。将x_1分到类$G_1^{(1)}$，将x_2分到类$G_2^{(1)}$，x_3分到类$G_2^{(1)}$，x_4分到类$G_2^{(1)}$，x_5分到类$G_1^{(1)}$。得到新的聚类$G_1^{(2)}=\{x_1, x_5\}$，$G_2^{(2)}=\{x_2, x_3, x_4\}$ 。

由于得到的新的类没有改变，聚类停止。得到聚类结果：$G_1^*=\{x_1, x_5\}$，$G_2^*=\{x_2, x_3, x_4\}$