统计机器学习
Statistical Machine Learning

魏莱

上海海事大学信息工程学院

2020 年 5 月 9 日

第九章：聚类

1. 聚类的概念

我们已经学习过机器学习的两大类任务：分类和回归。这两类任务都属于有监督学习（supervised learning）。机器学习还有第三类任务，称之为**聚类（clustering）**。聚类属于无监督学习（unsupervised learning）

聚类的含义是指：通过聚类，数据集中的样本可以被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)。并且，同一个“簇”中样本之间的相似性程度要大于不同“簇”之间样本的相似性程度。

注意，划分后每一个“簇”实际上对应了一个潜在的“类别”，但这个“类别”在聚类之前是不知道的，因此聚类是一种无监督学习。

形式化的，假定样本集 $\mathbf{D}=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n\}$，样本 $\mathbf{x}_i\in\mathbb{R}^D$。聚类算法可以将样本集 $\mathbf{D}$ 划分成 k 个不相交的簇 $\{\mathbf{C}_j|j=1,2,\cdots,k\}$，其中 $\mathbf{C}_j\bigcap\mathbf{C}_h=\emptyset,j\neq h$ 并且 $\mathbf{D}=\bigcup_{j=1}^{k}\mathbf{C}_j$。另外，可以用 l_i 表示样本 $\mathbf{x}_i$ 的“簇标记”。聚类算法运行结束，有 $\mathbf{x}_i\in\mathbf{C}_{l_i}$。

1. 聚类的应用

聚类既能作为一个单独过程，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。其主要应用有：

1. 用户画像: 在用户使用移动网络时，会自然的留下用户的位置信息。如百度与万达进行合作，通过定位用户的位置，结合万达的商户信息，向用户推送位置营销服务，提升商户效益。
2. 图像分割: 图像分割就是把图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标。图像分割广泛应用于医学、交通、军事等领域。

2. 性能度量

基于不同的学习策略，人们设计出多种类型的聚类算法。但这些聚类算法都涉及两个基本问题——性能度量和距离计算。

聚类性能度量亦称聚类“**有效性指标**”**(validity index)**，其用来评估聚类结果的好坏。实际上，若明确了最终将要使用的性能度量，则可直接将其作为聚类过程的优化目标，从而更好地得副符合要求的聚类结果。

2. 性能度量

前文我们提到，通过聚类同一个“簇”中样本之间的相似性程度要大于不同“簇”之间样本的相似性程度。同一个“簇”中样本之间的相似性程度称为**“簇内相似度”(intra-cluster similarity)**，不同“簇”之间样本的相似性程度称为**“簇间相似度”(inter-cluster similarity)**。因此，一个好的聚类算法得到的聚类结果应该满足簇内相似度高，而簇间相似度低。

2. 性能度量

具体的，假定数据集 $\mathbf{D}$ 通过聚类给出划分 $\mathcal{C} = \{\mathbf{C}_1, \cdots, \mathbf{C}_k\}$，而参考模型 (reference model)（大部分情况下是准确标记）给出的划分为 $\mathcal{C}^* = \{\mathbf{C}_1^*, \cdots, \mathbf{C}_k^*\}$，令 $\{l_1, \cdots, l_n\}$ 和 $\{l_1^*, \cdots, l_n^*\}$ 分别为对应的簇标记，则可以得到下面四个集合：

$$
\begin{aligned}
a &= |\mathbf{SS}| \quad \mathbf{SS} = \{(\mathbf{x}_i, \mathbf{x}_j) | l_i = l_j, l_i^* = l_j^*, i < j\} \\
b &= |\mathbf{SD}| \quad \mathbf{SD} = \{(\mathbf{x}_i, \mathbf{x}_j) | l_i = l_j, l_i^* \neq l_j^*, i < j\} \\
c &= |\mathbf{DS}| \quad \mathbf{DS} = \{(\mathbf{x}_i, \mathbf{x}_j) | l_i \neq l_j, l_i^* = l_j^*, i < j\} \\
d &= |\mathbf{DD}| \quad \mathbf{DD} = \{(\mathbf{x}_i, \mathbf{x}_j) | l_i \neq l_j, l_i^* \neq l_j^*, i < j\}
\end{aligned}
$$

很显然，$a + b + c + d = n(n-1)/2$。

2. 性能度量

根据上面得到的四个集合，可以定义聚类性能度量的指标，包括：

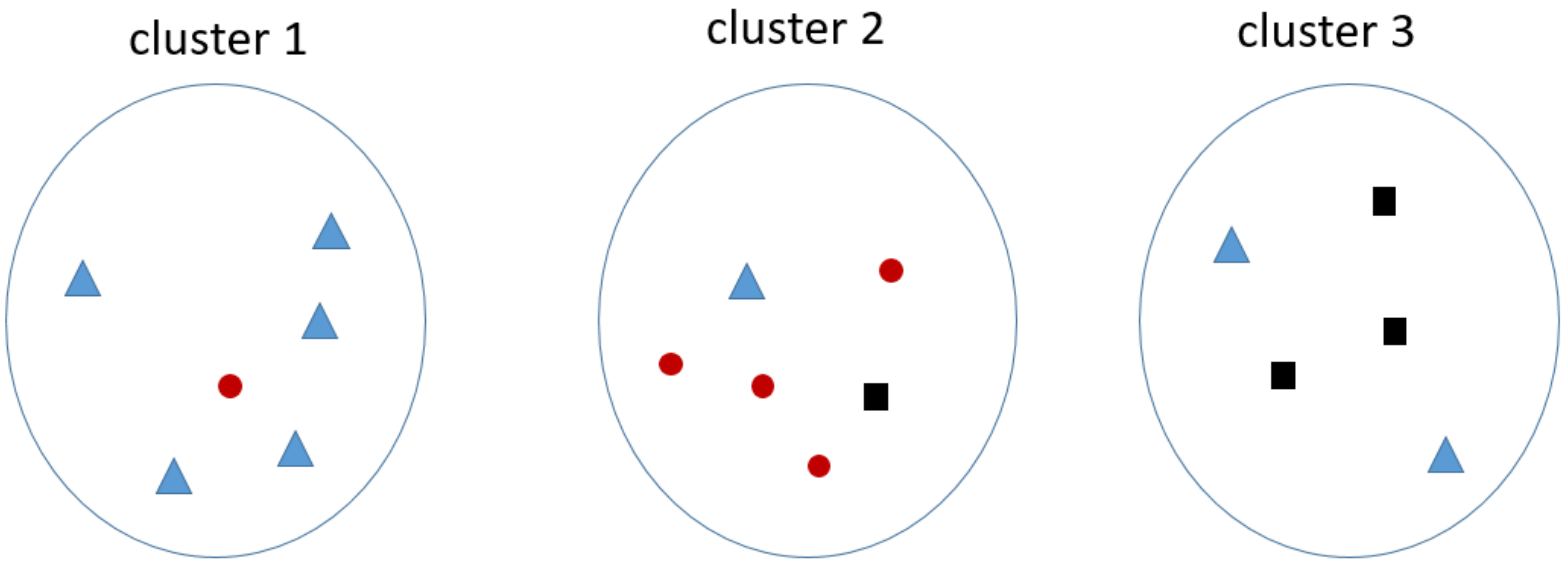
1. Jaccard 系数：$JC = \frac{a}{a+b+c}$
2. Rand 系数：$RI = \frac{2(a+d)}{n(n-1)}$

   上述性能度量的结果值均在 $[0,1]$ 区间，值越大越好。
3. 标准互信息 NMI: $NMI = \frac{I(\mathcal{C};\mathcal{C}^*)}{H(\mathcal{C})+H(\mathcal{C}^*)}$，其中

   $I(\mathcal{C};\mathcal{C}^*) = \sum_{l\in\mathcal{C}} \sum_{l^*\in\mathcal{C}^*} P(l,l^*) \log(\frac{P(l,l^*)}{P(l)P(l^*)})$,

   $H(\mathcal{C}) = -\sum_{j=1}^{k} P(l_j) \log P(l_j)$

2. 性能度量



假设上图中，样本准确的类别结果是其标注的形状，因此为 $\mathcal{C}^* = \{\{1,2,1,1,1,1\},\{1,2,2,2,2,3\},\{1,1,3,3,3\}\}$，假设某聚类算法得到的结果是三个圈标出的簇，则 $\mathcal{C} = \{\{1,1,1,1,1,1\},\{2,2,2,2,2,2\},\{3,3,3,3,3\}\}$。

联合概率分布：$P(1,1)=5/17, P(1,2)=1/17, P(1,3)=0; P(2,1)=1/17, P(2,2)=4/17, P(2,3)=1/17; P(3,1)=2/17, P(3,2)=0, P(3,3)=3/17;$ 从而可以计算出互信息。

边缘分布：$P(\mathcal{C}): P(1)=6/17, P(2)=6/17, P(3)=5/17$，

$P(\mathcal{C}^*): P(1)=8/17, P(2)=5/17, P(3)=4/17$，

于是信息熵：

$H(\mathcal{C})=-6/17\log 1/17-6/17\log 6/17-5/17\log 5/17$，

$H(\mathcal{C}^*)=-8/17\log 8/17-5/17\log 5/17-4/17\log 4/17$。

2. 性能度量

上述的指标是通过将聚类结果与某个参考模型进行比较得到的，因此称为“外部指标”(external index)。另一类指标是直接考察聚类结果而不利用任何参考模型，称为“内部指标”(internal index)，包括：

1. DB 指数（DBI）：$DBI = \frac{1}{k}\sum_{i=1}^{k} \max\left(\frac{avg(\mathbf{C}_i)+avg(\mathbf{C}_j)}{d_{cen}(\mu_i,\mu_j)}\right)$，其中 $avg(\mathbf{C}_i)$ 对应于簇 $\mathbf{C}_i$ 内样本的平均距离，$d_{cen}(\mu_i,\mu_j)$ 对应于簇 $\mathbf{C}_i$ 与 $\mathbf{C}_j$ 中心点的距离。
2. Dunn 指数（DI）：
   $DI = \min_{1\le i\le k}\left\{\min_{i\neq j}\left(\frac{d_{min}(\mathbf{C}_i,\mathbf{C}_j)}{\max_{1\le l\le k} diam(\mathbf{C}_l)}\right)\right\}$，其中 $d_{min}(\mathbf{C}_i,\mathbf{C}_j)$ 对应于簇 $\mathbf{C}_i$ 与 $\mathbf{C}_j$ 最近样本的距离，$diam(\mathbf{C}_l)$ 对应于簇 $\mathbf{C}_l$ 内样本间的最远距离。

显然，DBI 的值越小越好，而 DI 则相反，值越大越好。

3. 距离计算

可以发现，在计算聚类性能度量的内部指标时，需要计算簇样本、簇中心的距离，而距离对于很多机器学习算法来说都是非常重要的概念。广义的距离可以有多种定义，在某种程度上还可以表示样本之间的相似程度。但任何距离都应该满足一些基本性质：

1. 非负性：$dist(\mathbf{x}_i, \mathbf{x}_j) \geq 0$;
2. 同一性：$dist(\mathbf{x}_i, \mathbf{x_j}) = 0$, i.i.f. $\mathbf{x}_i = \mathbf{x}_j$;
3. 对称性：$dist(\mathbf{x}_i, \mathbf{x}_j) = dist(\mathbf{x}_j, \mathbf{x}_i)$;
4. 直递性 (三角不等式)：
   $dist(\mathbf{x}_i, \mathbf{x}_j) \leq dist(\mathbf{x}_i, \mathbf{x}_k) + dist(\mathbf{x}_k, \mathbf{x}_j)$

3. 距离计算

假设样本 $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$，最常用的距离为**闵可夫斯基距离 (Minkowski distance)**：

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{D} \|x_{ik} - x_{jk}\|^p \right)^{\frac{1}{p}} \tag{1}$$

当 $p \geq 1$，上式距离定义满足距离的四个基本性质。

1. 当 $p = 2$，等式（1）称为欧氏距离，
   $dist(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$；
2. 当 $p = 1$，等式（1）称为曼哈顿距离，
   $dist(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{D} \|x_{ik} - x_{jk}\|$；

注意：闵可夫斯基距离适用于属性值可以比较的有序属性（ordinal attribute）。

对于无序属性，样本之间的距离需要依赖于**值差异度量，VDM (Value Difference Metric)**。属性 F 上，两个值 a, b 之间的 VDM 定义如下：

$$VDM(a,b) = \sum_{i=1}^{k} \left| \frac{n_{F,a,i}}{n_{F,a}} - \frac{n_{F,b,i}}{n_{F,b}} \right|^p \tag{2}$$

其中 $n_{F,a}$ 表示属性 F 上取值为 a 的样本数量，$n_{F,a,i}$ 表示为 F 上取值为 a 的在第 i 簇中的数量。

于是，假定样本都具有 D_c 个有序属性，$D - D_c$ 个无序属性，则可以定义两个样本之间的距离：

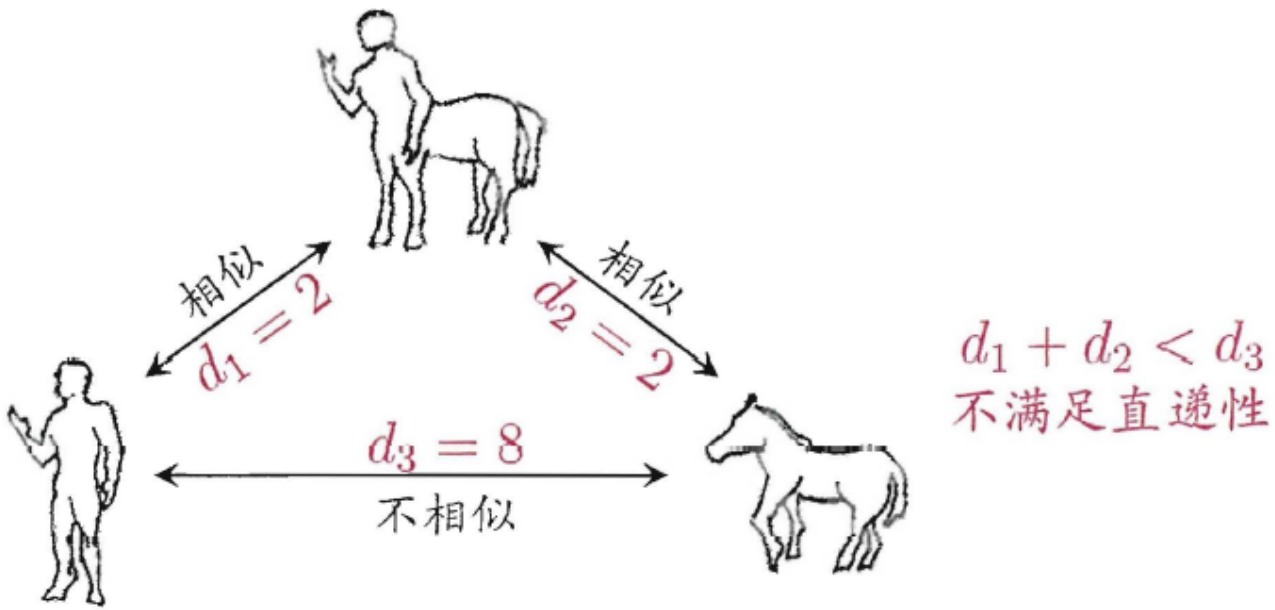
$$dist(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{D_c} |x_{ik} - x_{jk}|^p + \sum_{k=D_c+1}^{D} VDM(x_{ik} - x_{jk}) \right)^{\frac{1}{p}} \quad (3)$$

当样本空间中不同属性的重要性不同时，可使用**加权距离 (weighted distance)**。以加权闵可夫斯基距离定义为：

$$dist_w(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{D} w_k \|x_{ik} - x_{jk}\|^p \right)^{\frac{1}{p}} \quad (4)$$

3. 距离计算

上文我们说，距离可以视为某种相似度度量，因为相似性越大，距离应该越小，而相似性越小，距离应该越大。但反过来，相似度度量并不全是距离，即相似度度量不一定满足距离度量的所有基本性质。



4. 聚类方法：K 均值聚类 (K-means)

给定样本集 $\mathbf{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$，K 均值聚类所得的划分 $\mathcal{C} = \{\mathbf{C}_1, \cdots, \mathbf{C}_k\}$，能够最小化平方误差

$$E = \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathbf{C}_i} \|\mathbf{x} - \mu_i\|_2^2, \tag{5}$$

其中 $\mu_i = \frac{1}{|\mathbf{C}_i|} \sum_{\mathbf{x} \in \mathbf{C}_i} \mathbf{x}$ 是簇 $\mathbf{C}_i$ 的均值。按照前文聚类指标来看，上式实际上定义了簇内距离（簇内样本的紧密程度），希望簇内距离越小越好。

4. 聚类方法：K 均值聚类 (K-means)

要根据 (5) 来对样本集进行划分，这是一个 **NP 难问题（多项式非确定问题，即算法时间随样本规模指数级增长）**。因此，K 均值算法采用了贪心策略，通过迭代优化来近似求解式。

4. 聚类方法：K 均值聚类 (K-means)

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 $C_i = \varnothing \ (1 \leqslant i \leqslant k)$
  for $j = 1, 2, \ldots, m$ **do**
    计算样本 $\boldsymbol{x}_j$ 与各均值向量 $\boldsymbol{\mu}_i \ (1 \leqslant i \leqslant k)$ 的距离: $d_{ji} = \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|_2$;
    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 $i = 1, 2, \ldots, k$ **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
until 当前均值向量均未更新

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

4. 聚类方法：K 均值聚类 (K-means)

以下图为例来演示 K 均值算法的学习过程。

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

4. 聚类方法：K 均值聚类 (K-means)

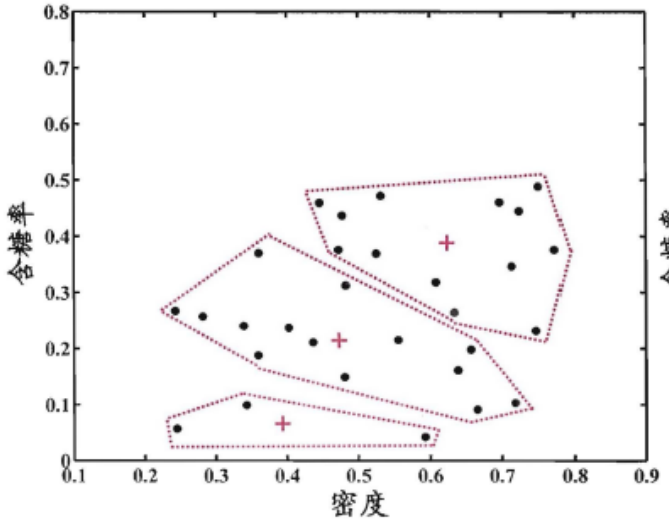
假定簇个数 K=3，

STEP 1. 随机选择 K 个样本作为三个簇的簇中心，$\mu_1 = \mathbf{x}_6 = (0.403, 0.237)^t, \mu_2 = \mathbf{x}_{12} = (0.343, 0.099)^t, \mu_3 = \mathbf{x}_{27} = (0.532, 0.472)^t$；

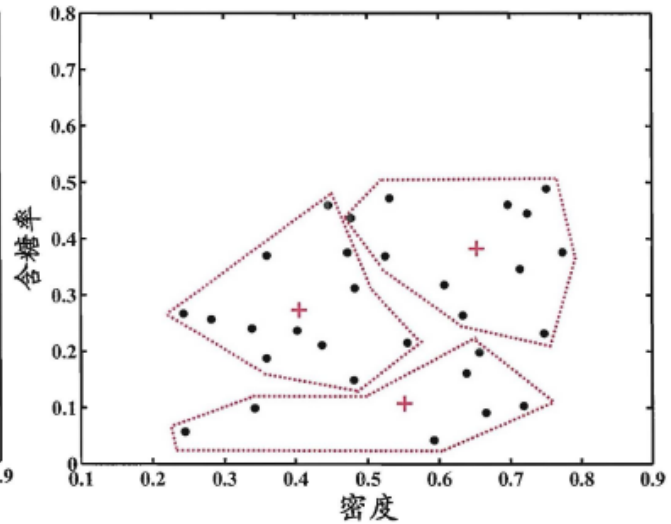
STEP 2. 计算所有样本到 K 个簇中心的距离（欧氏距离），然后根据每一样本到 K 个簇中心距离的大小，将其分配到相应的簇中，得到划分 $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$；

STEP 3. 根据划分 $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$，重新计算得到 K 个簇中心 $\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3$，检查新的簇中心与原簇中心的距离，如果距离接近，则跳出循环，否则跳转到 STEP2；

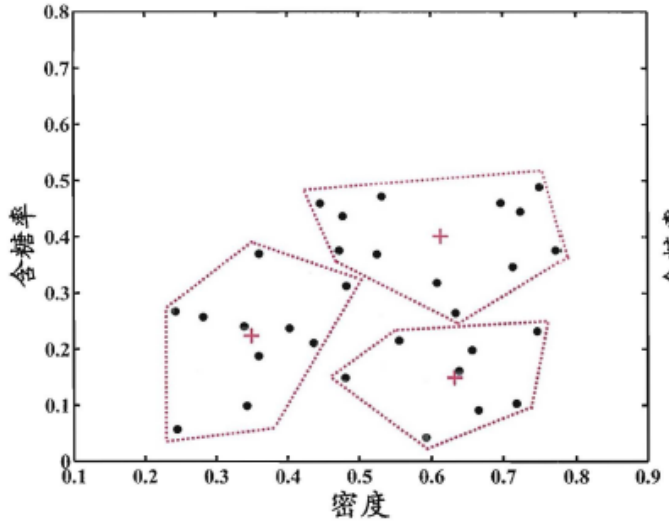
4. 聚类方法：K 均值聚类 (K-means)



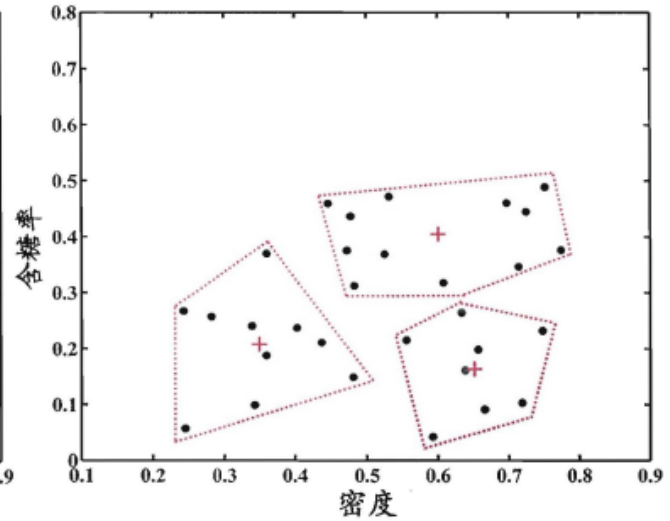
(a) 第一轮迭代后



(b) 第二轮迭代后



(c) 第三轮迭代后



(d) 第四轮迭代后

4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类

回顾 (多元) 高斯分布的定义。对于 D 维样本空间中的随机变量 $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$，其概率密度函数可以写成：

$$p(\mathbf{x}|\mu, \Sigma) = \frac{1}{(\sqrt{2\pi})^n \sqrt{|\Sigma|}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\mu)^t \Sigma^{-1}(\mathbf{x}-\mu)\Big), \quad (6)$$

其中 μ 是均值，$\Sigma \in \mathbb{R}^{D\times D}$ 是协方差矩阵。

对于数据集 $\mathbf{D}$ 来说，我们可以假设样本是由 K 个满足高斯分布的簇组合而成，因此对于 $\mathbf{x} \in \mathbf{D}$，其概率密度函数可以表示为：

$$p(\mathbf{x}) = \sum_{i=1}^{K} w_i p(\mathbf{x}|\mu_i, \Sigma_i) \quad (7)$$

其中 w_i 可以理解成样本 $\mathbf{x}$ 隶属于第 i 个簇的程度（可能性），很明显 $\sum_{i=1}^{K} w_i = 1$。

则样本 $\mathbf{x}_i$ 属于 j 簇的概率可以表示为：

$$
\begin{aligned}
p(l_i = j|\mathbf{x}_i) &= \frac{p(l_i{=}j)p(\mathbf{x}_i|l_i{=}j)}{p(\mathbf{x}_i)} = \frac{p(l_i{=}j)p(\mathbf{x}_i|l_i{=}j)}{\sum_{v=1}^{K} w_v p(\mathbf{x}_i|\mu_v,\Sigma_v)} \\
&= \frac{w_j p(\mathbf{x}_i|\mu_j,\Sigma_j)}{\sum_{v=1}^{K} w_v p(\mathbf{x}_i|\mu_v,\Sigma_v)}
\end{aligned} \tag{8}
$$

上式代表样本 $\mathbf{x}_i$ 属于 j 簇的后验概率。$p(\mathbf{x}_i|\mu_j,\Sigma_j)$ 代表样本 $\mathbf{x}_i$ 由 j 簇生成的概率。上式中未知变量包括：

$w_v, \mu_v, \Sigma_v, v = 1, \cdots, K$。

4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类

在前一章贝叶斯分类器中，我们讲到，最优分类器是依据后验概率最大来进行分类的分类器，因此一旦所有参数确定，可以根据后验概率来进行簇划分。在分类任务中，我们希望最大化边缘概率，但对于聚类任务来说，样本属于哪一簇是未知的，于是我们考虑如下最大化似然问题:

$$\begin{aligned} \max \prod_{i=1}^{n} p(\mathbf{x}_i) \quad & \Leftrightarrow \max \ln \prod_{i=1}^{n} p(\mathbf{x}_i) \\ & = \max \sum_{i=1}^{n} \ln p(\mathbf{x}_i) \\ & = \max \sum_{i=1}^{n} \ln \sum_{j=1}^{K} w_j p(\mathbf{x}_i | \mu_j, \Sigma_j) = J \end{aligned} \tag{9}$$

4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类

借助 EM 算法来交替优化 w_j, μ_j, Σ_j。

1. 固定其他变量，更新 μ_j。令 $\frac{\partial J}{\partial \mu_j} = 0$，则

$$\sum_{i=1}^{n} \frac{w_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\sum_{v=1}^{K} w_v p(\mathbf{x}_i|\mu_v, \Sigma_v)} (\mathbf{x}_i - \mu_j) = 0 \tag{10}$$

令 $\gamma_{ij} = p(l_i = j|\mathbf{x}_i)$，则可以得到

$$\mu_j = \frac{\sum_{i=1}^{n} \gamma_{ij} \mathbf{x}_i}{\sum_{i=1}^{n} \gamma_{ij}}, \tag{11}$$

即各混合成分的均值可通过样本加权平均来估计，样本权重是每个样本属于该成分的后验概率。

2. 固定其他变量，更新 Σ_j。令 $\frac{\partial J}{\partial \Sigma_j} = 0$，则可以得到

$$\Sigma_i = \frac{\sum_{i=1}^{n} \gamma_{ij}(\mathbf{x}_i - \mu_i)(\mathbf{x}_i - \mu_i)^t}{\sum_{i=1}^{n} \gamma_{ij}} \tag{12}$$

3. 固定其他变量，更新 w_j。由于 w_j 满足 $\sum_{j=1}^{K} w_j = 1, w_j \geq 0$，因此通过 Lagrange 乘子法求解，即定义 Lagrange 函数：$\mathcal{L} = J + \lambda(\sum_{j=1}^{K} - 1)$，然后令 $\frac{\partial \mathcal{L}}{\partial w_j} = 0$，可得

$$\sum_{i=1}^{n} \frac{p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\sum_{v=1}^{K} w_v p(\mathbf{x}_i|\mu_v, \Sigma_v)} + \lambda = 0 \tag{13}$$

等式两边乘以 w_j，可以得到 k 个等式，所有等式相加，最后可得 $\lambda = -n$，于是

$$w_j = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij} \tag{14}$$

4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类

由上述推导即可获得高斯混合模型的 EM 算法: 在每步迭代中，先根据当前参数来计算每个样本属于每个高斯成分的后验概率 γ_{ij}(E 步)，再根据式 (11)、(12) 和 (14) 更新模型参数 w_j, μ_j, Σ_j(M 步).

4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类

仍然以下图数据为例来演示高斯混合模型聚类算法的学习过程。

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

4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类

令高斯混合成分的个数 $K = 3$。

STEP 1. 初始化 $w_1 = w_2 = w_3 = \frac{1}{3}, \mu_1 = \mathbf{x}_6, \mu_2 = \mathbf{x}_{22}, \mu_3 = \mathbf{x}_{27}$，

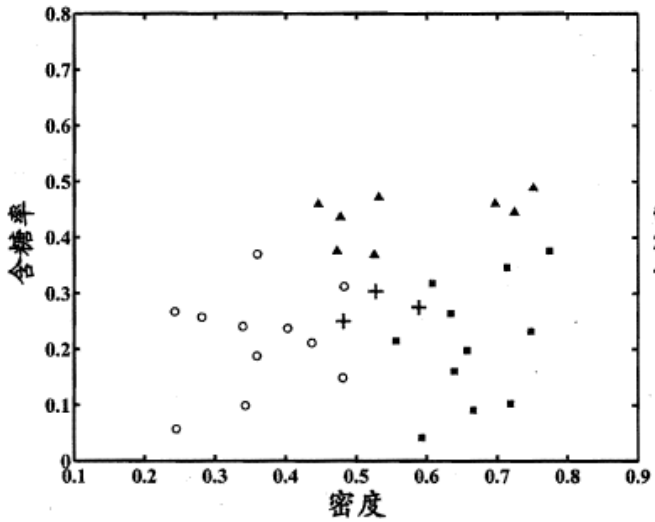
$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$$

STEP 2. 计算后验概率 $\gamma_{ij} = \frac{w_i p(\mathbf{x}_i|\mu_j,\Sigma_j)}{\sum_{v=1}^{3} w_v p(\mathbf{x}_i|\mu_v\Sigma_v)}$，$i = 1, \cdots, 30, j = 1, 2, 3$，并根据 (11),(12),(14) 式，更新 $w_1, w_2, w_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3$；

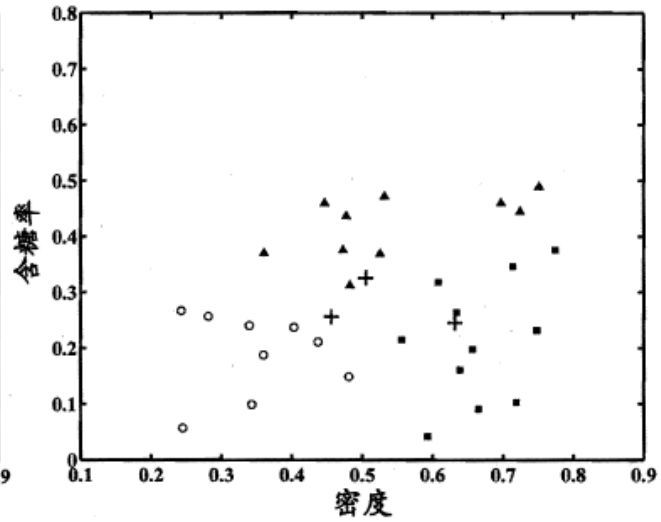
STEP 3. 对比更新值 $w_1, w_2, w_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3$ 与原有值之间的差，若小于某一固定值，则跳出循环，否则转 STEP 2。

STEP 4. 根据每一样本后验概率，进行簇划分。

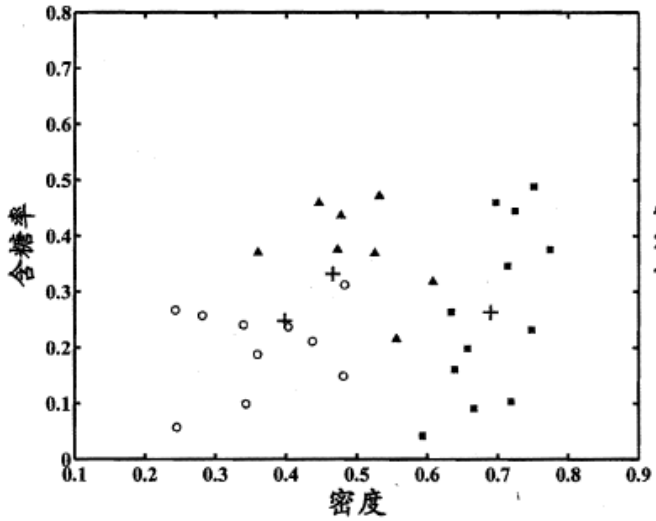
4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类



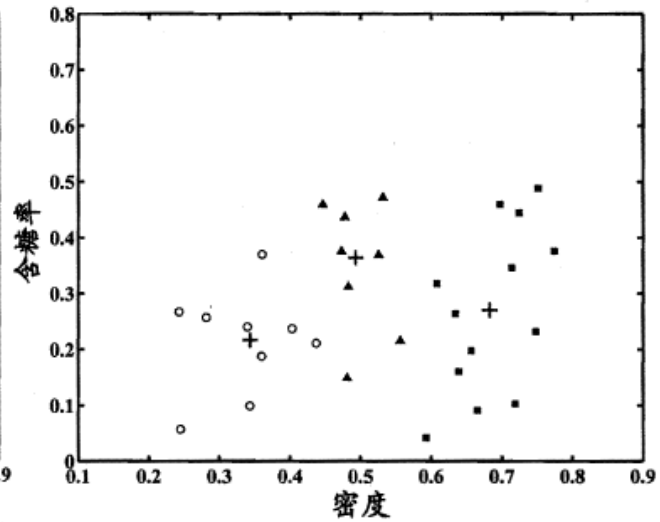
(a) 5 轮迭代后



(b) 10 轮迭代后



(c) 20 轮迭代后



(d) 50 轮迭代后

4. 聚类方法：高斯混合 (Mixture-of-Gaussian) 聚类

现在来考虑高斯混合模型聚类与 K-means 聚类之间的关系。假设高斯混合模型中每一个高斯模型的协方差矩阵 $\Sigma_j = \epsilon \mathbf{I}$，其中 ϵ 为一参数，$\mathbf{I}$ 是一个单位阵。则对于样本 $\mathbf{x}_i$ 其第 j 个高斯成分密度函数可以写成

$$p(\mathbf{x}_i|\mu_j, \Sigma_j) = \frac{1}{\sqrt{2\pi\epsilon}} \exp(-\frac{1}{2\epsilon}\|\mathbf{x}_i - \mu_j\|_2^2) \tag{15}$$

并且其相应的后验概率

$$\gamma_{ij} = \frac{w_j p(\mathbf{x}_i|\mu_j, \Sigma_j)}{\sum_{v=1}^{K} w_v p(\mathbf{x}_i|\mu_v, \Sigma_v)} = \frac{w_j \exp(-\frac{1}{2\epsilon}\|\mathbf{x}_i - \mu_j\|_2^2)}{\sum_{v=1}^{K} w_v \exp(-\frac{1}{2\epsilon}\|\mathbf{x}_i - \mu_v\|_2^2)} \tag{16}$$

令 $\epsilon \to 0$，可以发现只有 $\mathbf{x}_i$ 与 μ_j 接近的那一项趋向于 0 的速度要小于其他项。

因此当 $\epsilon \to 0$，

$$\gamma_{ij} = \begin{cases} 1, & j = l_i; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

由此，可得 $\mu_j = \frac{1}{|\mathbf{C}_j|} \sum_{\mathbf{x} \in \mathbf{C}_j} \mathbf{x}$, $w_j = \frac{|\mathbf{C}_j|}{|\mathbf{D}|}$。

再来看，由于 γ_{ij} 的性质，$p(\mathbf{x}_i) = \prod_{j=1}^{K} \left(w_j p(\mathbf{x}_i | \mu_j, \Sigma_j) \right)^{\gamma_{ij}}$，于是对所有样本求其对数似然，得到

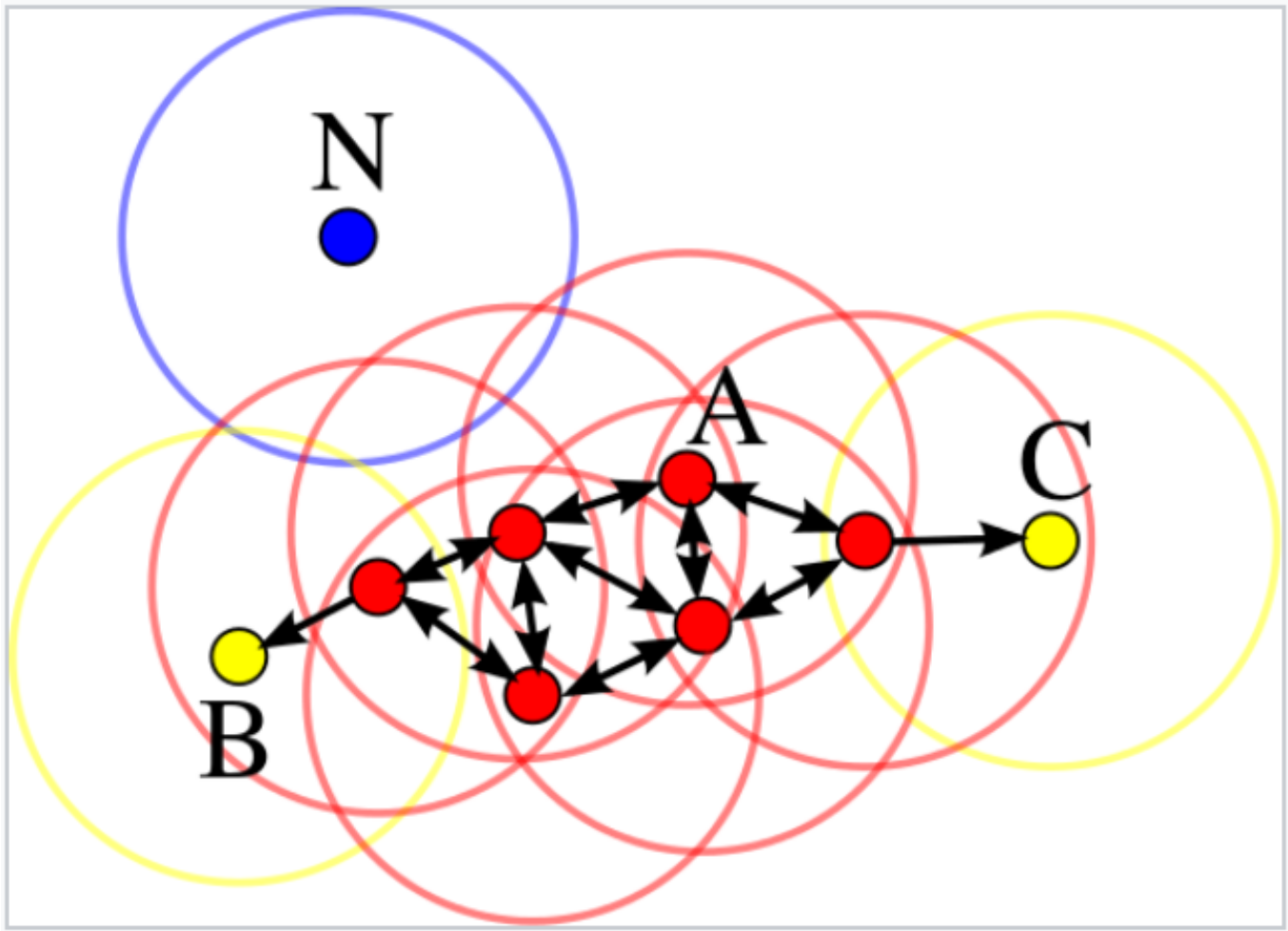
$$\begin{aligned} \max \ln \prod_{i=1}^{n} p(\mathbf{x}_i) &= \max \sum_{i=1}^{n} \sum_{j=1}^{K} \gamma_{ij} (\ln w_j + \ln p(\mathbf{x}_i | \mu_j, \Sigma_j)) \\ &= \min \sum_{i=1}^{n} \sum_{j=1}^{K} \gamma_{ij} (\mathbf{x}_i - \mu_j)^2 \end{aligned} \tag{18}$$

4. 聚类方法：DBSCAN

DBSCAN 是一种著名的基于密度聚类的算法，利用邻域参数来刻画样本分布的紧密程度。假设数据集 $\mathbf{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，DBSCAN 依赖于以下几个概念：

1. $\epsilon-$ 邻域，对任意 $\mathbf{x}_i \in \mathbf{D}$，其 $\epsilon-$ 邻域包含 $\mathbf{D}$ 中与样本 $\mathbf{x}_i$ 距离不大于 ϵ 的样本，记为 $N_\epsilon(\mathbf{x}_i) = \{\mathbf{x} \in \mathbf{D} | dist(\mathbf{x}, \mathbf{x}_i) \leq \epsilon\}$；
2. 若 $|N_\epsilon(\mathbf{x}_i)| \geq N_0$，则 $\mathbf{x}_i$ 是一个核心对象；
3. 若 $\mathbf{x}_j \in N_\epsilon(\mathbf{x}_i)$，则称 $\mathbf{x}_i, \mathbf{x}_j$ 密度直达；
4. 若 $\mathbf{x}_i = \mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n = \mathbf{x}_j$，为一个样本序列，其中 $\mathbf{p}_i, \mathbf{p}_j$ 密度直达，则称 $\mathbf{x}_i, \mathbf{x}_j$ 密度可达；
5. 若存在 $\mathbf{x}_k$，使得 $\mathbf{x}_i, \mathbf{x}_k$ 以及 $\mathbf{x}_j, \mathbf{x}_k$，密度可达，则 $\mathbf{x}_i, \mathbf{x}_j$ 密度相连。

4. 聚类方法：DBSCAN



4. 聚类方法：DBSCAN

基于这些概念，DBSCAN 将“簇”定义为：由密度可达关系导出的最大密度相连样本集合，即满足以下性质的样本子集：

1. 连接性：$\mathbf{x}_i \in \mathbf{C}, \mathbf{x}_j \in \mathbf{C} \Rightarrow \mathbf{x}_i, \mathbf{x}_j$ 密度相连；
2. 最大性：$\mathbf{x}_i \in \mathbf{C}, \mathbf{x}_i, \mathbf{x}_j$ 密度可达，则 $\mathbf{x}_j \in \mathbf{C}$。

4. 聚类方法：DBSCAN

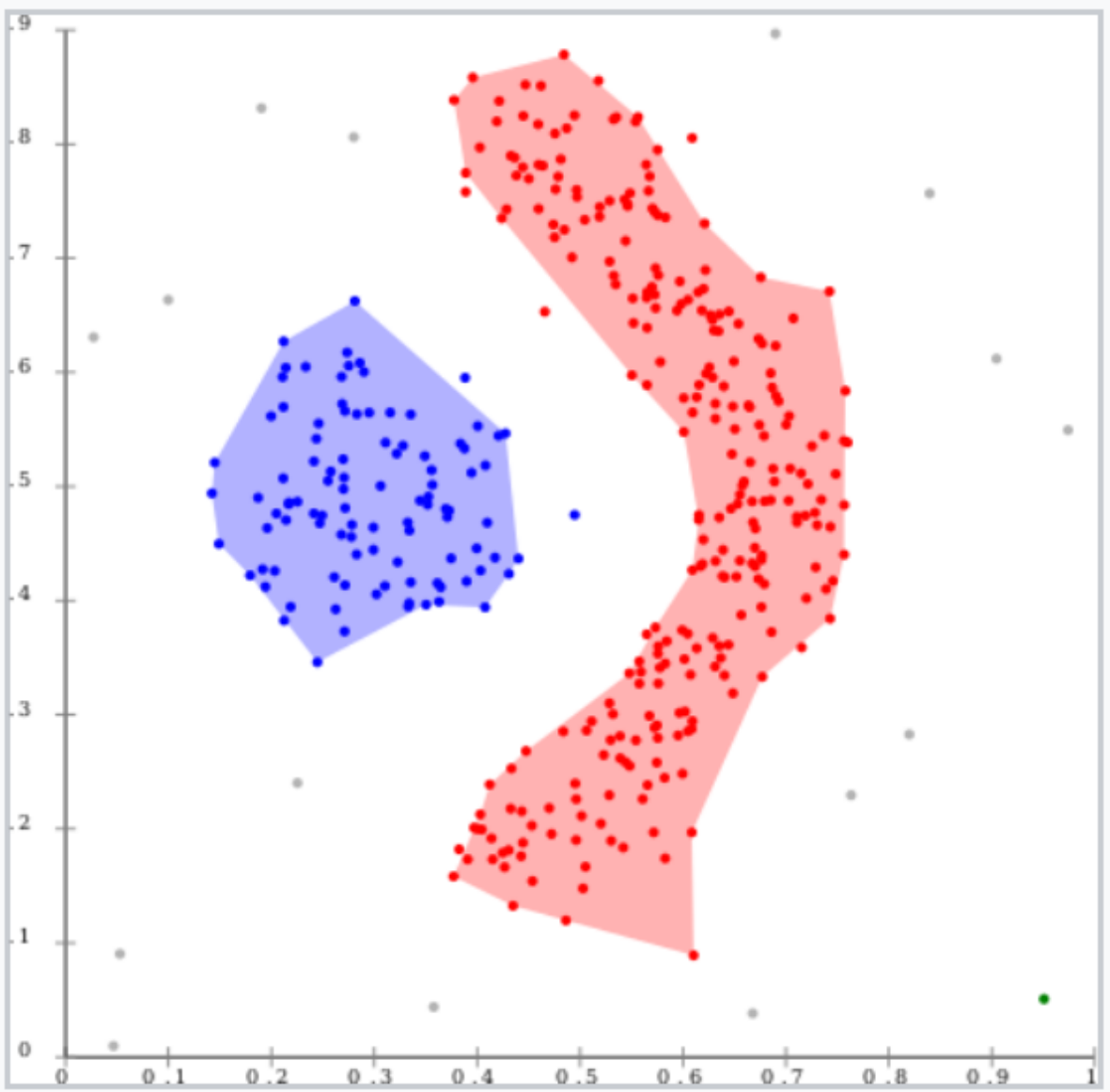
DBSCAN 算法：

STEP 1. 对数据集中每个样本确定其 $\epsilon-$ 邻域，确定所有核心对象；

STEP 2. 对每一核心对象找到其所有密度可达样本组成相应簇，忽略所有非核心对象；

STEP 3. 对每一非核心对象将其归入密度相连的簇，否则形成一个新的簇。

4. 聚类方法：DBSCAN



4. 聚类方法：DBSCAN

优点：

1. DBSCAN 不需要预先设定类簇数目；
2. DBSCAN 可以发现任意形状的簇，对噪声不敏感；
3. DBSCAN 包含两个参数 ϵ, N_0，可以充分利用领域知识；

缺点：

1. DBSCAN 质量依赖于距离定义；
2. DBSCAN 对具有不同密度分布的数据集效果不佳；
3. DBSCAN 包含两个参数 ϵ, N_0，调节困难。

4. 聚类方法：层次聚类 AGNES

层次聚类 (hierarchical clustering) 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. 数据集的划分可采用“自底向上”的聚合策略，也可采用“自顶向下”的分拆策略。

AGNES 是一种采用自底向上聚合策略的层次聚类算法。算法流程：

STEP 1. 将数据集中的每个样本看作一个初始聚类簇；

STEP 2. 找出距离最近的两个簇进行合并；

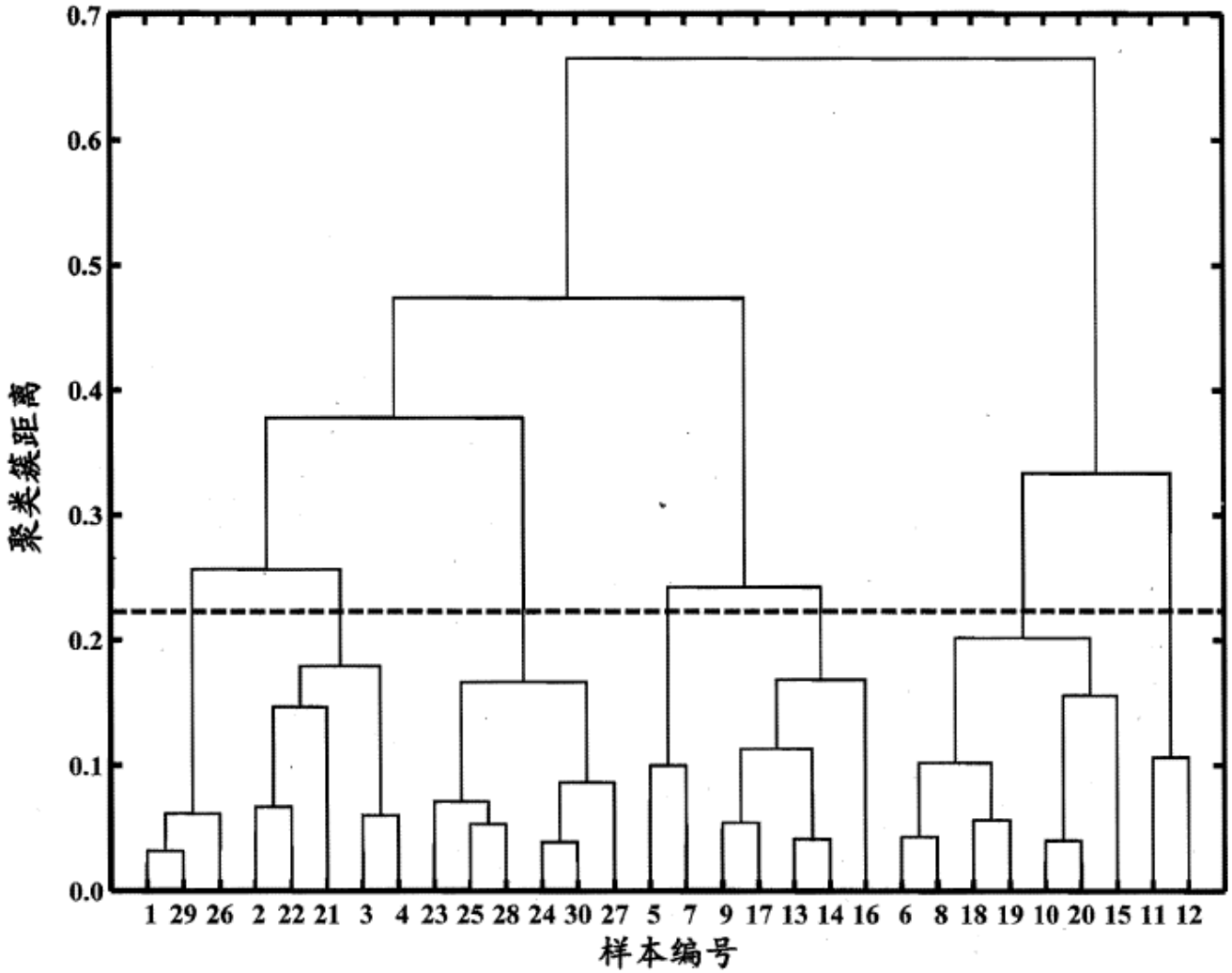
STEP 3. 检查簇数量，如果达到最大簇个数，跳出循环，否则转 STEP 2。

4. 聚类方法：层次聚类 AGNES

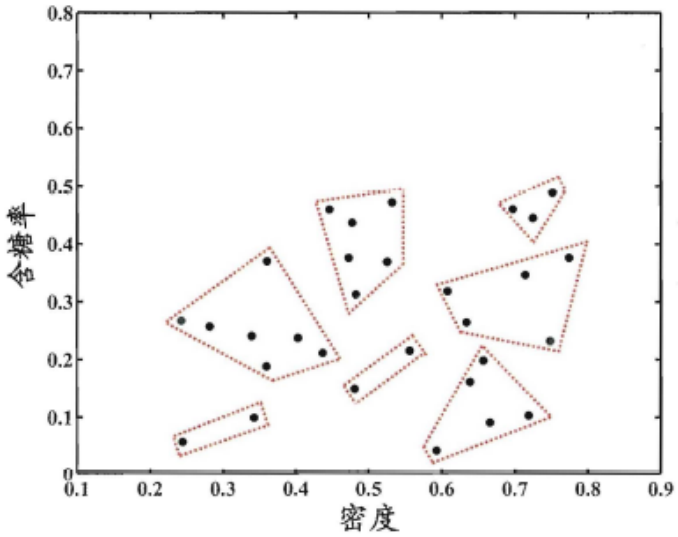
可以看到，层次聚类算法 AGNES 算法也依赖于簇与簇之间的距离定义，AGNES 定义簇与簇之间的距离包括以下三种：

1. 最小距离：$d_{min}(\mathbf{C}_i, \mathbf{C}_j) = \min_{\mathbf{x}\in\mathbf{C}_i, \mathbf{y}\in\mathbf{C}_j} dist(\mathbf{x}, \mathbf{y})$
2. 最大距离：$d_{max}(\mathbf{C}_i, \mathbf{C}_j) = \max_{\mathbf{x}\in\mathbf{C}_i, \mathbf{y}\in\mathbf{C}_j} dist(\mathbf{x}, \mathbf{y})$
3. 平均距离：$d_{avg}(\mathbf{C}_i, \mathbf{C}_j) = \frac{1}{|\mathbf{C}_i||\mathbf{C}_j|} \sum_{\mathbf{x}\in\mathbf{C}_i, \mathbf{y}\in\mathbf{C}_j} dist(\mathbf{x}, \mathbf{y})$

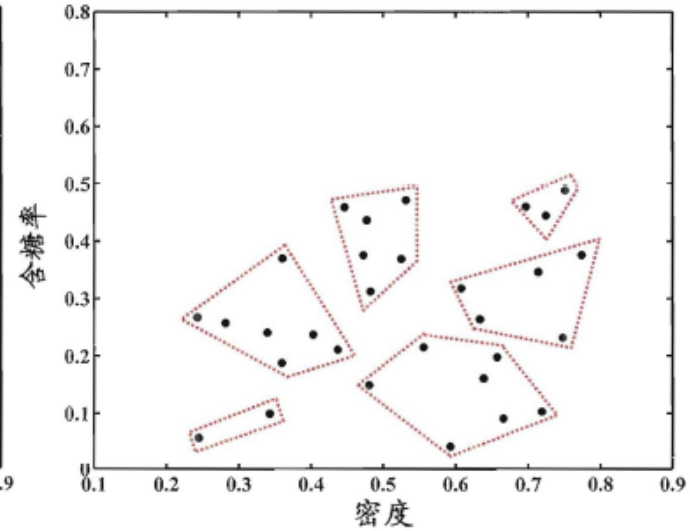
4. 聚类方法：层次聚类 AGNES



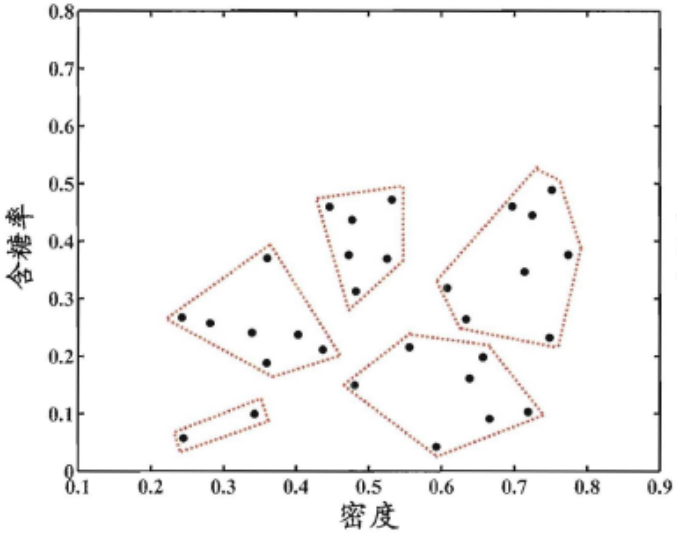
4. 聚类方法：层次聚类 AGNES



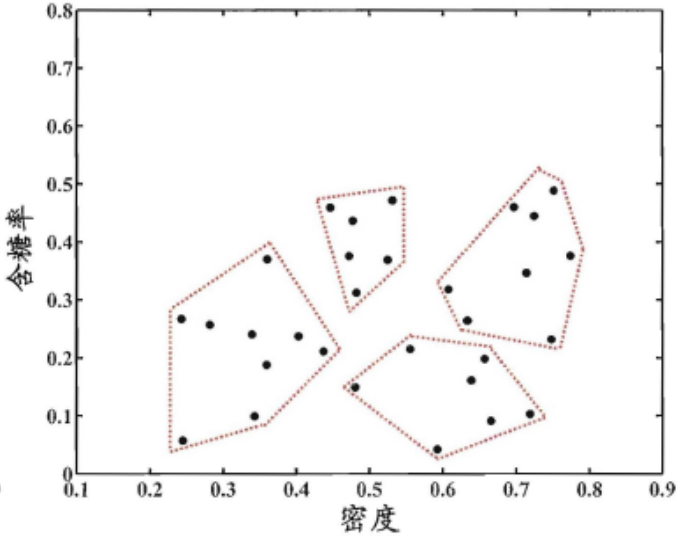
(a) 聚类簇数 $k = 7$



(b) 聚类簇数 $k = 6$



(c) 聚类簇数 $k = 5$



(d) 聚类簇数 $k = 4$

Thanks !