智能软件开发
方向基础

第八章 聚类(clustering)
--PART2. 聚类模型

张朝晖

2023.3~6

河北师范大学软件学院
Software College of Hebei Normal University

序号	内容
1	概述
2	机器学习的基本概念
3	模型的选择与性能评价
4	数据的获取、探索与准备
5	近邻模型------分类、回归
6	决策树模型------分类、回归
7	集成学习------分类、回归
8	(朴素)贝叶斯模型------分类
9	聚类
10	特征降维及低维可视化(PCA, t-SNE)
11	总复习

主要内容

1. 动态聚类

   K-均值聚类(K-Means Clustering)

   学习向量量化(LVQ)

2. 密度聚类
3. 层次聚类
4. 高斯混合模型

动态聚类的三个要点：

① 选定某种距离度量作为样本间的相异性度量

② 确定某个准则函数，用于评价聚类结果的质量

③ 给定初始划分方法，以迭代方式找出使准则函数取值最优的划分结果

动态聚类过程：

多次迭代,逐步调整类别划分，最终使准则最优。

- C-Means Clustering
- Fuzzy C-Means Clustering
- …

河北师范大学软件学院
Software College of Hebei Normal University

K-Means Clustering 聚类问题描述

输入： 样本集 $D=\{x_1,...,x_m\}, x_i=[x_{i1} \quad \cdots \quad x_{id}]^T \in R^d$

要生成的簇的数目K

输出： K个互不相交的簇 $C=\{C_l \mid l=1,...,K\}$

1. 聚类准则--"最小误差平方和"准则

将样本集D划分成k簇：$D=C_1 \cup \cdots \cup C_k$

簇的数目k　　样本数目m

误差平方和目标函数　$E(\mu_1,...,\mu_k,C_1,...,C_k)=\sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_i\|^2$

其中　C_i --第i簇，$i=1,...,k$

N_i --第i簇的样本数目

μ_i --第i簇的中心，$\mu_i=\frac{1}{|C_i|}\sum_{x\in C_i}x$

注：$\{\mu_i,\ i=1,...,k\}$ --*codebook*

$\sum_{x\in C_i}\|x-\mu_i\|^2$ -- 簇内样本关于该簇中心的距离平方和

河北师范大学软件学院
Software College of Hebei Normal University

2. 算法实现

初始化方式

交叉迭代的动态更新

$\left\{\boldsymbol{\mu}_i^{(j)},\ \ \boldsymbol{i}=1,\ldots,\boldsymbol{k}\right\}$

$\rightarrow \left\{\boldsymbol{C}_i^{(j+1)},\ \ \boldsymbol{i}=1,\ldots,\boldsymbol{k}\right\}$

$\rightarrow \left\{\boldsymbol{\mu}_i^{(j+1)},\ \ \boldsymbol{i}=1,\ldots,\boldsymbol{k}\right\}$

$\rightarrow \ldots$

算法终止条件

河北师范大学软件学院
Software College of Hebei Normal University

输入: 样本集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$;
聚类簇数 k.

- 初始化方式
- 交叉迭代的动态更新方式
- 迭代终止条件

过程:

1: 从 D 中随机选择 k 个样本作为初始均值向量 $\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_k\}$
2: **repeat**
3: 令 $C_i=\varnothing\ (1\leqslant i\leqslant k)$
4: **for** $j=1,2,\ldots,m$ **do**
5: 计算样本 $\boldsymbol{x}_j$ 与各均值向量 $\boldsymbol{\mu}_i\ (1\leqslant i\leqslant k)$ 的距离: $d_{ji}=\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|_2$;
6: 根据距离最近的均值向量确定 $\boldsymbol{x}_j$ 的簇标记: $\lambda_j=\arg\min_{i\in\{1,2,\ldots,k\}} d_{ji}$;
7: 将样本 $\boldsymbol{x}_j$ 划入相应的簇: $C_{\lambda_j}=C_{\lambda_j}\bigcup\{\boldsymbol{x}_j\}$;
8: **end for**
9: **for** $i=1,2,\ldots,k$ **do**
10: 计算新均值向量: $\boldsymbol{\mu}_i'=\frac{1}{|C_i|}\sum_{\boldsymbol{x}\in C_i}\boldsymbol{x}$;
11: **if** $\boldsymbol{\mu}_i'\neq\boldsymbol{\mu}_i$ **then**
12: 将当前均值向量 $\boldsymbol{\mu}_i$ 更新为 $\boldsymbol{\mu}_i'$
13: **else**
14: 保持当前均值向量不变
15: **end if**
16: **end for**
17: **until** 当前均值向量均未更新

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

A. 聚类中心的初始化

″代表点″选择

样本集**D**划分之前，先选择代表点作为初始聚类核心；
再基于最近距离法，产生各簇
聚类结果与初始代表点选择有关

几种″代表点″的选择方法：

--经验选择

--″密度法″选择代表点

--随机选择代表点

河北师范大学软件学院
Software College of Hebei Normal University

K-Means ++ 中更为有效的聚类中心初始化算法

1.聚类中心集合M的初始化：$M \leftarrow \Phi$

2.在样本集D内随机选择1个样本初始化聚类中心 μ_1 ：

$M \leftarrow M \cup \{\mu_1\}$

3. ***for*** $i = 2,...,K$ ***do***

3.1 对于样本集D内不同于M内$(i-1)$个中心的每个样本x，计算该样本关于M内$(i-1)$个中心的距离平方最小值，记为$\left[d(x,M)\right]^2 = \min\limits_{j=1,...,i-1}\left[d(x,\mu_j)\right]^2$

3.2 按照概率 $\dfrac{\left[d(x^*,M)\right]^2}{\sum\limits_{x_i \in D \setminus M}\left[d(x_i,M)\right]^2}$ 随机选择一个样本x^*，将其作为初选的第i个聚类中心μ_i

3.3 $M \leftarrow M \cup \{\mu_i\}$

4.返回M

B. 簇的数目k的确定

通常要求事先给定**“簇”数目k**.

若k未知，可按如下方法确定

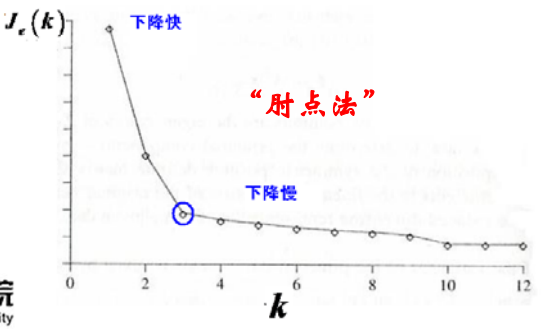
(a)一般根据领域先验知识确定；

(b)实验确定：

令$k=1,2,3,....,$分别进行聚类，得$J_e(k)$,

绘制$J_e(k)$-k曲线图；

找出拐点，对应聚类数目为最终类别数。



该方法并不总是有效。

C. 其它

是否需要进行样本集的规范化预处理？--是

是否需要重复多次？

--从多个聚类结果中选择最好的那个

任意形状的聚类都可处理吗？--不

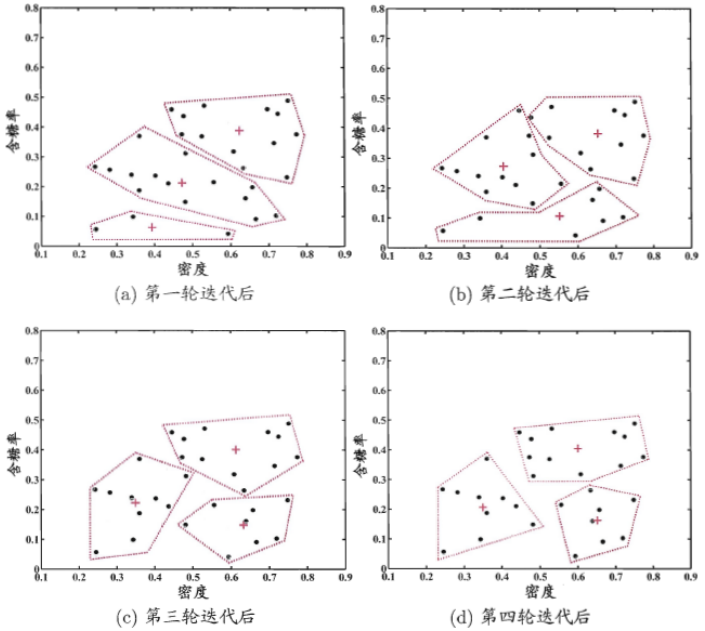


图 9.3 西瓜数据集 4.0 上 k 均值算法($k = 3$)在各轮迭代后的结果. 样本点与均值向量分别用"•"与"+"表示, 红色虚线显示出簇划分.

主要内容

1. 动态聚类

   K-均值聚类(K-Means Clustering)

   学习向量量化(LVQ)

2. 密度聚类
3. 层次聚类
4. 高斯混合模型

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

问题描述

输入： 样本集 $D=\{(x_1,y_1),...,(x_m,y_m)\}$

要生成的原型向量数目 q

其中 $x_i=[x_{i1} \quad \cdots \quad x_{id}]^T \in X \subset R^d$

$y_i \in Y$

输出：

(1) q 个原型向量 $\{p_l \mid l=1,...,q\}$，

每个原型向量代表一个聚类簇，簇的标记值 $t_l \in Y$

(2) 基于学得的原型向量，对原始空间进行划分

第一阶段 基于有标签样本集的原型向量的学习

--找特征空间的代表点

注意：原型向量数目应不低于标签值的数目

输入： 样本集 $D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$；
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：

1: 初始化一组原型向量 $\{p_1,p_2,\ldots,p_q\}$
2: **repeat**
3: 从样本集 D 随机选取样本 (x_j,y_j)；
4: 计算样本 x_j 与 p_i $(1 \leqslant i \leqslant q)$ 的距离：$d_{ji}=\|x_j-p_i\|_2$；
5: 找出与 x_j 距离最近的原型向量 p_{i^*}，$i^*=\arg\min_{i\in\{1,2,\ldots,q\}} d_{ji}$；
6: **if** $y_j=t_{i^*}$ **then**
7: $p'=p_{i^*}+\eta\cdot(x_j-p_{i^*})$
8: **else**
9: $p'=p_{i^*}-\eta\cdot(x_j-p_{i^*})$
10: **end if**
11: 将原型向量 p_{i^*} 更新为 p'
12: **until** 满足停止条件

比如：最大迭代次数

输出： 原型向量 $\{p_1,p_2,\ldots,p_q\}$

若p_{i*}与x_j**类别一致**，则p_{i*}更新后为 $p'=p_{i*}+\eta\left(x_j-p_{i*}\right)$

$$\left\|p'-x_j\right\|_2=\left\|p_{i*}+\eta\left(x_j-p_{i*}\right)-x_j\right\|_2$$

$$=\left\|(1-\eta)p_{i*}-(1-\eta)x_j\right\|_2$$

$$=(1-\eta)\left\|p_{i*}-x_j\right\|_2<\left\|p_{i*}-x_j\right\|_2$$

若p_{i*}与x_j**类别不一致**，则p_{i*}更新后为 $p'=p_{i*}-\eta\left(x_j-p_{i*}\right)$

$$\left\|p'-x_j\right\|_2=\left\|p_{i*}-\eta\left(x_j-p_{i*}\right)-x_j\right\|_2$$

$$=\left\|(1+\eta)p_{i*}-(1+\eta)x_j\right\|_2$$

$$=(1+\eta)\left\|p_{i*}-x_j\right\|_2>\left\|p_{i*}-x_j\right\|_2$$

第二阶段，基于原型向量的样本空间X的簇划分。
--Voronoi Tessellation(V图剖分)

q个原型向量$\{p_l \mid l=1,\ldots,q\}$

$$\forall x\in X\subset R^d$$

$$R_i=\left\{x\in X \mid \left\|x-p_i\right\|_2\le\left\|x-p_{i'}\right\|_2, i'\ne i\right\}$$

$$X=R_1\cup R_2\cup\cdots\cup R_q$$

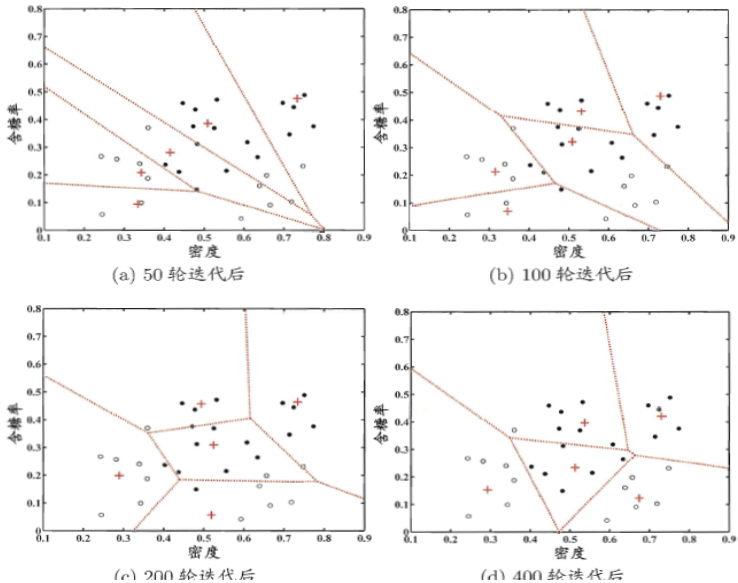


图 9.5 西瓜数据集 4.0 上 LVQ 算法($q = 5$)在不同轮数迭代后的聚类结果. c_1, c_2 类样本点与原型向量分别用"•", "○"与"+"表示, 红色虚线显示出聚类形成的 Voronoi 剖分.

主要内容

1. 动态聚类
2. 密度聚类(DBSCAN)

Density - Based Spatial Clustering of Applications with Noise

3. 层次聚类
4. 高斯混合模型

- DBSCAN 算法是一种**基于高密度连通区域**的、基于密度的聚类算法
- 能够将具有足够高密度的区域划分为簇
- 能够在具有噪声的数据中发现**任意形状的簇**
- 能够发现**异常点**

1. 有关概念[给定样本集$\boldsymbol{D}=\{x_1,\cdots,x_m\}$]

[1]两个**全局邻域参数**$(\varepsilon, \boldsymbol{MinPts})$

ε--邻域最大半径

$\boldsymbol{MinPts}$--给定样本的$\varepsilon-$**邻域**内最小样本数.

[2]$\varepsilon-$**邻域**

对于$\forall \boldsymbol{x}_j \in \boldsymbol{D}$, $\boldsymbol{x}_j$的$\varepsilon-$邻域为$\boldsymbol{N}_\varepsilon(\boldsymbol{x}_j)=\{\boldsymbol{x}_i \in \boldsymbol{D} \mid \boldsymbol{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)\le\varepsilon\}$

[3]**核心对象**$(core\ \ object)$

若$|\boldsymbol{N}_\varepsilon(\boldsymbol{x}_j)| \ge \boldsymbol{MinPts}$, 则称$\boldsymbol{x}_j$为一个**核心对象**.

[4]**密度直达**$(\boldsymbol{directly\ \ density-reacheable})$

若$\boldsymbol{x}_j \in \boldsymbol{N}_\varepsilon(\boldsymbol{x}_i)$,并且$\boldsymbol{x}_i$为一个核心对象，则称$\boldsymbol{x}_j$为由$\boldsymbol{x}_i$**密度直达**.

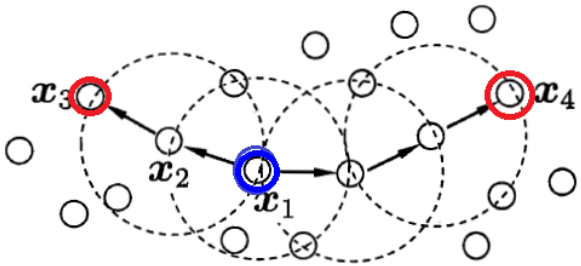
[5]**密度可达**$(\boldsymbol{density-reacheable})$

对于$\boldsymbol{x}_i,\boldsymbol{x}_j$，若存在样本序列$\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_n$,其中$\boldsymbol{p}_1=\boldsymbol{x}_i,\boldsymbol{p}_n=\boldsymbol{x}_j$,且$\boldsymbol{p}_{i+1}$由$\boldsymbol{p}_i$密度直达，则称$\boldsymbol{x}_j$由$\boldsymbol{x}_i$**密度可达**.

[6]**密度相连**(*density – connected*)

对于x_i与x_j，若存在样本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**.

邻域参数$(\varepsilon, MinPts)$，ε–邻域，**核心对象**，**密度直达**，**密度可达**，**密度相连**



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.

[7]**边界对象**(*border object*)

若$N_\varepsilon(x_i) \geq MinPts$，$x_j \in N_\varepsilon(x_i)$，并且$N_\varepsilon(x_j) < MinPts$，则称$x_j$为一个边界对象.

边界对象位于聚类簇的边界处.

[8]**噪声对象**(*noise object*)

核心对象、边界对象以外的其它样本.

[9]**簇**

由**密度可达**关系导出的**最大密度相连样本集**。

给定邻域参数$(\varepsilon, MinPts)$，簇$C \subseteq D$为满足以下性质的非空样本子集：

连接性(*connectivity*)：$x_i, x_j \in C \Rightarrow x_i$与$x_j$**密度相连**.

最大性(*maximality*)：$x_i \in C$，并且x_j由x_i**密度可达** $\Rightarrow x_j \in C$.

样本集***D***的主要组成

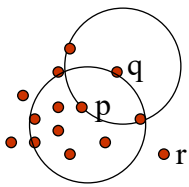
- **核心对象**（聚类簇的种子点）
- **边界对象**（可由核心对象密度可达的非核心对象）

} 聚类簇

- **噪声对象**（异常点，孤立点）

高密度区

低密度区



MinPts = 5

Eps = 1 cm

2. DBSCAN算法纲要

特点

--无需指定聚类簇的数目；

--只需输入两个全局参数(**全局参数取值如何选？**).

启发

--识别样本集内的核心对象；

--任选一个核心对象，作为一个聚类簇的种子点；

--获取该种子点及其所有密度可达样本，构成一个聚类簇.

河北师范大学软件学院
Software College of Hebei Normal University

➢ 遍历整个样本集D，确定核心对象、边界对象、及噪声对象

➢ 将所有噪声对象标记为异常.

➢ 考查每个核心对象：核心对象彼此可密度直达，则应划分至相同的聚类簇。

➢ 所有边界对象：若它们是密度相连的，应划分至与其核心对象一致的聚类簇内.

2. 算法描述

STEP1.识别给定样本集D的所有核心对象.
得到核心对象集合 Ω

STEP2. 初始化聚类簇的数目为0；初始化未被访问的样本集为整个数据集D.

STEP3. 重复如下过程，生成一系列聚类簇,直到核心对象集合为空.
➢从核心对象集合中，任选1核心对象，作为聚类簇的一个种子点，找出其密度可达的所有样本，构成1个聚类簇.
➢ 更新核心对象集合；
➢ 更新未访问的样本集合

STEP4. 输出所有聚类簇.

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: Ω = ∅
for j = 1,2,...,m do
    确定样本 x_j 的 ε-邻域 N_ε(x_j);
    if |N_ε(x_j)| ≥ MinPts then
        将样本 x_j 加入核心对象集合: Ω = Ω ∪ {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: Γ = D
while Ω ≠ ∅ do
    记录当前未访问样本集合: Γ_old = Γ;
    随机选取一个核心对象 o ∈ Ω, 初始化队列 Q =< o >;
    Γ = Γ \ {o};          更新未访问的样本集合
    while Q ≠ ∅ do
        取出队列 Q 中的首个样本 q;
        if |N_ε(q)| ≥ MinPts then
            令 Δ = N_ε(q) ∩ Γ;
            将 Δ 中的样本加入队列 Q;
            Γ = Γ \ Δ;        更新未访问的样本集合
        end if
    end while
    k = k + 1, 生成聚类簇 C_k = Γ_old \ Γ;
    Ω = Ω \ C_k          更新核心对象集合
end while
```

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

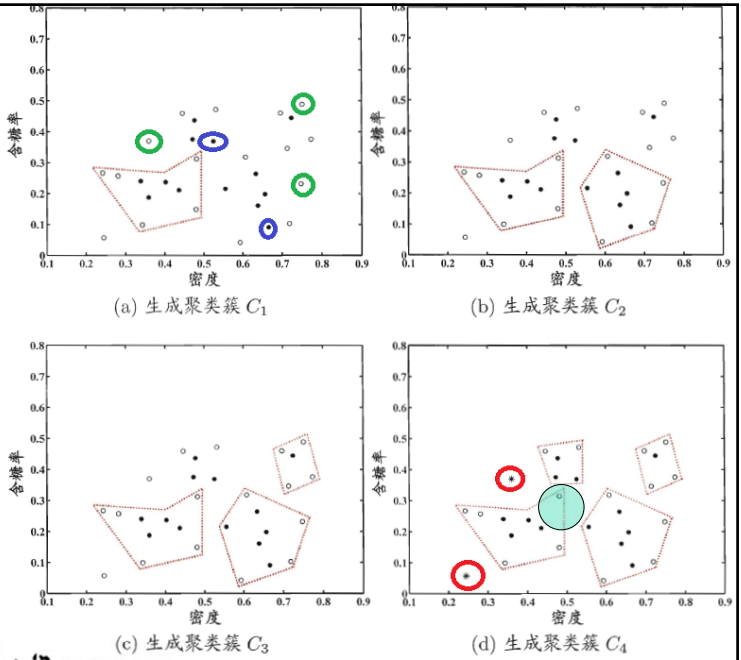
DBSCAN算法的聚类结果

核心对象
非核心对象

密度直达
密度可达
密度相连

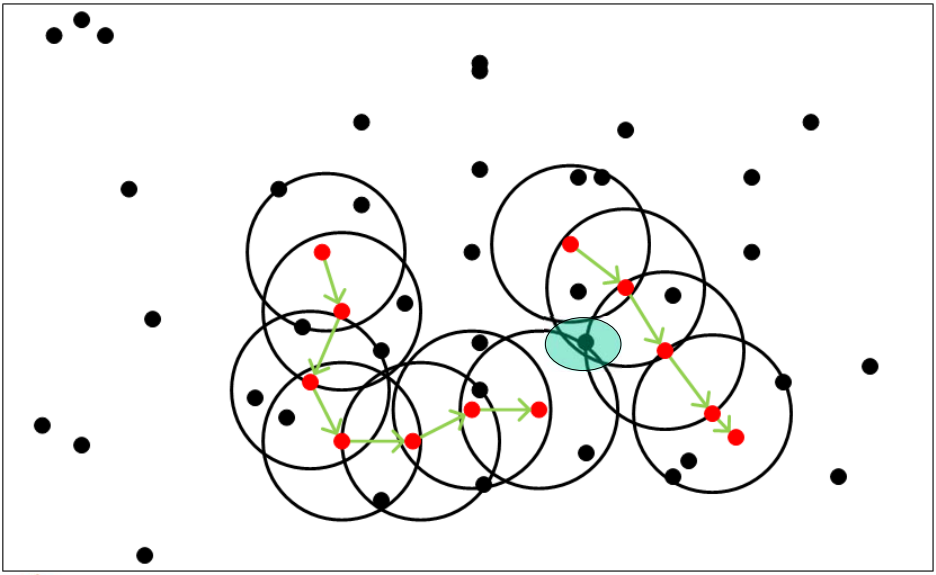
聚类簇

噪声或异常点



(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



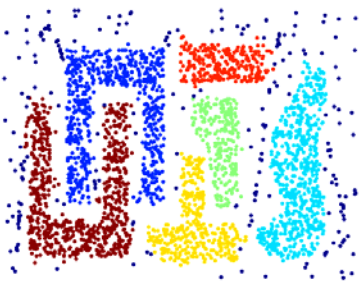
河北师范大学软件学院
Software College of Hebei Normal University

DBSCAN聚类算法的细节

河北师范大学软件学院
Software College of Hebei Normal University

DBSCAN运行效果好的时候

Original Points



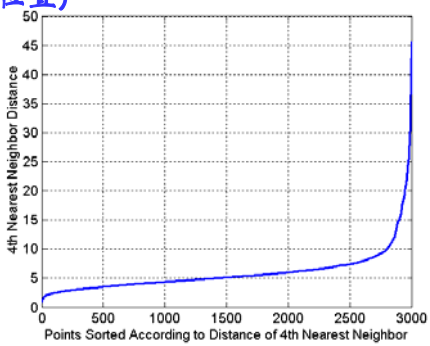
Clusters

- 对噪声不敏感
- 可处理不同形状和大小的数据

河北师范大学软件学院
Software College of Hebei Normal University

如何适当选取EPS和MinPts
--基于k-距离

- 位于同一聚类簇的所有样本点，它们到其第k个最近邻的距离应大致一样
- 噪声点到其第k个最近邻的距离比较远
- 获取每个样本到其第k个最近邻的距离，从小到大升序排列得到k-距离变化曲线图
- 找到曲线的拐点(变化剧烈的位置)



- 然后：

Eps即为变化剧烈位置对应的K-距离.

MinPts取k

DBSCAN算法的优缺点

- 优点

基于密度定义，相对抗噪音；
能处理任意形状和大小的簇

- 缺点

密度分布不均匀、或密度变化较大的簇时，会有麻烦；
邻域半径小，数量多的小簇，核心点数量减小
邻域半径大，本不属于同一簇的样本会聚至相同簇

对于高维问题，密度定义是个比较麻烦的问题

主要内容

1. 聚类的引入
2. 动态聚类
3. 密度聚类(density-based clustering) DBSCAN
4. 层次聚类

hierarchical clustering

也称系统聚类、分级聚类

1.层次聚类的引入

(1)问题描述

数据集$\boldsymbol{D}=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m\}\Rightarrow$划分为合理的$\boldsymbol{k}$个聚类簇。

$\boldsymbol{1}\le\boldsymbol{k}\le\boldsymbol{m}$

$\boldsymbol{k}$值的极端情况$\begin{cases}\text{最多}\boldsymbol{m}\text{簇，每簇包含一个样本}\\ \text{最少}1\text{簇，所有样本同属一簇}\end{cases}$

(2)**层次聚类的实现方式**

样本数据的递归聚类；聚类过程中逐级考察簇间的距离，以此决定类别数.

合并式(聚合式)聚类(***agglomerative clustering***)

从聚类簇数目最多开始，**自底向上**，逐级**合并**距离最近的两个聚类簇，聚类簇的数目**逐渐减少**,直至与预设的聚类簇数目一致. **经常使用.**

如：***AGNES***算法(***AGglomerative NESting***)

分裂式聚类(***divisive clustering***)

从聚类簇数目最少开始，**自顶向下**，逐级**分裂**每级中最松散的聚类簇，聚类簇的数目逐级增加，直至与预设聚类簇数目一致.

如：***TSVQ***,可用于码本(***codebook***)的生成

(3)**层次聚类的几个关键问题**

相似性(相异性)度量

聚类簇之间；

簇内样本之间

聚类簇合并 / 分裂停止的条件

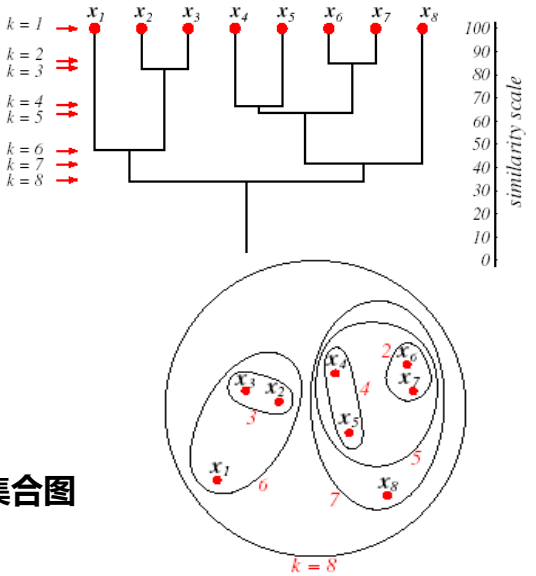
距离阈值；

预设的聚类簇数目

计算复杂度

(4)**聚合式系统聚类结果的表示**

A.树状图(*dendrogram*),**聚类树**



可定量表示聚类簇间的相似性(或相异性),簇间相似性标尺随着层数增加，逐渐减小。

B.维恩图(*venn diagram*),**集合图**

只能**定性表示类间相似性**

2. 聚合式系统聚类算法

给定样本集$\boldsymbol{D}=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m\}$,选定簇间距离度量$\boldsymbol{d}(\boldsymbol{C}_i,\boldsymbol{C}_j)$,将样本$\boldsymbol{D}$聚成$\boldsymbol{k}$簇.

不同的距离度量，可产生不同的聚类结果

***STEP1* 初始化.**

A. 指定最终类别数k;

B. 初始划分
- 每个样本自成1簇　$\boldsymbol{C}_i \Leftarrow \{\boldsymbol{x}_i\}$　$\boldsymbol{i}=1,\ldots,\boldsymbol{m}$
- 初始类别数　$\boldsymbol{q} \Leftarrow \boldsymbol{m}$

***STEP2* 合并.** 重复以下工作，直到$\boldsymbol{q}=\boldsymbol{k}$.

A. 寻找最相似(距离最小)的两簇$C_i,C_j\ (i<j)$

$$\boldsymbol{d}(\boldsymbol{C}_i,\boldsymbol{C}_j)=\min_{\substack{\forall \boldsymbol{l},\boldsymbol{m}\in\{1,\ldots,\boldsymbol{q}\}\\ \text{并且}\boldsymbol{l}<\boldsymbol{m}}}\mathbf{d}(\boldsymbol{C}_l,\boldsymbol{C}_m)$$

B. 记录最小距离，合并两类C_i,C_j: $\boldsymbol{q} \Leftarrow \boldsymbol{q}-1$

***STEP3* 输出k个聚类簇.**

AGNES算法描述

STEP1.初始化聚类簇

STEP2.初始化距离矩阵

STEP3.初始化聚类簇的数目

STEP4.逐级合并最相似的两簇，直到满足规定的聚类簇数目：
(1) 合并最相似的两簇；
(2) 更新相关簇的序号；
(3) 更新相关距离矩阵；
(4) 更新簇的数目

STEP5.输出结果。

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 $\mathcal{C}=\{C_1,C_2,\ldots,C_k\}$

3.聚类簇C_i,C_j之间的连接（*linkage*）

(1)**最小距离**

$$d_{\min}(C_i,C_j)=\min_{\substack{\boldsymbol{x}\in C_i\\ \boldsymbol{z}\in C_j}} dist(\boldsymbol{x},\boldsymbol{z})$$

相应聚类算法就是"单链接"算法（***single linkage*** 或 ***single – link***）

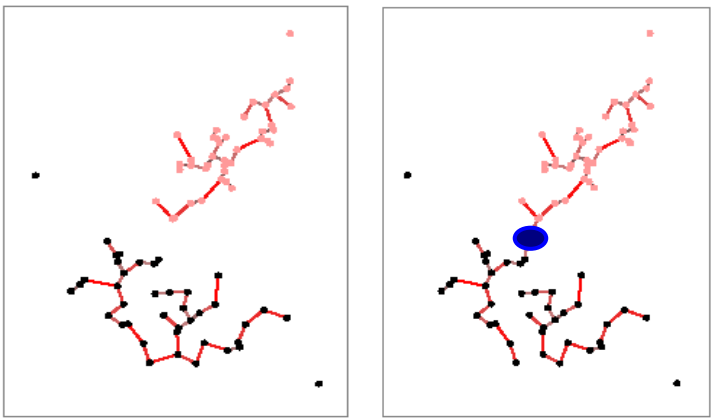
采用该距离度量**聚类簇C_i,C_j之间**的相异程度，进行聚类，就是产生**最小生成树**（***minimal spanning tree***）

缺陷　链接效应，产生细长的聚类；
聚类结果对噪声或数据点波动敏感。

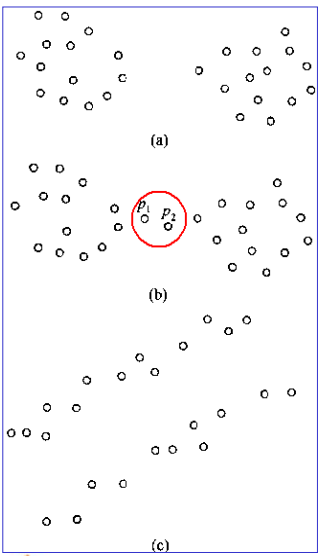
图. 二维高斯样本最小距离法层次聚类

左图 无干扰点

右图 有干扰点



最小距离法层次聚类(*single linkage* 或 *single – link*)



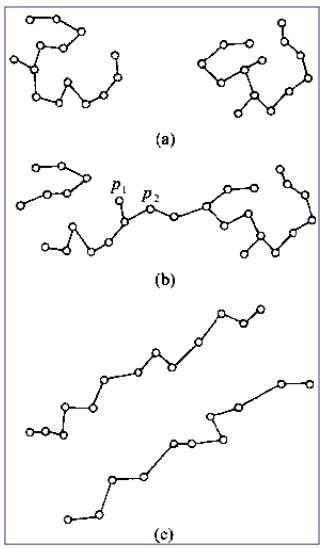
左图：

三种数据分布；

右图：

最近距离法的聚类结果

注意：a,b的区别



(2)**最大距离**

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

最远邻算法

(*the farthest-neighbor clustering algorithm*)

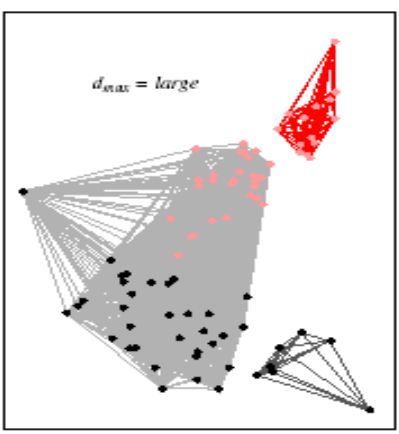
全连接算法

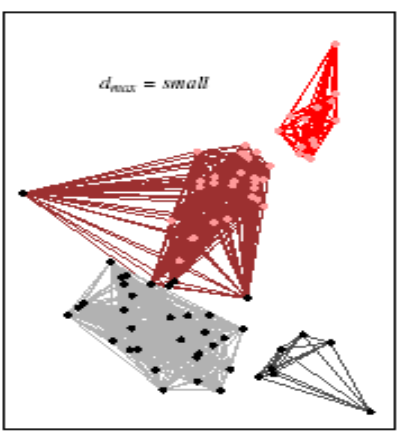
(*complete-linkage algorithm*)

若$d_{\max}(C_i, C_j) \geq d_0$,则聚类过程结束。

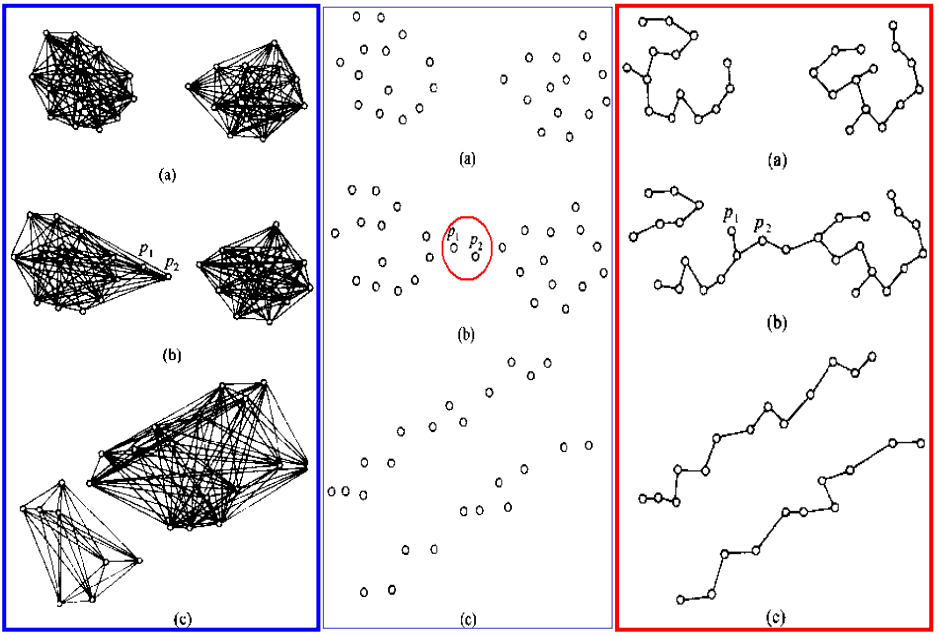
特点
- 防止两个密集点集通过某个路径聚为一簇的可能
- 不能检测出具有长条形状的聚类
- 适合紧密、体积相近的类别划分
- 聚类结果对个别远离点敏感

最大距离阈值d_0越大，聚类簇的数目越小。





最大距离法与最小距离法聚类结果比较



(3)平均距离

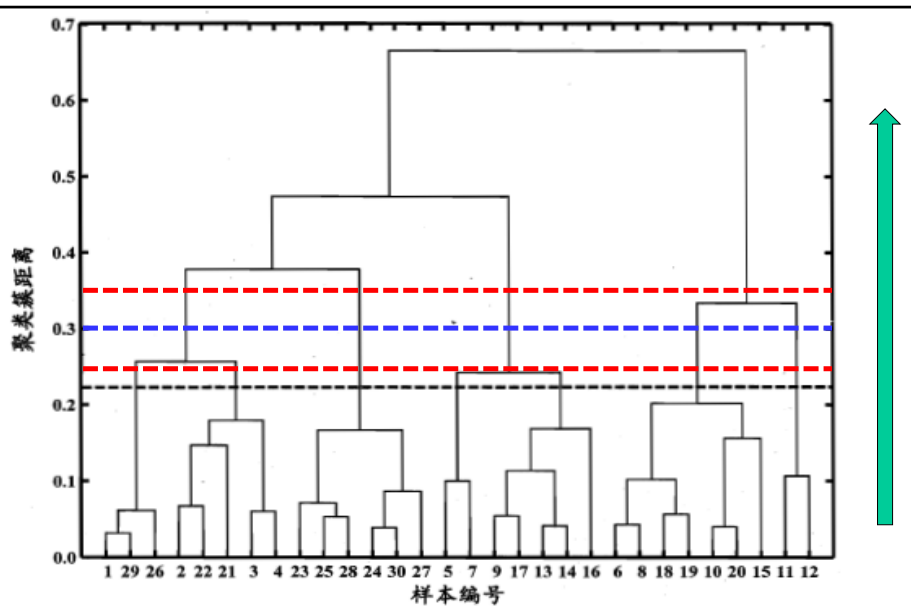
$$d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{z \in C_j} dist(x, z)$$

相应的聚类算法称为"均链接"算法 $(average-link, average\ \ linkage)$.

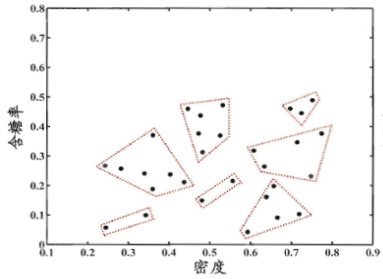
是关于前两种聚类算法的折中；计算简单。

西瓜数据集4.0

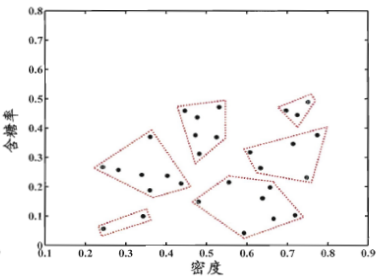
编号	密度	含糖率	编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	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



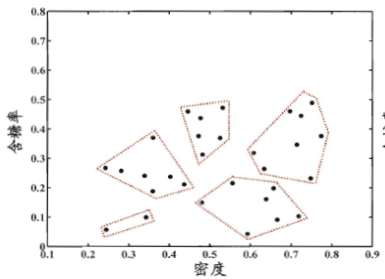
基于西瓜数据集4.0，采用**聚合式层次聚类**，得到的聚类树状图，簇之间的相异性度量采用**最大距离法**. **逐步提升分割层，聚类数目逐渐减小**。



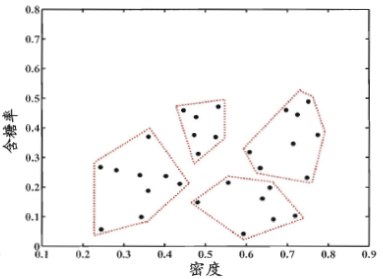
(a) 聚类簇数 $k=7$



(b) 聚类簇数 $k=6$



(c) 聚类簇数 $k=5$



(d) 聚类簇数 $k=4$

思考题

1. 什么是聚类？什么是分类？
   请给出二者的区别与联系.

2. 若采用不同模型对给定的数据集D进行划分.请给出不同聚类算法的实现步骤。

(1) k-均值聚类 (目标函数形式、参数意义)
(2)层次聚类(如何计算样本点之间距离、集合之间距离)
(3)DBSCAN聚类(几个术语、控制参数的意义)

3. 上述聚类模型的适用场合.

河北师范大学软件学院
Software College of Hebei Normal University