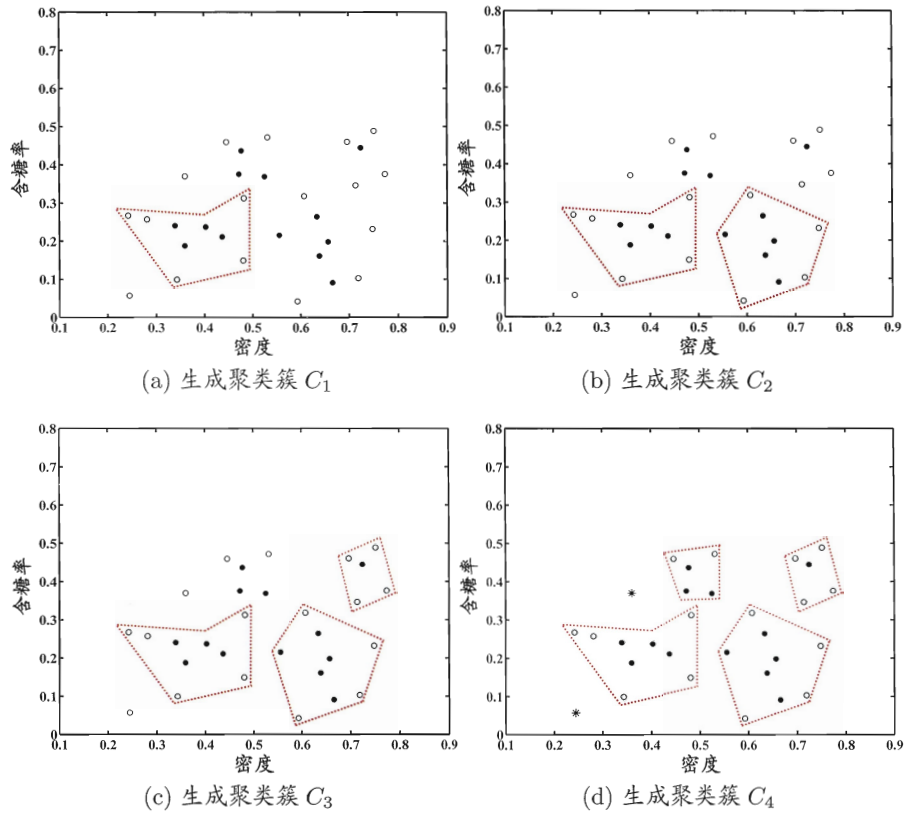


(a) 生成聚类簇 C_1 (b) 生成聚类簇 C_2

(c) 生成聚类簇 C_3 (d) 生成聚类簇 C_4

图 9.10 DBSCAN 算法($\epsilon = 0.11$, $MinPts = 5$)生成聚类簇的先后情况. 核心对象、非核心对象、噪声样本分别用"•" "○" "∗"表示, 红色虚线显示出簇划分.

$$C_2 = \{\boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_9, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{16}, \boldsymbol{x}_{17}, \boldsymbol{x}_{21}\} ;$$

$$C_3 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_{22}, \boldsymbol{x}_{26}, \boldsymbol{x}_{29}\} ;$$

$$C_4 = \{\boldsymbol{x}_{24}, \boldsymbol{x}_{25}, \boldsymbol{x}_{27}, \boldsymbol{x}_{28}, \boldsymbol{x}_{30}\} .$$

9.6 层次聚类

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 AGglomerative NESting 的简写.

AGNES 是一种采用自底向上聚合策略的层次聚类算法. 它先将数据集中的每个样本看作一个初始聚类簇, 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, 直至达到预设的聚类簇个数. 这里的关键是如何计算聚类簇之间的距离. 实际上, 每个簇是一个样本集合, 因此, 只需采用关于集合的某种距离即可. 例如, 给定聚类簇 C_i 与 C_j, 可通过下面的式子来计算距离:

集合间的距离计算常采用豪斯多夫距离 (Hausdorff distance), 参见习题 9.2.

$$\text{最小距离: } d_{\min}(C_i, C_j) = \min_{\boldsymbol{x}\in C_i, \boldsymbol{z}\in C_j} \operatorname{dist}(\boldsymbol{x}, \boldsymbol{z}) , \tag{9.41}$$

$$\text{最大距离: } d_{\max}(C_i, C_j) = \max_{\boldsymbol{x}\in C_i, \boldsymbol{z}\in C_j} \operatorname{dist}(\boldsymbol{x}, \boldsymbol{z}) , \tag{9.42}$$

$$\text{平均距离: } d_{\text{avg}}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\boldsymbol{x}\in C_i} \sum_{\boldsymbol{z}\in C_j} \operatorname{dist}(\boldsymbol{x}, \boldsymbol{z}) . \tag{9.43}$$

显然, 最小距离由两个簇的最近样本决定, 最大距离由两个簇的最远样本决定, 而平均距离则由两个簇的所有样本共同决定. 当聚类簇距离由 $d_{\min}$、$d_{\max}$ 或

通常使用 $d_{\min}$, $d_{\max}$ 或 d_{avg}.

初始化单样本聚类簇.

初始化聚类簇距离矩阵.

$i^* < j^*$.

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;
聚类簇数 k.

过程:

```
for j = 1, 2, ..., m do
    C_j = {x_j}
end for
for i = 1, 2, ..., m do
    for j = 1, 2, ..., 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*} 和 C_{j*};
    合并 C_{i*} 和 C_{j*}: C_{i*} = C_{i*} ⋃ C_{j*};
    for j = j* + 1, j* + 2, ..., 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, 2, ..., q - 1 do
        M(i*, j) = d(C_{i*}, C_j);
        M(j, i*) = M(i*, j)
    end for
    q = q - 1
end while
```

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

图 9.11 AGNES 算法

d_{avg} 计算时, AGNES 算法被相应地称为“单链接”(single-linkage)、“全链接”(complete-linkage) 或“均链接”(average-linkage)算法.

AGNES 算法描述如图 9.11 所示. 在第 1–9 行, 算法先对仅含一个样本的初始聚类簇和相应的距离矩阵进行初始化; 然后在第 11–23 行, AGNES 不断合并距离最近的聚类簇, 并对合并得到的聚类簇的距离矩阵进行更新; 上述过程不断重复, 直至达到预设的聚类簇数.

西瓜数据集 4.0 见 p.202 的表 9.1.

以西瓜数据集 4.0 为例, 令 AGNES 算法一直执行到所有样本出现在同一个簇中, 即 $k = 1$, 则可得到图 9.12 所示的“树状图” (dendrogram), 其中每层链接一组聚类簇.

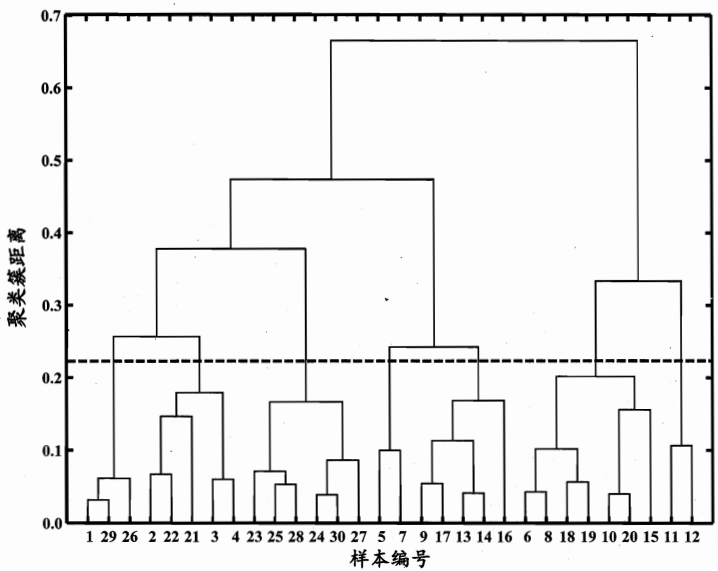


图 9.12 西瓜数据集 4.0 上 AGNES 算法生成的树状图(采用 $d_{\max}$). 横轴对应于样本编号, 纵轴对应于聚类簇距离.

在树状图的特定层次上进行分割, 则可得到相应的簇划分结果. 例如, 以图 9.12 中所示虚线分割树状图, 将得到包含 7 个聚类簇的结果:

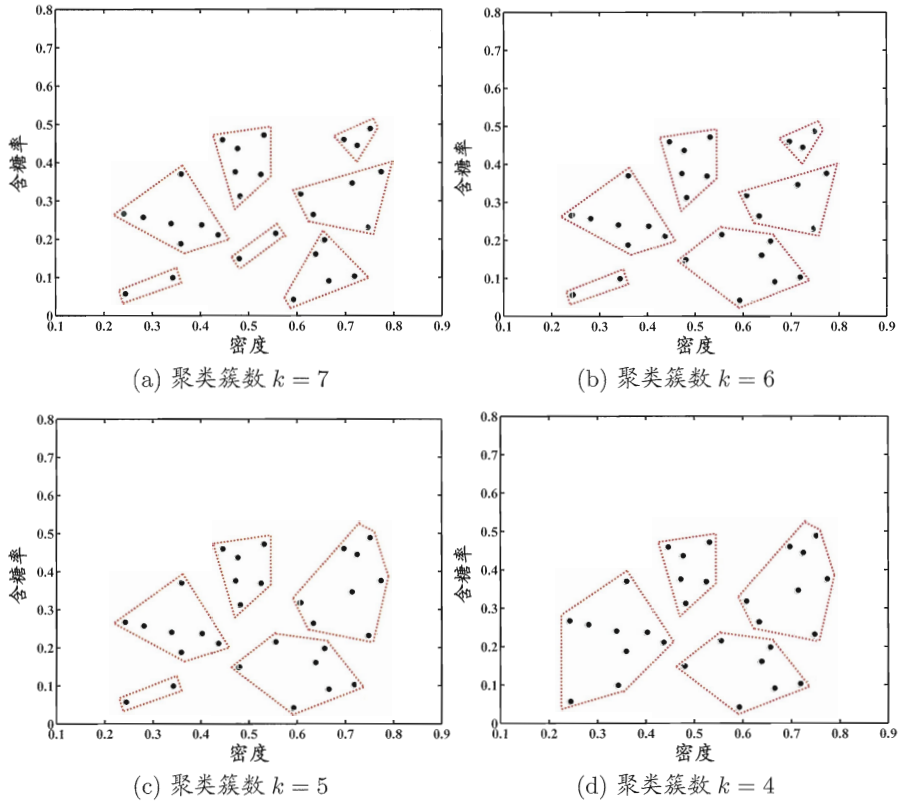
$$C_1 = \{\boldsymbol{x}_1, \boldsymbol{x}_{26}, \boldsymbol{x}_{29}\};\ C_2 = \{\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_{21}, \boldsymbol{x}_{22}\};$$

$$C_3 = \{\boldsymbol{x}_{23}, \boldsymbol{x}_{24}, \boldsymbol{x}_{25}, \boldsymbol{x}_{27}, \boldsymbol{x}_{28}, \boldsymbol{x}_{30}\};\ C_4 = \{\boldsymbol{x}_5, \boldsymbol{x}_7\};$$

$$C_5 = \{\boldsymbol{x}_9, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{16}, \boldsymbol{x}_{17}\};\ C_6 = \{\boldsymbol{x}_6, \boldsymbol{x}_8, \boldsymbol{x}_{10}, \boldsymbol{x}_{15}, \boldsymbol{x}_{18}, \boldsymbol{x}_{19}, \boldsymbol{x}_{20}\};$$

$$C_7 = \{\boldsymbol{x}_{11}, \boldsymbol{x}_{12}\}.$$

将分割层逐步提升, 则可得到聚类簇逐渐减少的聚类结果. 例如图 9.13 显示出了从图 9.12 中产生 7 至 4 个聚类簇的划分结果.



(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 $k = 4$

图 9.13 西瓜数据集 4.0 上 AGNES 算法(采用 $d_{\max}$)在不同聚类簇数($k = 7, 6, 5, 4$)时的簇划分结果. 样本点用"•"表示, 红色虚线显示出簇划分.

9.7 阅读材料

例如同一堆水果, 既能按大小, 也能按颜色, 甚至能按产地聚类.

聚类也许是机器学习中"新算法"出现最多、最快的领域. 一个重要原因是聚类不存在客观标准; 给定数据集, 总能从某个角度找到以往算法未覆盖的某种标准从而设计出新算法 [Estivill-Castro, 2002]. 相对于机器学习其他分支来说, 聚类的知识还不够系统化, 因此著名教科书 [Mitchell, 1997] 中甚至没有关于聚类的章节. 但聚类技术本身在现实任务中非常重要, 因此本章勉强采用了"列举式"的叙述方式, 相较于其他各章给出了更多的算法描述. 关于聚类