

## 九、聚类

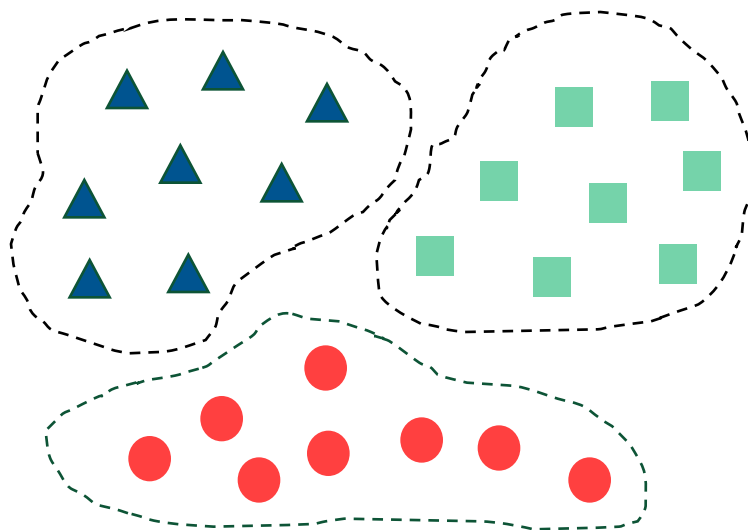
# 聚类 (Clustering)

---

在“无监督学习”任务中研究最多、应用最广

目标：将数据样本划分为若干个通常不相交的“簇”(cluster)

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



# 性能度量

---

聚类性能度量，亦称聚类“有效性指标” (validity index)

## □ 外部指标 (external index)

将聚类结果与某个“参考模型” (reference model) 进行比较  
如 Jaccard 系数, FM 指数, Rand 指数

## □ 内部指标 (internal index)

直接考察聚类结果而不用任何参考模型  
如 DB 指数, Dunn 指数等

基本想法：

- “簇内相似度” (intra-cluster similarity) 高, 且
- “簇间相似度” (inter-cluster similarity) 低

# 距离计算

---

距离度量 (distance metric) 需满足的基本性质：

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

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

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

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

常用距离形式：

闵可夫斯基距离 (Minkowski distance)

$$\text{dist}_{\text{mk}}(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_{u=1}^n |x_{iu} - x_{ju}|^p \right)^{\frac{1}{p}}$$

$p = 2$ : 欧氏距离(Euclidean distance)

$p = 1$ : 曼哈顿距离(Manhattan distance)

## 距离计算(续)

□ 对无序(non-ordinal)属性，可使用 VDM (Value Difference Metric)

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

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

□ 对混合属性，可使用 MinkovDM

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

# 必须记住

---



聚类的“好坏”不存在绝对标准

**the goodness of clustering depends on  
the opinion of the user**

# 常见聚类方法

---

## □ 原型聚类

- 亦称“基于原型的聚类” (prototype-based clustering)
- 假设：聚类结构能通过一组原型刻画
- 过程：先对原型初始化，然后对原型进行迭代更新求解
- 代表：**k均值聚类**，**学习向量量化(LVQ)**，**高斯混合聚类**

## □ 密度聚类

- 亦称“基于密度的聚类” (density-based clustering)
- 假设：聚类结构能通过样本分布的紧密程度确定
- 过程：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
- 代表：**DBSCAN**, OPTICS, DENCLUE

## □ 层次聚类 (hierarchical clustering)

- 假设：能够产生不同粒度的聚类结果
- 过程：在不同层次对数据集进行划分，从而形成树形的聚类结构
- 代表：**AGNES** (自底向上), **DIANA** (自顶向下)

# k-means

每个簇以该簇中所有样本点的“均值”表示

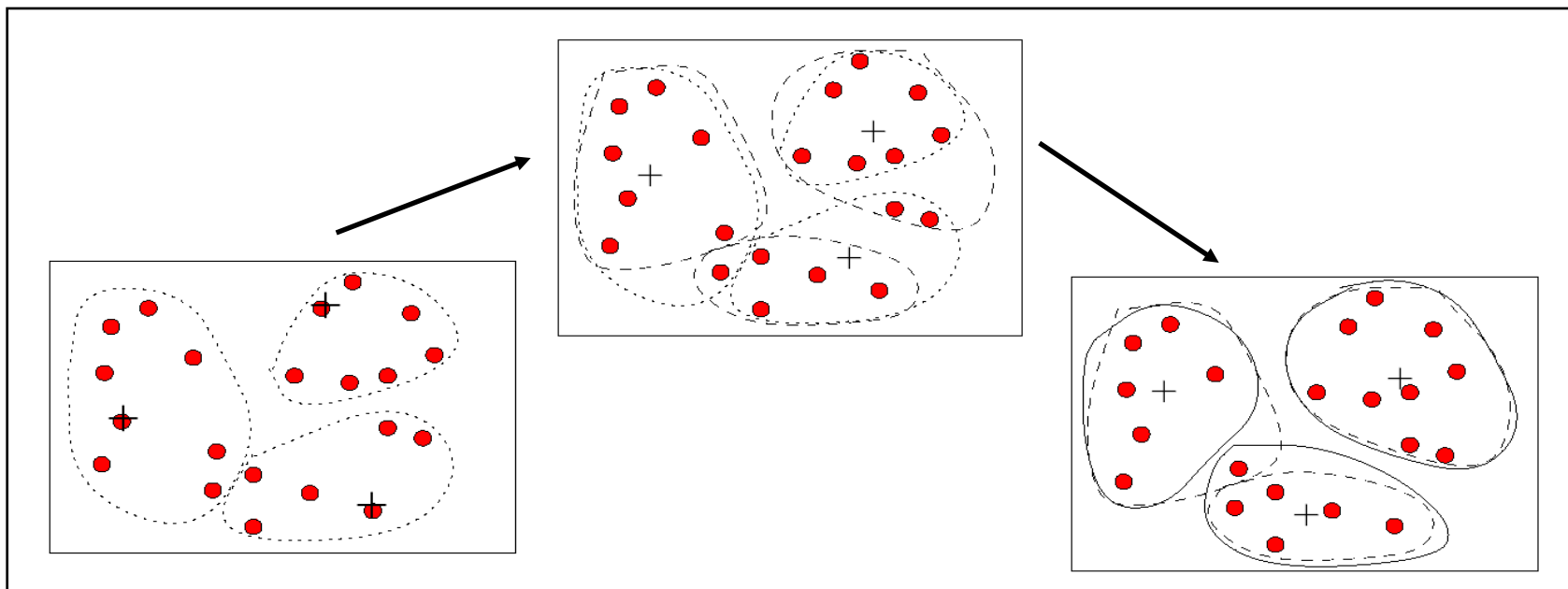
若不以均值向量为原型，而是以距离它最近的样本点为原型，则得到 k-medoids算法

Step1: 随机选取k个样本点作为簇中心

Step2: 将其他样本点根据其与簇中心的距离，划分给最近的簇

Step3: 更新各簇的均值向量，将其作为新的簇中心

Step4: 若所有簇中心未发生改变，则停止；否则执行 Step 2





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

也是试图找到一组原型向量来刻画聚类结构，但假设数据样本带有类别标记

实际上是通过聚类来形成类别的“子类”结构，每个子类对应一个聚类簇

---

**输入：** 样本集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ ;  
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  $\mathbf{p}_{i^*}$ ,  $i^* = \arg \min_{i \in \{1, 2, \dots, q\}} d_{ji}$ ;
- 6:   **if**  $y_j = t_{i^*}$  **then**
- 7:      $\mathbf{p}' = \mathbf{p}_{i^*} + \eta \cdot (\mathbf{x}_j - \mathbf{p}_{i^*})$     $\mathbf{x}_j$  与  $\mathbf{p}_{i^*}$  的类别相同
- 8:   **else**
- 9:      $\mathbf{p}' = \mathbf{p}_{i^*} - \eta \cdot (\mathbf{x}_j - \mathbf{p}_{i^*})$     $\mathbf{x}_j$  与  $\mathbf{p}_{i^*}$  的类别不同
- 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

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

---

# 高斯混合聚类 (Gaussian Mixture Clustering, GMM)

## 采用概率模型来表达聚类原型

$n$  维样本空间中的随机向量  $\mathbf{x}$  若服从高斯分布, 则其概率密度函数为

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

假设样本由下面这个高斯混合分布生成:

生成式模型

$$p_{\mathcal{M}}(\mathbf{x}) = \sum_{i=1}^k \alpha_i \cdot p(\mathbf{x} \mid \mu_i, \Sigma_i)$$

- 根据  $\alpha_1, \alpha_2, \dots, \alpha_k$  定义的先验分布选择高斯混合成分, 其中  $\alpha_i$  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

## 高斯混合聚类（续）

样本  $\mathbf{x}_j$  由第  $i$  个高斯混合成分生成的后验概率为：

$$p_{\mathcal{M}}(z_j = i \mid \mathbf{x}_j) = \frac{P(z_j = i) \cdot p_{\mathcal{M}}(\mathbf{x}_j \mid z_j = i)}{p_{\mathcal{M}}(\mathbf{x}_j)} = \frac{\alpha_i \cdot p(\mathbf{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^k \alpha_l \cdot p(\mathbf{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

简记为  $\gamma_{ji}$  ( $i = 1, 2, \dots, k$ )

参数估计可采用极大似然法，考虑最大化对数似然

$$LL(D) = \ln \left( \prod_{j=1}^m p_{\mathcal{M}}(\mathbf{x}_j) \right) = \sum_{j=1}^m \ln \left( \sum_{i=1}^k \alpha_i \cdot p(\mathbf{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right)$$

**EM 算法：**

- (E步) 根据当前参数计算每个样本属于每个高斯成分的后验概率  $\gamma_{ji}$
- (M步) 更新模型参数  $\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$

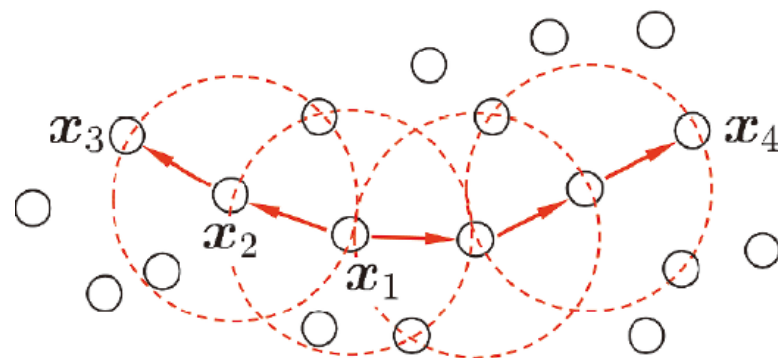
# DBSCAN

关键概念:

- 核心对象(core object): 若  $x_j$  的  $\epsilon$ -邻域至少包含  $MinPts$  个样本, 即  $|N_\epsilon(x_j)| \geq MinPts$ , 则  $x_j$  是一个核心对象;
- 密度直达(directly density-reachable): 若  $x_j$  位于  $x_i$  的  $\epsilon$ -邻域中, 且  $x_i$  是核心对象, 则称  $x_j$  由  $x_i$  密度直达;
- 密度可达(density-reachable): 对  $x_i$  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, 则称  $x_j$  由  $x_i$  密度可达;
- 密度相连(density-connected): 对  $x_i$  与  $x_j$ , 若存在  $x_k$  使得  $x_i$  与  $x_j$  均由  $x_k$  密度可达, 则称  $x_i$  与  $x_j$  密度相连.

令  $MinPts = 3$ ,  
虚线显示出  $\epsilon$  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

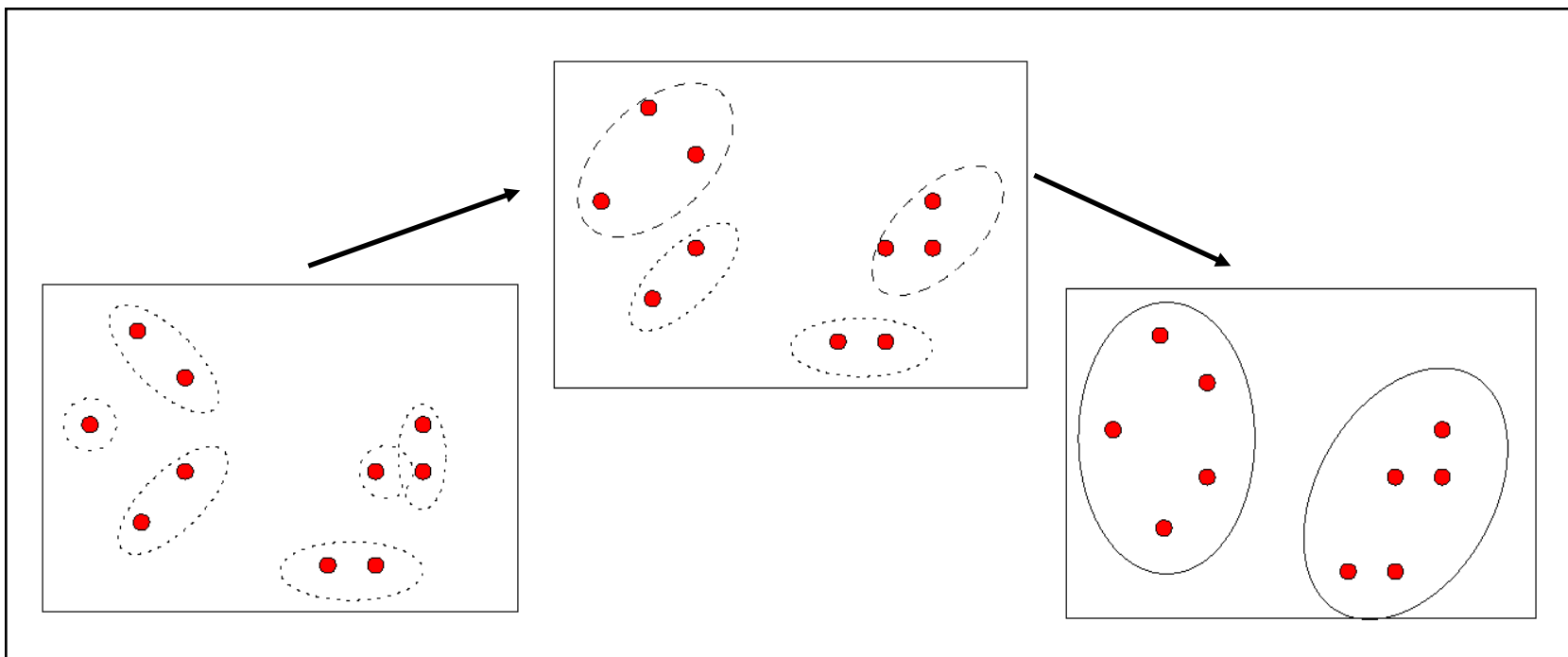


# AGNES (AGglomerative NESting)

Step1: 将每个样本点作为一个簇

Step2: 合并最近的两个簇

Step3: 若所有样本点都存在与一个簇中，则停止；否则转到 Step2



# AGNES

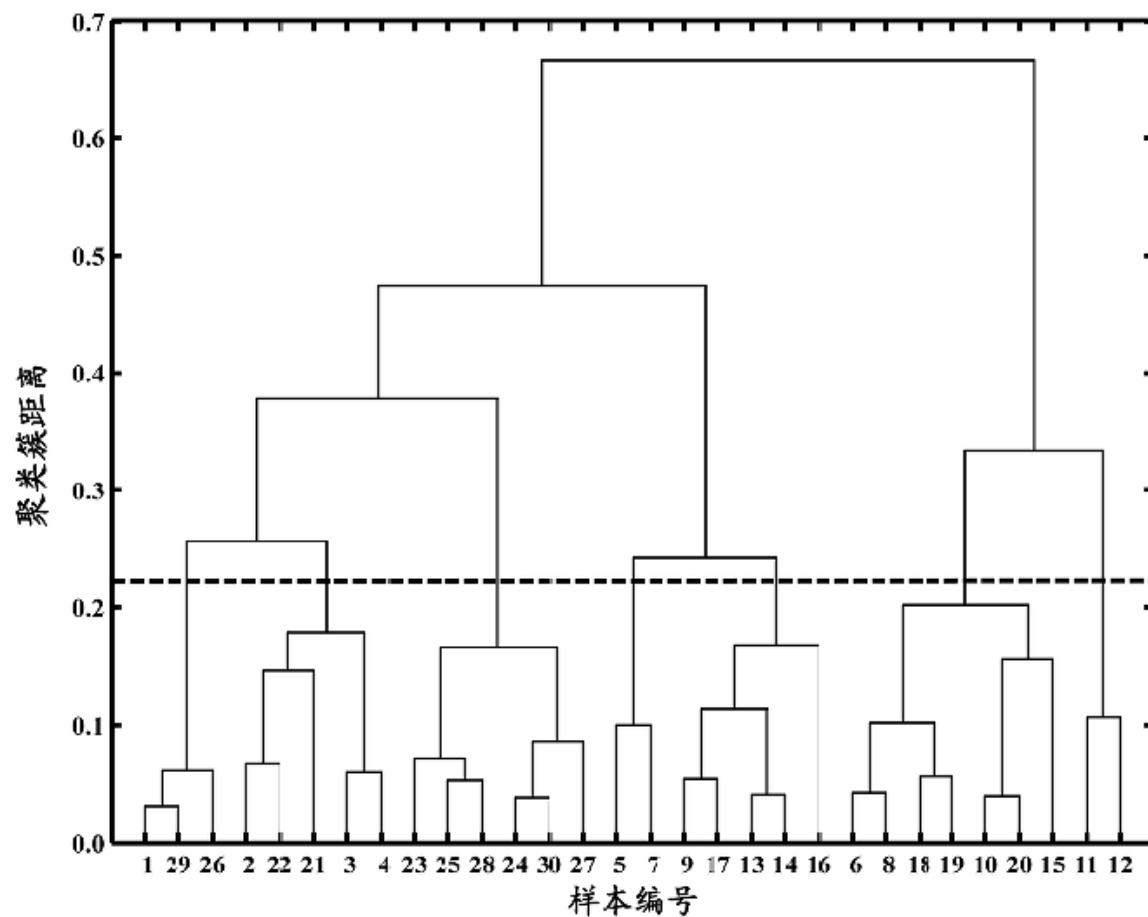


图 9.12 西瓜数据集 4.0 上 AGNES 算法生成的树状图(采用  $d_{\max}$ ). 横轴对应于样本编号, 纵轴对应于聚类簇距离.

# 聚类常用软件/工具包

---

## □ Kmeans (Parallel implementation)

<http://www.ece.northwestern.edu/~wkliao/Kmeans/index.html>

## □ Agnes etc. (Hierarchical clustering in R 'cluster' package)

<https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/agnes.html>

## □ Spectral Clustering

<https://www.cis.upenn.edu/~jshi/software/>

□ ... ..

前往第十站.....

