東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

人工智能概论

第八章：聚类与无监督学习

丁烨，计算机科学与技术学院
dingye@dgut.edu.cn

目录

基本概念

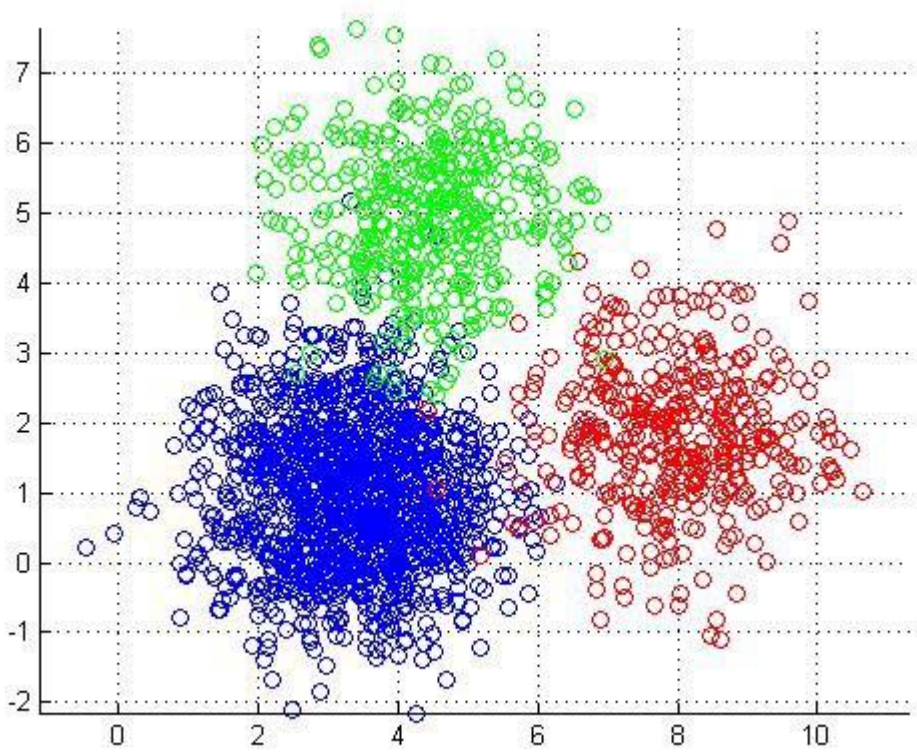
无监督学习

- 无监督学习（Unsupervised Learning）
- 训练样本的标记信息未知
- 通过对无标记训练样本的学习
- 来揭示数据的内在性质及规律
- 为进一步的数据分析提供基础

基本概念

聚类

- 聚类（Clustering）
- 无监督学习中研究最多、应用最广的任务
- 聚类试图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）”



基本概念

聚类

- 通过聚类的划分，每个簇可能对应于一些潜在的概念（类别）
- 如“浅色瓜”、“深色瓜”，“有籽瓜”、“无籽瓜”等
- 这些概念对聚类算法而言事先是未知的
- 聚类过程仅能自动形成簇结构
- 簇所对应的概念语义需由使用者来把握和命名

基本概念

聚类

- 假定样本集 $D = \{x_1, x_2, \ldots, x_m\}$ 包含 m 个无标记样本
- 每个样本 x_i 是一个 n 维特征向量
- 则聚类算法将样本集 D 划分为 k 个不相交的簇 $\{C_l | l = 1,2, \ldots, k\}$
- 其中 $C_{l'} \cap_{l' \neq l} C_l = \emptyset$ 且 $D = \bigcup_{l=1}^{k} C_l$
- 相应地，用 $\lambda_j \in \{1,2, \ldots, k\}$ 表示样本 x_j 的“簇标记（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表示：
- $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$

基本概念

聚类

- 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
- 例如，在一些商业应用中需对新用户的类型进行判别
- 但定义“用户类型”对商家来说却可能不太容易
- 此时往往可先对用户数据进行聚类
- 根据聚类结果将每个簇定义为一个类
- 然后再基于这些类训练分类模型，用于判别新用户的类型

基本概念

性能度量

- 什么样的聚类结果比较好呢？
- 直观上看，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
- 簇间相似度（Inter-cluster Similarity）：低

基本概念

性能度量

- 聚类性能度量大致有两类：
- 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利用任何参考模型

基本概念

性能度量

- 对数据集 $D = \{x_1, x_2, \ldots, x_m\}$
- 我们将聚类给出的簇划分和参考模型给出的簇划分两两配对考虑，则：

$$a = |SS|, SS = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}$$

$$b = |SD|, SD = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}$$

$$c = |DS|, DS = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}$$

$$d = |DD|, DD = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}$$

基本概念

性能度量

- 可以导出下面这些常用的聚类性能度量外部指标：
- Jaccard 系数（Jaccard Coefficient，JC）：

$$JC = \frac{a}{a + b + c}$$

- FM 指数（Fowlkes-Mallows Index，FMI）：

$$FMI = \sqrt{\frac{a}{a + b} \cdot \frac{a}{a + c}}$$

基本概念

性能度量

- 可以导出下面这些常用的聚类性能度量外部指标：
- Rand 指数（Rand Index，RI）：

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

- 显然，上述性能度量的结果值均在 $[0,1]$ 区间，值越大越好

基本概念

性能度量

- 对数据集 $D = \{x_1, x_2, \ldots, x_m\}$
- 考虑聚类结果的簇划分 $C = \{C_1, C_2, \ldots, C_k\}$，定义：
- 簇内样本间的平均距离：

$$\mathrm{avg}(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \leqslant i < j \leqslant |C|} \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- 簇内样本间的最远距离：

$$\mathrm{diam}(C) = \max_{1 \leqslant i < j \leqslant |C|} \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

基本概念

性能度量

- 对数据集 $D = \{x_1, x_2, \ldots, x_m\}$
- 考虑聚类结果的簇划分 $C = \{C_1, C_2, \ldots, C_k\}$，定义：
- 簇 C_i 与簇 C_j 最近样本间的距离：

$$d_{\min}(C_i, C_j) = \min_{\boldsymbol{x}_i \in C_i, \boldsymbol{x}_j \in C_j} \operatorname{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- 簇 C_i 与簇 C_j 中心点间的距离：

$$d_{\text{cen}}(C_i, C_j) = \operatorname{dist}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$$

基本概念

性能度量

- 可以导出下面这些常用的聚类性能度量内部指标：
- DB 指数（Davies-Bouldin Index，DBI），值越小越好：

$$\mathrm{DBI} = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\mathrm{avg}(C_i)+\mathrm{avg}(C_j)}{d_{\mathrm{cen}}(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j)}\right)$$

基本概念

性能度量

- 可以导出下面这些常用的聚类性能度量内部指标：
- Dunn 指数（Dunn Index，DI），值越大越好：

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

基本概念

距离计算

- 在聚类问题的训练过程中
- 我们需要对样本的“相似度（Similarity）”做度量
- 通常我们是基于某种形式的距离来定义相似度
- 距离越大，相似度越小

基本概念

距离计算

- 距离度量（Distance Measure）函数
- 非负性：$\mathrm{dist}(x_i, x_j) \geq 0$
- 同一性：$\mathrm{dist}(x_i, x_j) = 0$ 当且仅当 $x_i = x_j$
- 对称性：$\mathrm{dist}(x_i, x_j) = \mathrm{dist}(x_j, x_i)$
- 直递性：$\mathrm{dist}(x_i, x_j) \leq \mathrm{dist}(x_i, x_k) + \mathrm{dist}(x_k, x_j)$

基本概念

距离计算

- 闵可夫斯基距离（Minkowski Distance）

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

基本概念

距离计算

- 闵可夫斯基距离（Minkowski Distance）
- $p = 2$ 时，闵可夫斯基距离即欧氏距离（Euclidean Distance）

$$\mathrm{dist}_{\mathrm{ed}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2 = \sqrt{\sum_{u=1}^{n} |x_{iu} - x_{ju}|^2}$$

基本概念

距离计算

- 闵可夫斯基距离（Minkowski Distance）
- $p = 1$ 时，闵可夫斯基距离即曼哈顿距离（Manhattan Distance）

$$\mathrm{dist}_{\mathrm{man}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_1 = \sum_{u=1}^{n} |x_{iu} - x_{ju}|$$

基本概念

距离计算

- 在讨论距离计算时，特征上是否定义了“序”关系非常重要
- 有序（Ordinal）特征：
- 例如，定义域为 $\{1,2,3\}$ 的离散特征与连续特征的性质更接近一些
- 能直接在特征值上计算距离 1 与 2 比较接近、1 与 3 比较远

基本概念

距离计算

- 无序（Non-ordinal）特征：
- 定义域为“飞机、火车、轮船”这样的离散属性
- 则不能直接在属性值上计算距离
- 显然，闵可夫斯基距离可用于有序属性

基本概念

距离计算

- VDM（Value Difference Metric）
- 令 $m_{u,a}$ 表示在特征 u 上取值为 a 的样本数
- 令 $m_{u,a,i}$ 表示在第 i 个样本簇中在特征 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：

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

基本概念

距离计算

- 将闵可夫斯基距离和 VDM 结合即可处理混合属性
- 假定有 n_c 个有序属性，$n - n_c$ 个无序属性
- 不失一般性，令有序属性排列在无序属性之前，则：

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

基本概念

距离计算

- 当样本空间中不同特征的重要性不同时
- 可使用“加权距离（Weighted Distance）”
- 以加权闵可夫斯基距离为例：

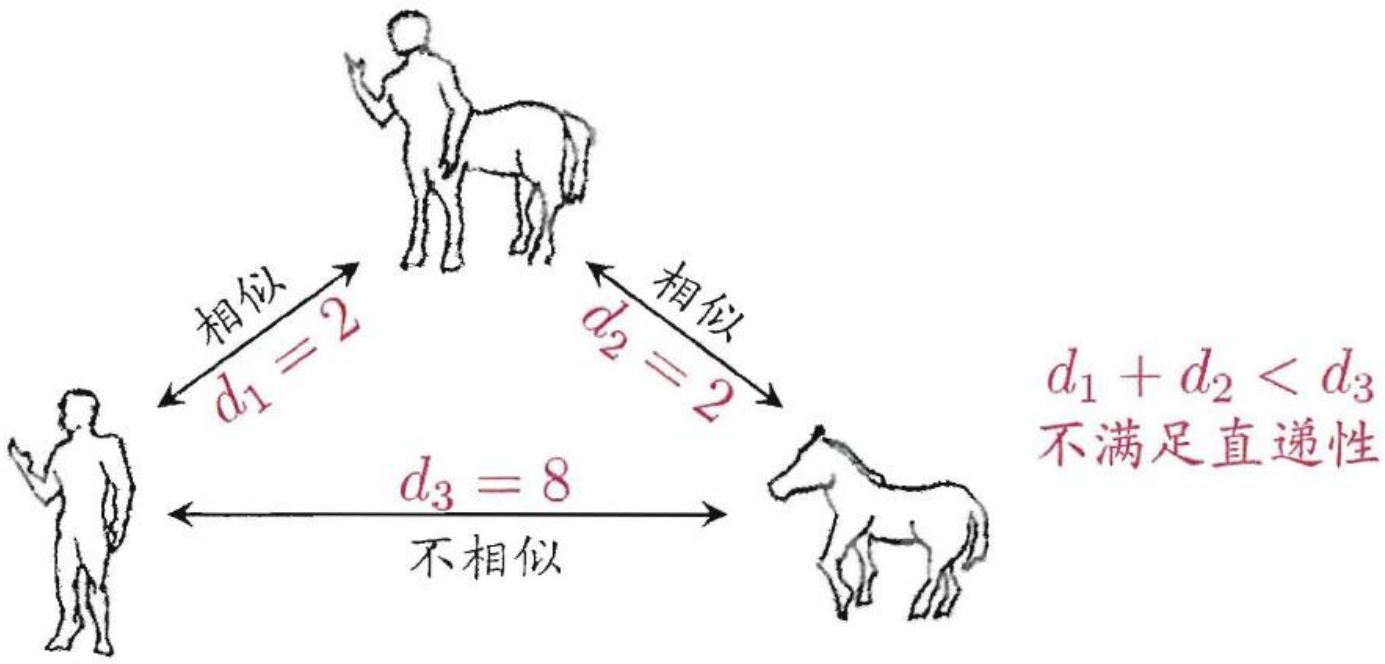
$$\mathrm{dist}_{\mathrm{wmk}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(w_1 \cdot |x_{i1} - x_{j1}|^p + \ldots + w_n \cdot |x_{in} - x_{jn}|^p \right)^{\frac{1}{p}}$$

- 其中权重 $w_i \geq 0\ (i = 1,2,\ldots,n)$ 表征不同特征的重要性
- 通常 $\sum_{i=1}^{n} w_i = 1$

基本概念

距离计算

- 非度量距离
- Non-metric Distance
- 实际距离不符合度量距离的情况
- 可通过“距离度量学习（Distance Metric Learning）”来实现



目录

原型聚类

基本概念

- 原型聚类（Prototype-based Clustering）
- 此类算法假设聚类结构能通过一组原型刻画
- 在现实聚类任务中极为常用
- 通常情形下，算法先对原型进行初始化
- 然后对原型进行迭代更新求解
- 采用不同的原型表示、不同的求解方式将产生不同的算法

原型聚类

k 均值

- k 均值（k-Means）
- 给定样本集 $D = \{x_1, x_2, \ldots, x_m\}$
- 针对聚类所得簇划分 $C = \{C_1, C_2, \ldots, C_k\}$ 最小化平方误差：

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

原型聚类

k 均值

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

- 其中 $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ 是簇 C_i 的均值向量
- k 均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

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

- 最小化该式并不容易
- 找到它的最优解需考察样本集 D 所有可能的簇划分
- 这是一个 NP 难问题
- 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

原型聚类

k 均值

表 9.1 西瓜数据集 4.0

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

- 假定聚类簇数 $k = 3$
- 算法开始时随机选取三个样本 x_6，x_{12}，x_{27} 作为初始均值向量：

$$\mu_1 = (0.403; 0.237)$$
$$\mu_2 = (0.343; 0.099)$$
$$\mu_3 = (0.532; 0.472)$$

原型聚类

k 均值

- 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：

$$C_1 = \{\boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{15}, \boldsymbol{x}_{17}, \boldsymbol{x}_{18}, \boldsymbol{x}_{19}, \boldsymbol{x}_{20}, \boldsymbol{x}_{23}\};$$

$$C_2 = \{\boldsymbol{x}_{11}, \boldsymbol{x}_{12}, \boldsymbol{x}_{16}\};$$

$$C_3 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_{21}, \boldsymbol{x}_{22}, \boldsymbol{x}_{24}, \boldsymbol{x}_{25}, \boldsymbol{x}_{26}, \boldsymbol{x}_{27}, \boldsymbol{x}_{28}, \boldsymbol{x}_{29}, \boldsymbol{x}_{30}\}.$$

原型聚类

k 均值

- 于是，可从 C_1、C_2、C_3 分别求出新的均值向量：

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

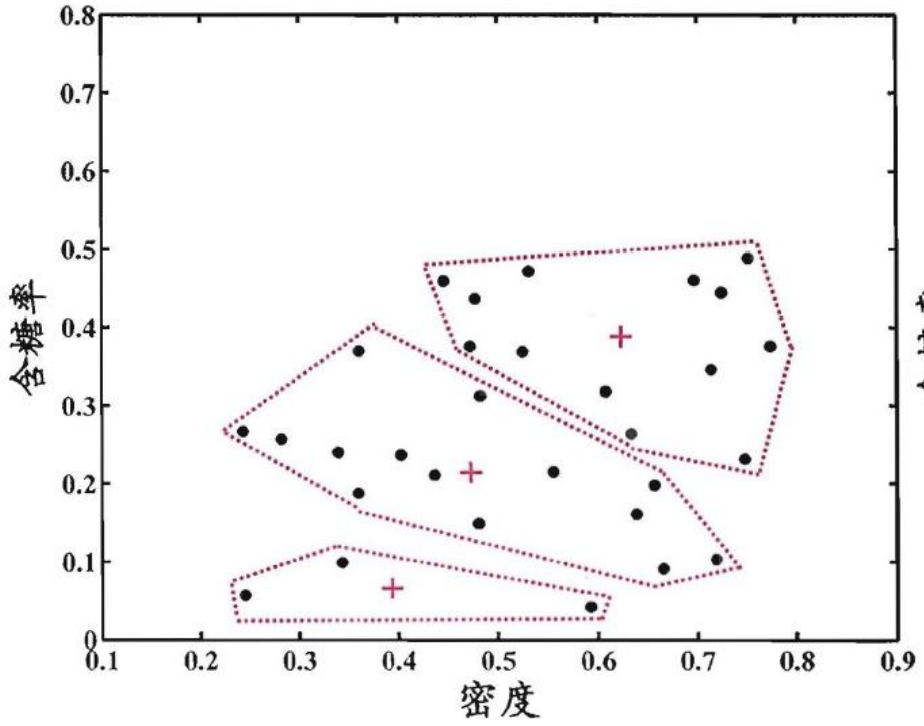
原型聚类

k 均值

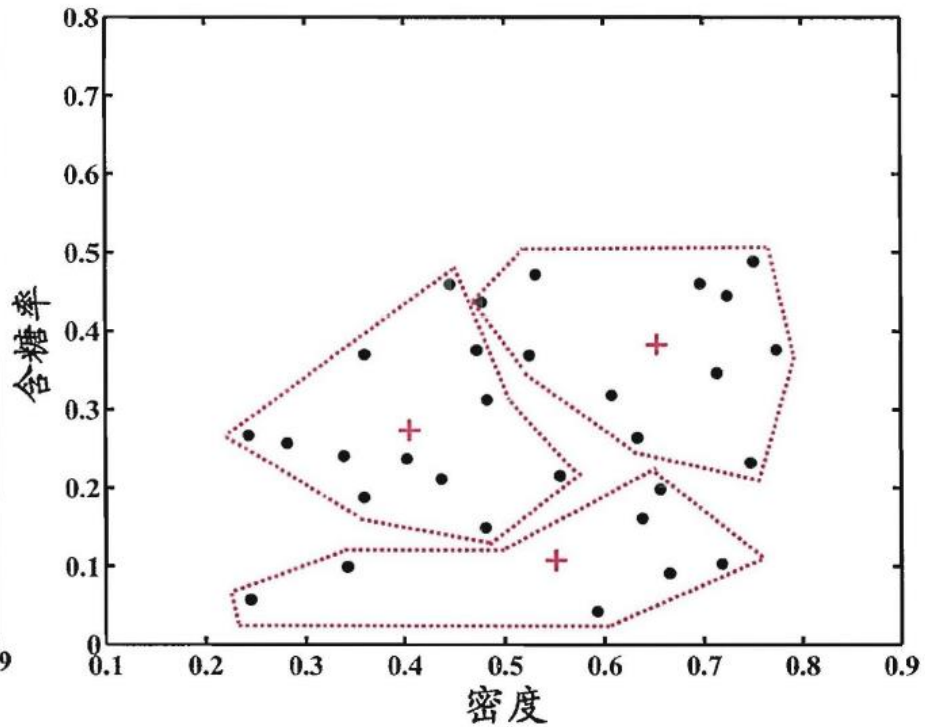
- 更新当前均值向量后，不断重复上述过程
- 第五轮迭代产生的结果与第四轮迭代相同
- 于是算法停止，得到最终的簇划分

原型聚类

k 均值



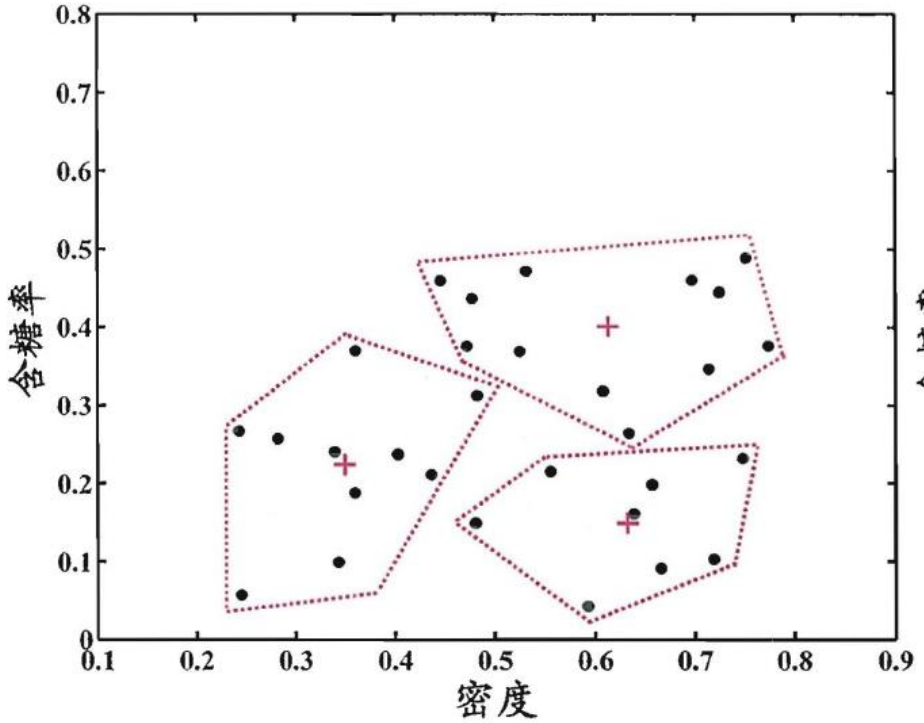
(a) 第一轮迭代后



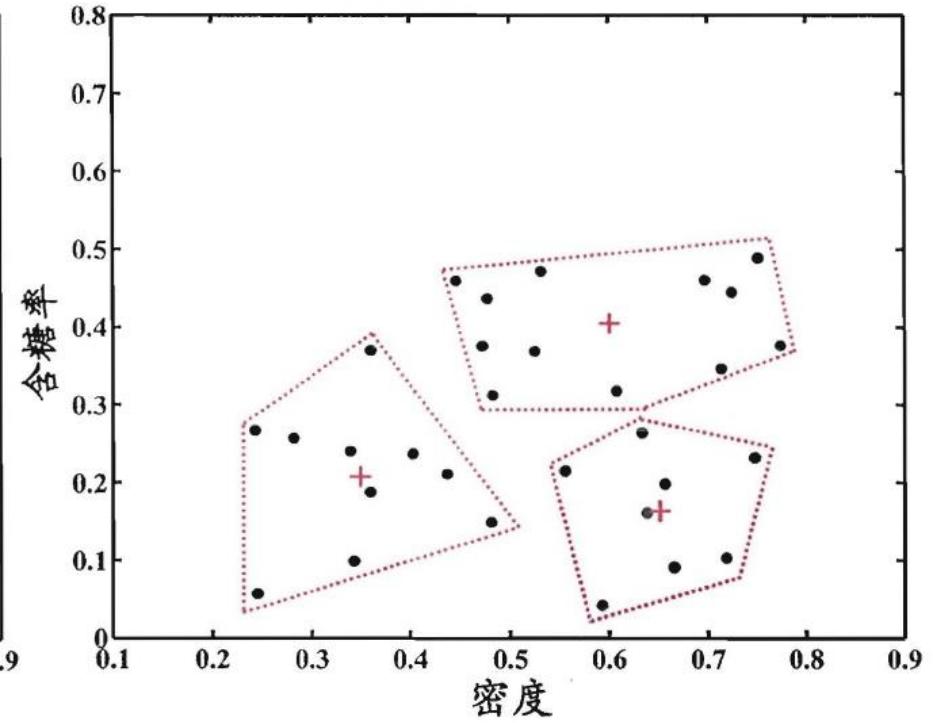
(b) 第二轮迭代后

原型聚类

k 均值



(c) 第三轮迭代后



(d) 第四轮迭代后

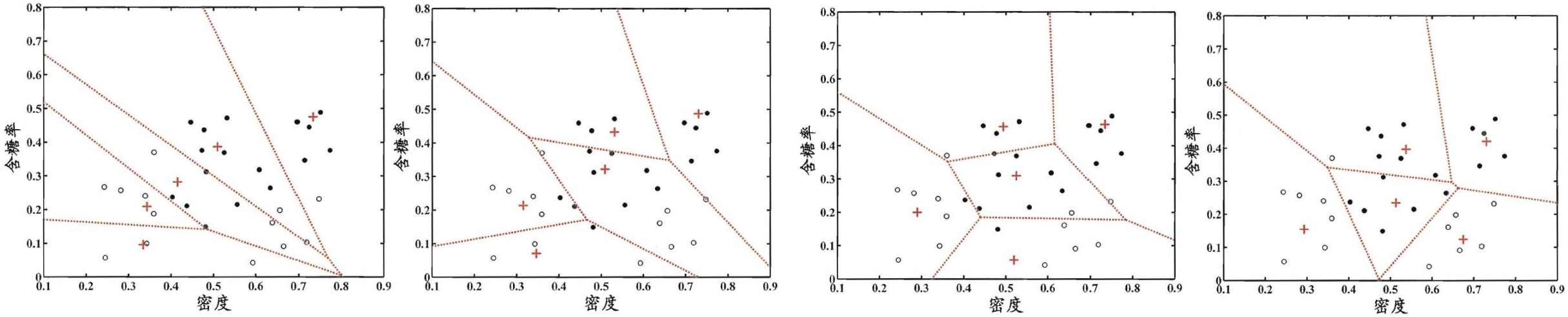
原型聚类

学习向量量化

- 学习向量量化（Learning Vector Quantization，LVQ）
- 与 k 均值算法类似，LVQ 也试图找到一组原型向量来刻画聚类结构
- 但与一般聚类算法不同的是
- LVQ 假设数据样本带有类别标记
- 学习过程利用样本的这些监督信息来辅助聚类

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

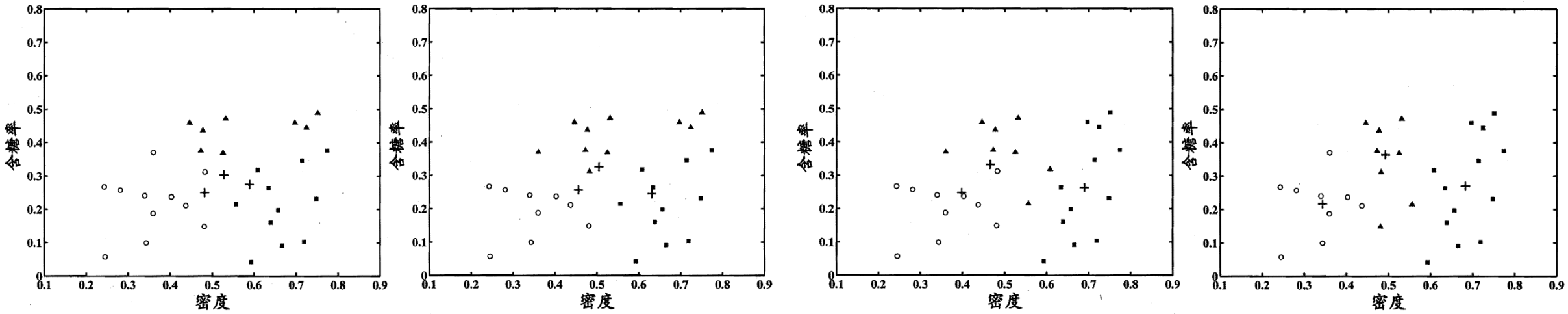
原型聚类

高斯混合聚类

- 高斯混合（Mixture-of-Gaussian）聚类
- 采用概率模型来表达聚类原型
- 假设样本的生成过程由高斯混合分布给出：
- 首先，根据先验分布选择高斯混合成分
- 然后，根据被选择的混合成分的概率密度函数进行采样生成样本
- 簇划分由原型对应后验概率确定

原型聚类

高斯混合聚类



(a) 5 轮迭代后　(b) 10 轮迭代后　(c) 20 轮迭代后　(d) 50 轮迭代后

目录

密度聚类

基本概念

- 密度聚类（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确定
- 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以获得最终的聚类结果

密度聚类

DBSCAN

- DBSCAN
- 一种著名的密度聚类算法
- 它基于一组“邻域（Neighborhood）”参数 $(\epsilon, MinPts)$
- 来刻画样本分布的紧密程度

密度聚类

DBSCAN

- DBSCAN 定义了以下的概念：
- ϵ 邻域
- 样本集 D 中与 x_j 的距离不大于 ϵ 的样本
- 核心对象（Core Object）
- 若 x_j 的 ϵ 邻域至少包含 $MinPts$ 个样本，则 x_j 是一个核心对象

密度聚类

DBSCAN

- 密度直达（Directly Density-reachable）
- 若 x_j 位于 x_i 的 ϵ 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）
- 若存在样本序列 $p_1, p_2, \ldots, p_n$，其中 $p_1 = x_i, p_n = p_j$，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）
- 若存在 x_k 使得 x_i 与 x_j 均由 x_k 密度可达，则称 x_i 和 x_j 密度相连

密度聚类

DBSCAN

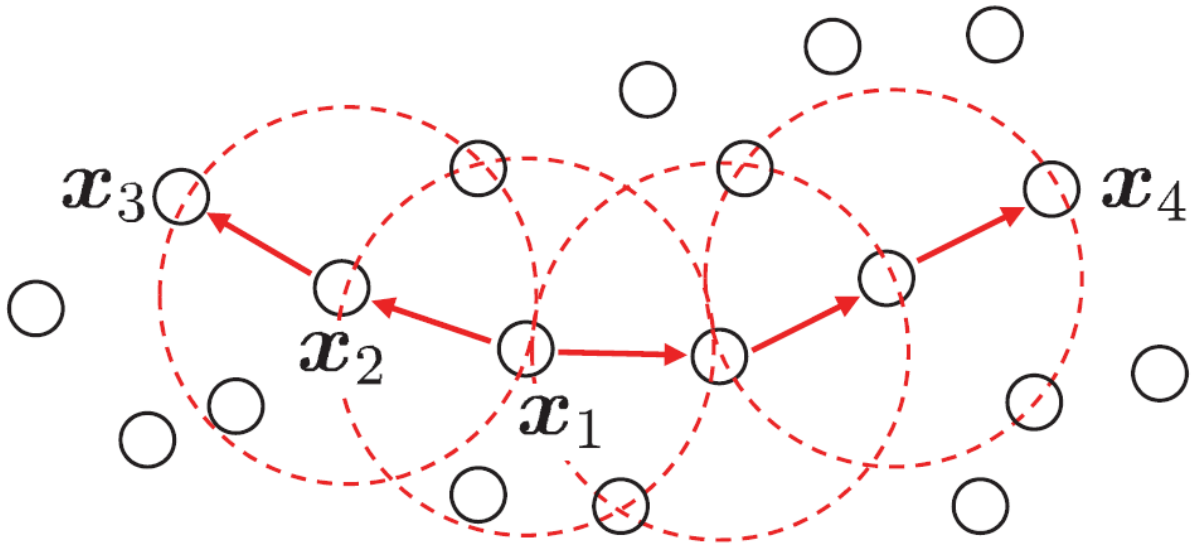


图 9.8 DBSCAN 定义的基本概念($MinPts = 3$): 虚线显示出 ϵ-邻域, x_1 是核心对象, x_2 由 x_1 密度直达, x_3 由 x_1 密度可达, x_3 与 x_4 密度相连.

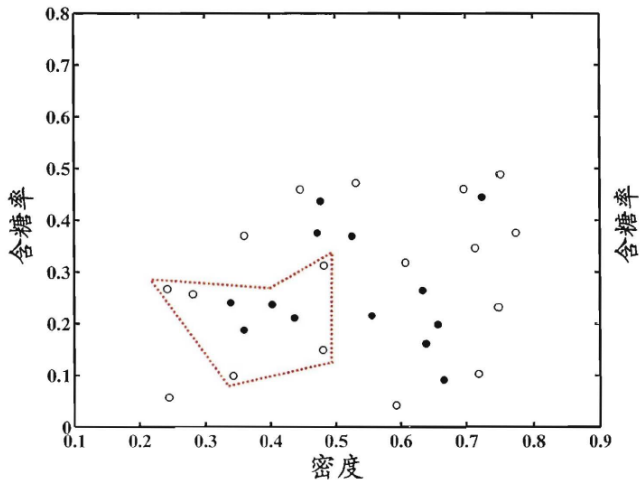
密度聚类

DBSCAN

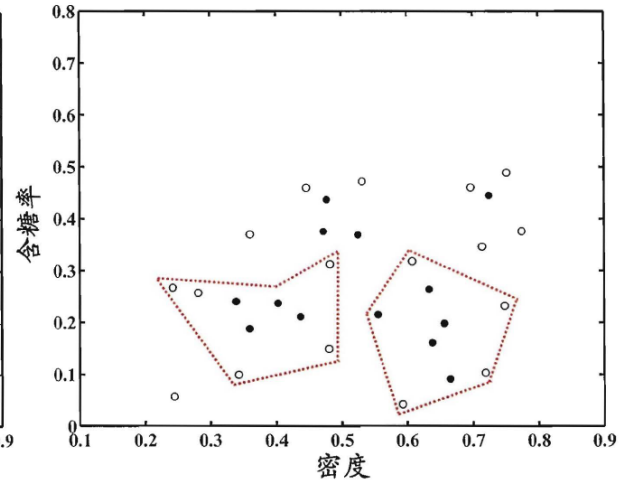
- 基于这些概念，DBSCAN 将“簇”定义为：
- 由密度可达关系导出的最大的密度相连样本集合
- 于是，DBSCAN 算法先任选数据集中的一个核心对象为种子
- 再由此出发确定相应的聚类簇

密度聚类

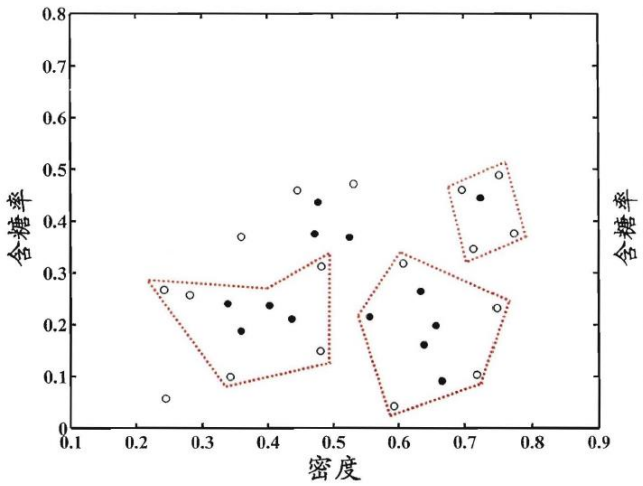
DBSCAN



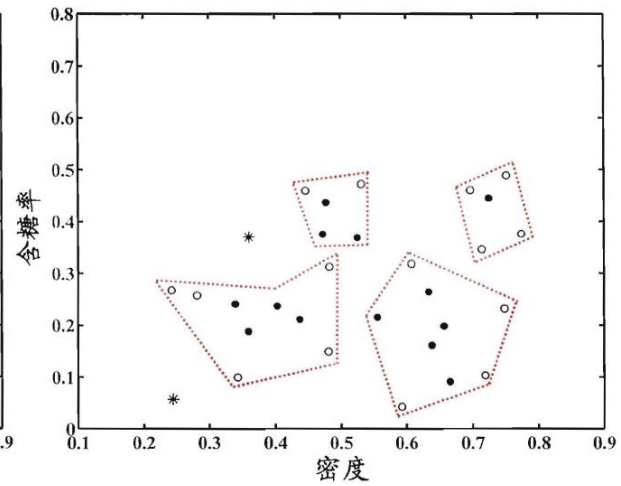
(a) 生成聚类簇 C_1



(b) 生成聚类簇 C_2



(c) 生成聚类簇 C_3



(d) 生成聚类簇 C_4

目录

层次聚类

基本概念

- 层次聚类（Hierarchical Clustering）
- 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

层次聚类

AGNES

- AGNES（Agglomerative Nesting）
- 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

层次聚类

AGNES

- 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：

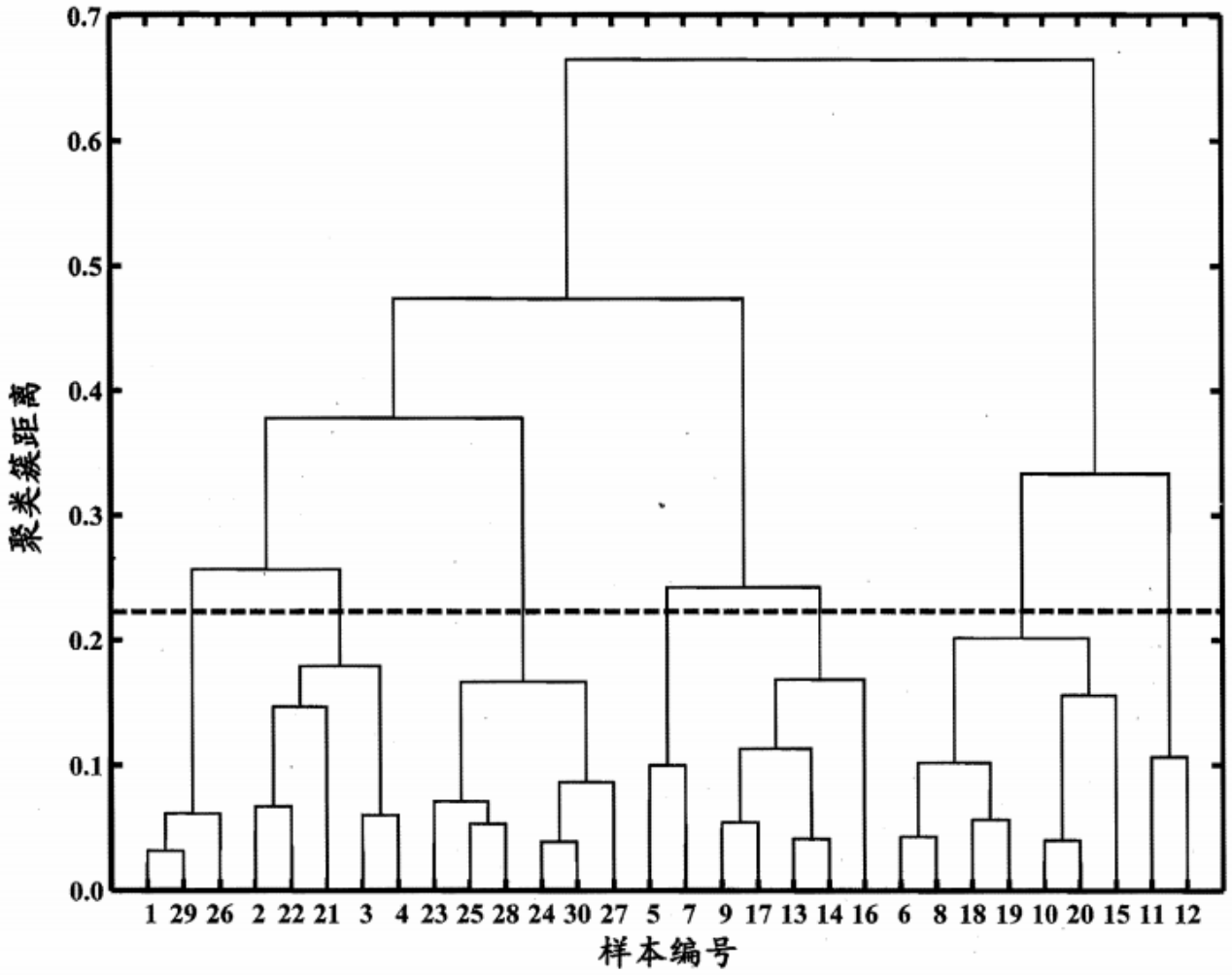
$$\text{最小距离：}\ d_{\min}(C_i, C_j) = \min_{\boldsymbol{x}\in C_i, \boldsymbol{z}\in C_j} \operatorname{dist}(\boldsymbol{x}, \boldsymbol{z}) \ ,$$

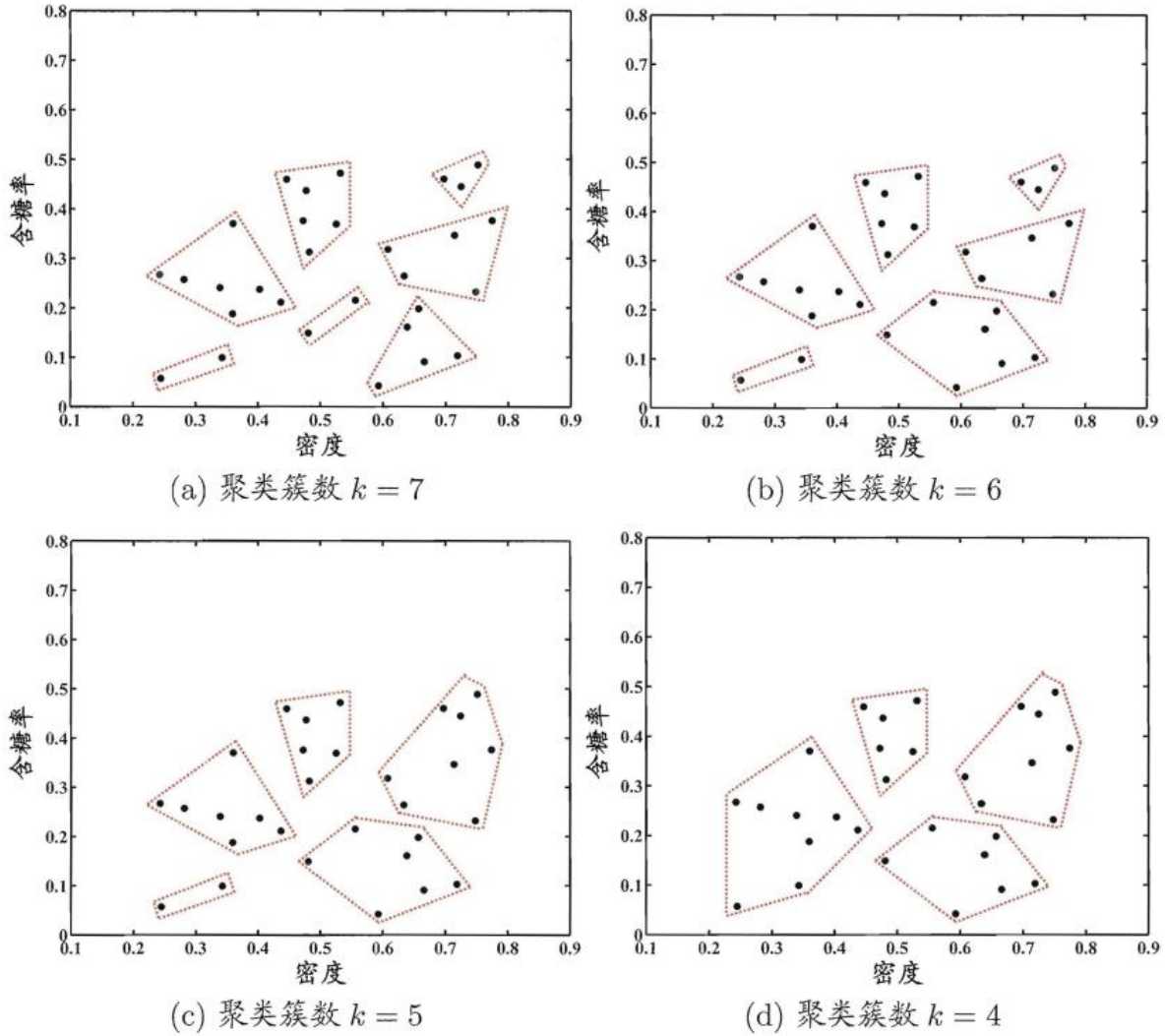
$$\text{最大距离：}\ d_{\max}(C_i, C_j) = \max_{\boldsymbol{x}\in C_i, \boldsymbol{z}\in C_j} \operatorname{dist}(\boldsymbol{x}, \boldsymbol{z}) \ ,$$

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

层次聚类

AGNES


聚类簇距离
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1 29 26 2 22 21 3 4 23 25 28 24 30 27 5 7 9 17 13 14 16 6 8 18 19 10 20 15 11 12
样本编号


含糖率
密度
(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 $k = 4$

总结

- 「西瓜书」周志华《机器学习》，清华大学出版社
- https://item.jd.com/12762673.html

- 「南瓜书」谢文睿、秦州《机器学习公式详解》，人民邮电出版社
- https://github.com/datawhalechina/pumpkin-book/

总结

- Scikit-Learn
- https://scikit-learn.org

- TensorFlow
- https://www.tensorflow.org

東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

Thank You!

丁烨，计算机科学与技术学院
dingye@dgut.edu.cn