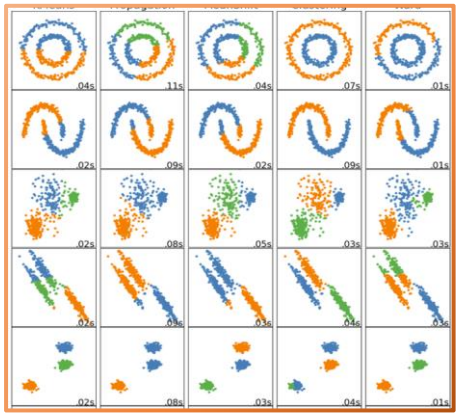


第十讲　　聚类分析

云南大学

孙正宝

zbsun@ynu.edu.cn

提　纲

一　聚类方法

二　主成分分析（PCA）

聚类方法

聚类

- 聚类是针对给定的样本集合，依据他们特征的相似度或距离，将其归并到若干个“类”或“簇”的过程。
- 一个类是给定样本集合的一个子集。直观上，相似的样本聚到相同的类，不相似的样本分散在不同的类。
- 聚类的目的是通过得到的类或簇来发现数据的特点或者对数据进行处理，在数据挖掘、模式识别等领域有着广泛的应用。
- 因为类或簇事先并不知道，因此聚类属于无监督学习 。

聚类方法

聚类

- 聚类的对象是观测数据，或样本集合。假设有n个样本，每个样本由m个属性的特征向量组成，样本合集可以用矩阵$\mathbf{X}$表示

$$X = [x_{ij}]_{m\times n} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

- 聚类的核心概念是相似度（similarity）或距离（distance)的度量。

1. 距离或相似度

1.1 闵可夫斯基距离

样本x_i与样本x_j 的闵可夫斯基距离（Minkowski distance）定义为

$$d_{ij}=\left(\sum_{k=1}^{m}|x_{ki}-x_{kj}|^{p}\right)^{\frac{1}{p}} \qquad p\geqslant 1$$

- 闵可夫斯基距离越大相似度越小，距离越小相似度越大。

1. 距离或相似度

1.1 闵可夫斯基距离

当 $p = 2$ 时，为欧氏距离

$$d_{ij} = \sqrt{\sum_{k=1}^{m} |x_{ki} - x_{kj}|^2}$$

当 $p = 1$ 时，为曼哈顿距离

$$d_{ij} = \sum_{k=1}^{m} |x_{ki} - x_{kj}|$$

当 $p = \infty$ 时，取各个坐标数之差的绝对值的最大值，称为切比雪夫距离

$$d_{ij} = \max_{k} |x_{ki} - x_{kj}|$$

1. 距离或相似度

1.2 马哈拉诺比斯距离

马哈拉诺比斯距离（Mahalanobis distance)，简称马氏距离，也是另一种常用的相似度，考虑各个分量（特征）之间的相关性。样本x_i与样本x_j 的马氏距离定义为

$$d_{ij} = \sqrt{(x_i - x_j)^T S^{-1}(x_i - x_j)}$$

- 马哈拉诺比斯距离越大相似度越小，距离越小相似度越大。

1. 距离或相似度

1.3 相关系数

样本x_i与样本x_j的相关系数定义为

$$r_{ij}=\frac{\sum_{k=1}^{m}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j)}{\sqrt{\sum_{k=1}^{m}(x_{ki}-\bar{x}_i)^2\sum_{k=1}^{m}(x_{kj}-\bar{x}_j)^2}}$$

$$\bar{x}_i=\frac{1}{m}\sum_{k=1}^{m}x_{ki},\quad \bar{x}_j=\frac{1}{m}\sum_{k=1}^{m}x_{kj}$$

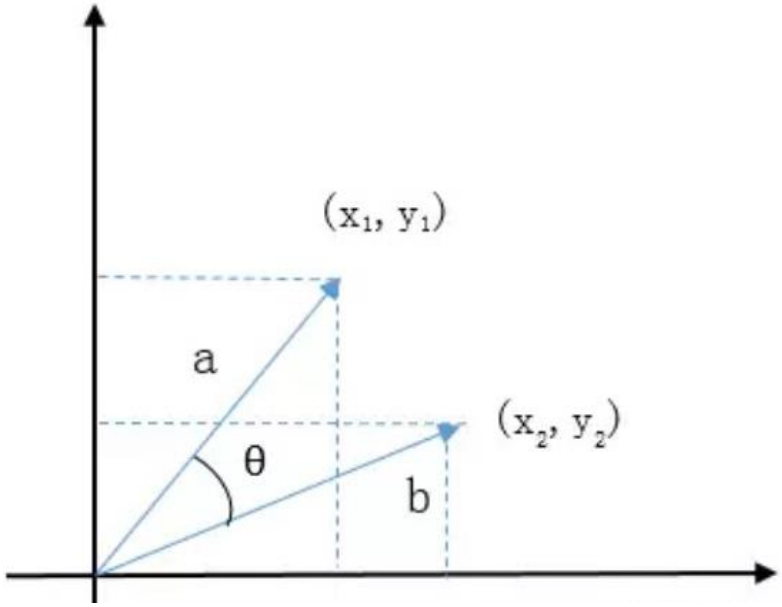
- 相关系数的绝对值越接近于1，表示样本越相似。

1. 距离或相似度

1.4 余弦相似/夹角余弦

样本x_i与样本x_j的夹角余弦定义为

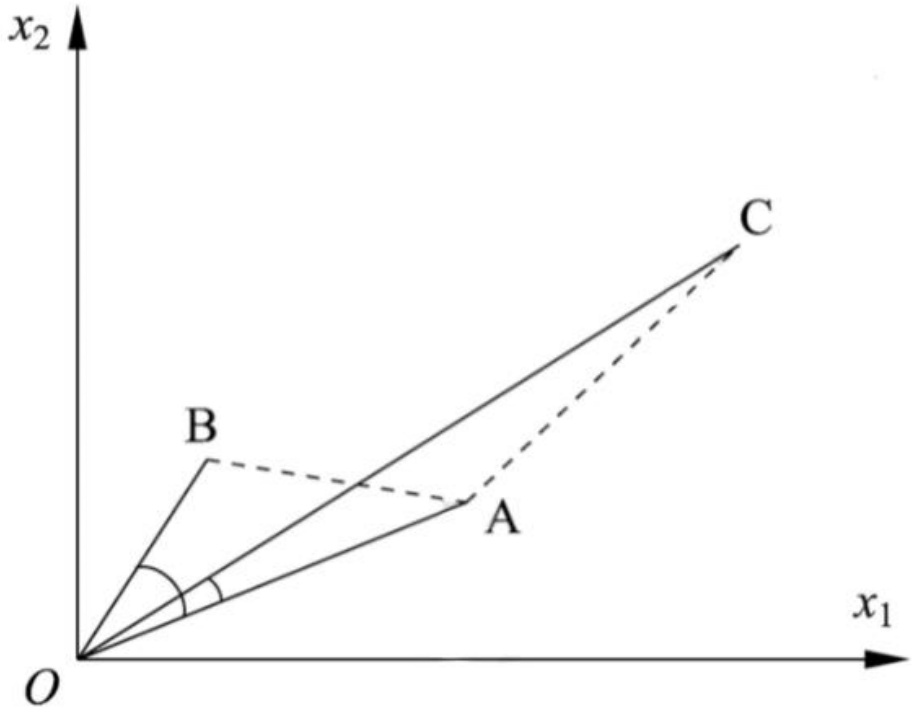
$$s_{ij} = \frac{\sum_{k=1}^{m} x_{ki}x_{kj}}{\left[\sum_{k=1}^{m} x_{ki}^2 \sum_{k=1}^{m} x_{kj}^2\right]^{\frac{1}{2}}}$$



- 夹角余弦越接近于1，表示样本越相似；越接近于0，表示样本越不相似。

1. 距离或相似度

需要注意的是，不同相似度度量得到的结果并不一定一致。



2. 簇（类）

2.1 簇的定义

通过聚类得到的簇，本质上是样本的子集。

若集合$\boldsymbol{G}$中存在样本x_i与样本x_j之间的距离满足下式

$$d_{ij} \ll T$$

则称$\boldsymbol{G}$为一个簇（类），$\boldsymbol{T}$为任意正数。

2. 簇（类）

2.2 簇中心

簇中心又称簇平均值，定义为

$$\bar{x}_G = \frac{1}{n_G} \sum_{i=1}^{n_G} x_i$$

其中，n_G 为类 G 中包含的样本数。

2. 簇（类）

2.3 簇直径

簇（类）的直径$\boldsymbol{D_G}$定义为簇（类）中任意两个样本之间的最大距离，即

$$D_G = \max_{x_i, x_j \in G} d_{ij}$$

2. 簇（类）

2.4 簇间距离

- 中心距离

$$D_{pq} = d_{\bar{x}_p \bar{x}_q}$$

- 最短距离（单连接）

$$D_{pq} = \min\{d_{ij} | x_i \in G_p, x_j \in G_q\}$$

- 最长距离（全连接）

$$D_{pq} = \max\{d_{ij} | x_i \in G_p, x_j \in G_q\}$$

- 平均距离

$$D_{pq} = \frac{1}{n_p n_q} \sum_{x_i \in G_p} \sum_{x_j \in G_q} d_{ij}$$

3. *k*-means聚类算法

3.1 算法思想

- 给定n个样本的集合 $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$
- 每个样本由一个特征向量表示，特征向量的维数是m。
- k均值聚类的目标是将n个样本分到k个不同的类或簇中，这里假设$k<n$。
- k个类$G_1,G_2,\ldots,G_k$形成对样本集合X的划分，其中

$$G_i \cap G_j = \phi, \bigcup_{i=1}^{k} G_i = \mathbf{X}$$

- 用C表示划分，一个划分对应着一个聚类结果。

3. *k*-means聚类算法

3.2 算法步骤

首先，采用欧氏距离平方(squared Euclidean distance)作为样本之间的距离

$$d^2(x_i, x_j) = \sum_{k=1}^{m} (x_{ki} - x_{kj})^2 = \|x_i - x_j\|^2$$

3. *k*-means聚类算法

3.2 算法步骤

然后，定义样本与其所属类的中心之间的距离的总和为损失函数，即

$$W(C) = \sum_{l=1}^{k} \sum_{C(i)=l} \|x_i - \bar{x}_l\|^2$$

其中，$\bar{x}_l = (\bar{x}_{1l}, \bar{x}_{2l}, \cdots, \bar{x}_{ml},)^T$ 是第 l 个类的均值或中心。

3. *k*-means聚类算法

3.2 算法步骤

k-means聚类就是求解最优化问题

$$C^* = arg\min_C = \sum_{l=1}^{k} \sum_{C(i)=l} ||x_i - \bar{x}_l||^2$$

*k*确定的情况下，对于给定的聚类中心目标函数可以重写为

$$C^* = \min_{m_1, m_2, \cdots, m_k} = \sum_{l=1}^{k} \sum_{C(i)=l} ||x_i - m_l||^2$$

3. *k*-means聚类算法

3.2 算法步骤

对于每个包含n_l个样本的类$\boldsymbol{G_l}$，其更新均值为，即

$$m_l = \frac{1}{n_l} \sum_{C(i)=l} x_i, \quad l = 1, 2, \cdots, k$$

重复以上步骤，直到收敛为止。

3. *k*-means聚类算法

3.3 例题

给定含有5个样本的数据集如下

$$X = \begin{bmatrix} 0 & 0 & 1 & 5 & 5 \\ 2 & 0 & 0 & 0 & 2 \end{bmatrix}$$

使用*k*-means算法将其聚到2个簇中。

3. *k*-means聚类算法

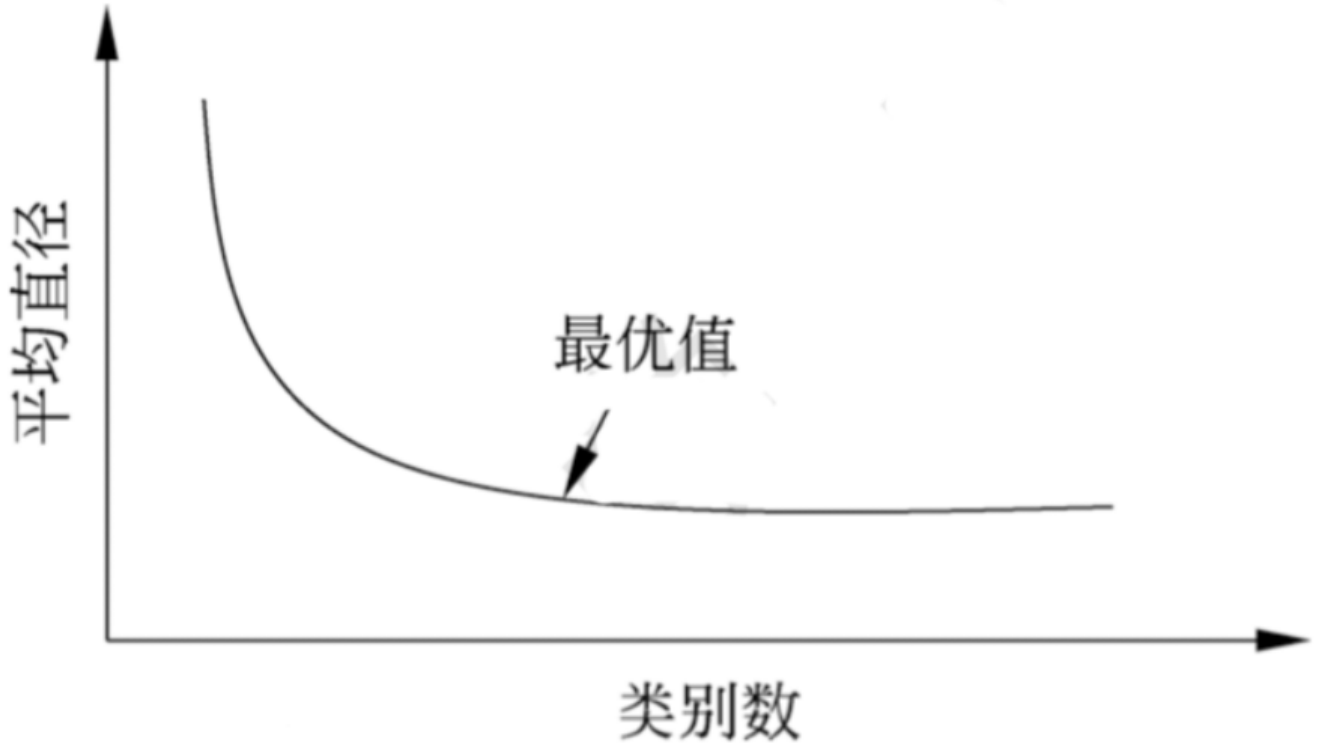
3.4 算法分析

- *k*需要事先确定；
- 迭代求解，无法确保全局最优；
- 初始聚类中心的选择会影响最终聚类结果；
- 算法复杂度O(*nmk*)；
- ……

3. *k*-means聚类算法

3.4 算法分析

- *k*的选择



4. 层次聚类算法

4.1 算法思想

- 层次聚类假设类别之间存在层次结构，将样本聚到层次化的类中。
- 分自底向上（聚合聚类）与自顶向下（分裂聚类）两种方法。
- 聚合聚类：对于给定的样本集合，开始将每个样本分到一个类中；然后按照一定的规则，例如类间最小距离，将满足条件的两个类进行合并；如此重复进行，每次减少一个类，直到满足停止条件为止，比如所有的样本聚为一类。

4. 层次聚类算法

4.1 算法思想

- 层次聚类的三要素
  - ✓距离/相似度
    - -闵可夫斯基距离
    - -马哈拉诺比斯距离
    - -相关系数
    - -夹角余弦
  - ✓合并/分裂规则
    - -类间距离最小
    - -类间距离可以是最短距离、最长距离、中心距离、平均距离
  - ✓停止条件
    - -停止条件可以是类的个数达到闭值（极端情况类的个数是1)
    - -类的直径超过阈值

4. 层次聚类算法

4.2 算法步骤

- 开始将每个样本各自分到一个类
- 之后将相距最近的两类合并，建立一个新的类
- 重复此操作直到满足停止条件
- 得到层次化的类别

4. 层次聚类算法

4.3 例题

给定5个样本的集合，样本之间的欧氏距离由如下矩阵**D**表示

$$\mathbf{D} = [d_{ij}]_{5\times5} = \begin{bmatrix} 0 & & & & \\ 7 & 0 & & & \\ 2 & 5 & 0 & & \\ 9 & 4 & 8 & 0 & \\ 3 & 6 & 1 & 5 & 0 \end{bmatrix}$$

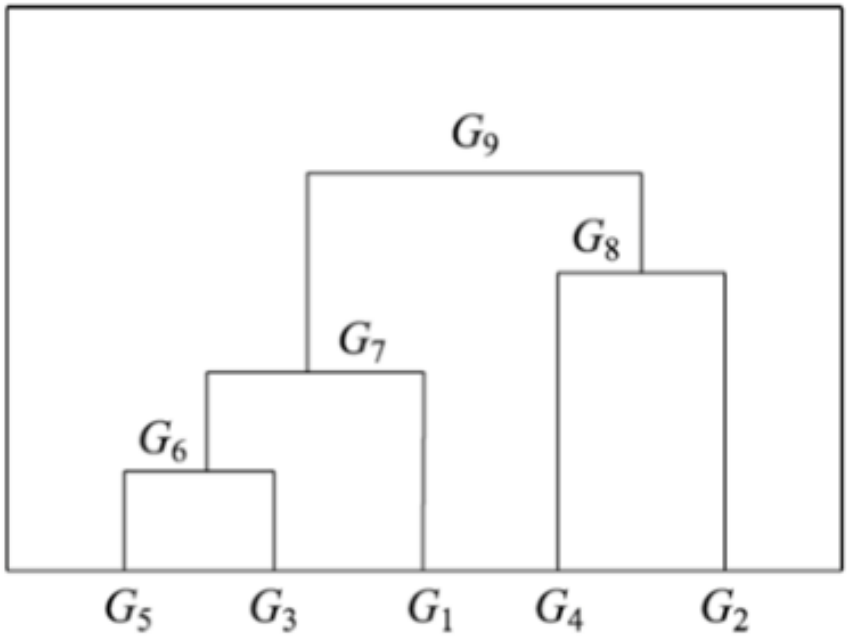
其中d_{ij}表示第i个样本与第j个样本之间的欧氏距离。

显然**D**为对称矩阵。应用聚合层次聚类法对这5个样本进行聚类。

4. 层次聚类算法

4.3 例题

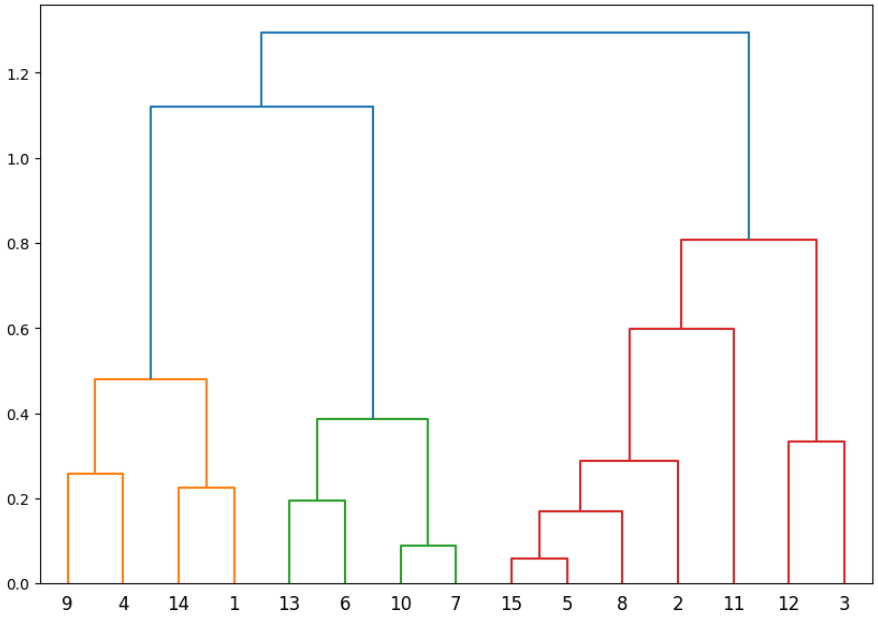
$$\mathbf{D} = [d_{ij}]_{5\times5} = \begin{bmatrix} 0 & 7 & 2 & 9 & 3 \\ 7 & 0 & 5 & 4 & 6 \\ 2 & 5 & 0 & 8 & 1 \\ 9 & 4 & 8 & 0 & 5 \\ 3 & 6 & 1 & 5 & 0 \end{bmatrix}$$

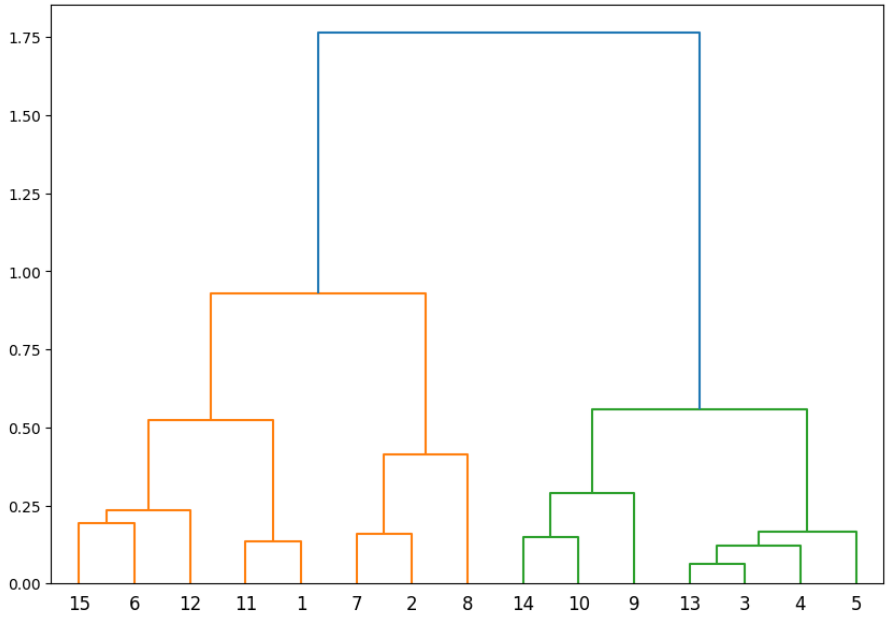


聚类方法

4. 层次聚类算法——示例

□ Cut_tree





4. 层次聚类算法

4.4 算法分析

- 距离的计算；
- 策略对结果的影响；
- 类的确定；
- 算法复杂度$O(n^3m)$；
- ……

5. DBSCAN

5.1 算法分析

定义

DBSCAN 是一个比较有代表性的密度聚类算法。它将簇定义为密度相连的点的最大集合，把具有足够高密度的区域划分为簇，并可在有噪声的空间数据库中发现任意形状的聚类。

要求

聚类空间中的一定区域内所包含对象(点或其他空间对象)的数目不小于某一给定阈值。

优点

- ✓ 聚类速度快
- ✓ 能够有效处理噪声点和发现任意形状的空间聚类

缺点

- ✓ 当数据量增大时，要求较大的内存支持，I/O消耗也很大
- ✓ 当空间聚类的密度不均匀、聚类间距差相差很大时，聚类质量较差

聚类方法

5. DBSCAN

ε 邻域

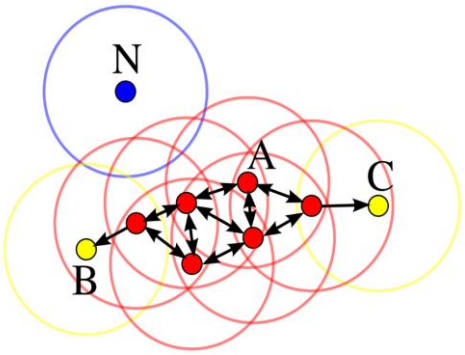
给定对象半径为ε 内的区域称为该对象的ε 邻域。

核心对象

如果给定对象邻域内的样本点数大于等于最少数目MinPts，则称该对象为核心对象。

直接密度可达

对于样本集合D，如果样本点q 在p 的邻域内，并且p为核心对象，那么对象q 从对象p 直接密度可达。



密度可达

对于样本集合 D，给定一串样本点 p1, p2 ,…, pn , p = p1, q = pn ,假如对象 pi从 pi−1直接密度可达，那么对象q从对象p密度可达。

密度相连

存在样本集合D 中的一点o，如果对象o 到对象p 和对象q 都是密度可达的，那么p 和q 密度相连。

✓可以发现，密度可达是直接密度可达的传递闭包，并且这种关系是非对称的。密度相连是对称关系。

✓DBSCAN 的目的是找到密度相连对象的最大集合。例如，图中MinPts=5，灰色的点都是核心对象，因为其邻域至少有5 个样本。黑色的样本是非核心对象。所有核心对象密度直达的样本在以灰色核心对象为中心的超球体内，如果不在超球体内，则不能密度直达。图中用箭头连起来的核心对象组成了密度可达的样本序列。在这些密度可达的样本序列的邻域内所有的样本相互都是密度相连的。

5. DBSCAN

5.2 算法步骤

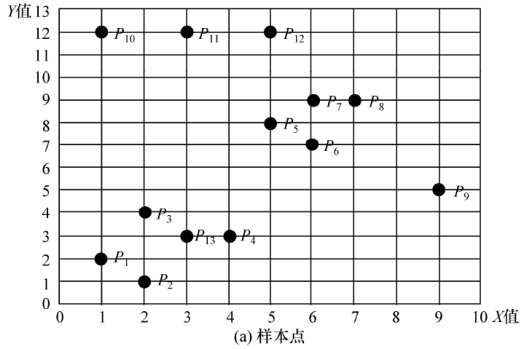
- 输入：包含n 个对象的数据库，半径ε，最少数目MinPts。
- 输出：所有生成的簇，达到密度要求。
- Repeat：从数据库中抽出一个未处理的点。
- IF 抽出的点是核心点THEN 找出所有从该点密度可达的对象，形成一个簇。
- ELSE 抽出的点是边缘点(非核心对象)，跳出本次循环，寻找下一个点。
- UNTIL 所有的点都被处理。
- DBSCAN 对用户定义的参数很敏感，细微的不同都可能导致差别很大的结果，而参数的选择无规律可循，只能靠经验确定。

聚类方法

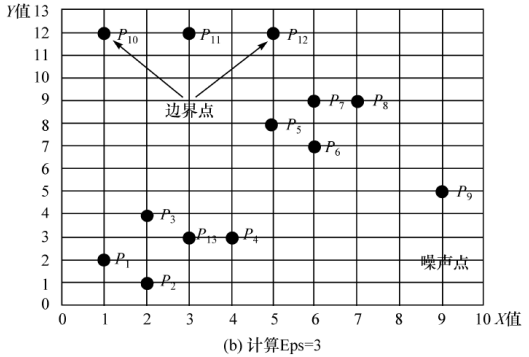
5. DBSCAN

5.3 例题：对表中的13个样本点使用DBSCAN 进行聚类

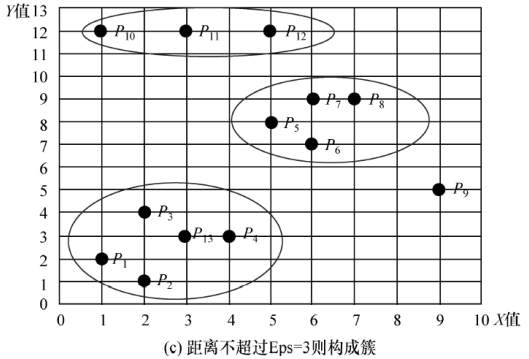
变量	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}
X	1	2	2	4	5	6	6	7	9	1	3	5	3
Y	2	1	4	3	8	7	9	9	5	12	12	12	3



(a) 样本点



(b) 计算Eps=3



(c) 距离不超过Eps=3则构成簇

STEP 1

取Eps=3，MinPts=3，依据DBSCAN 对所有点进行聚类(曼哈顿聚类)，对每个点计算其邻域Eps=3 内的点的集合。集合内点的个数超过MinPts=3 的点为核心点。

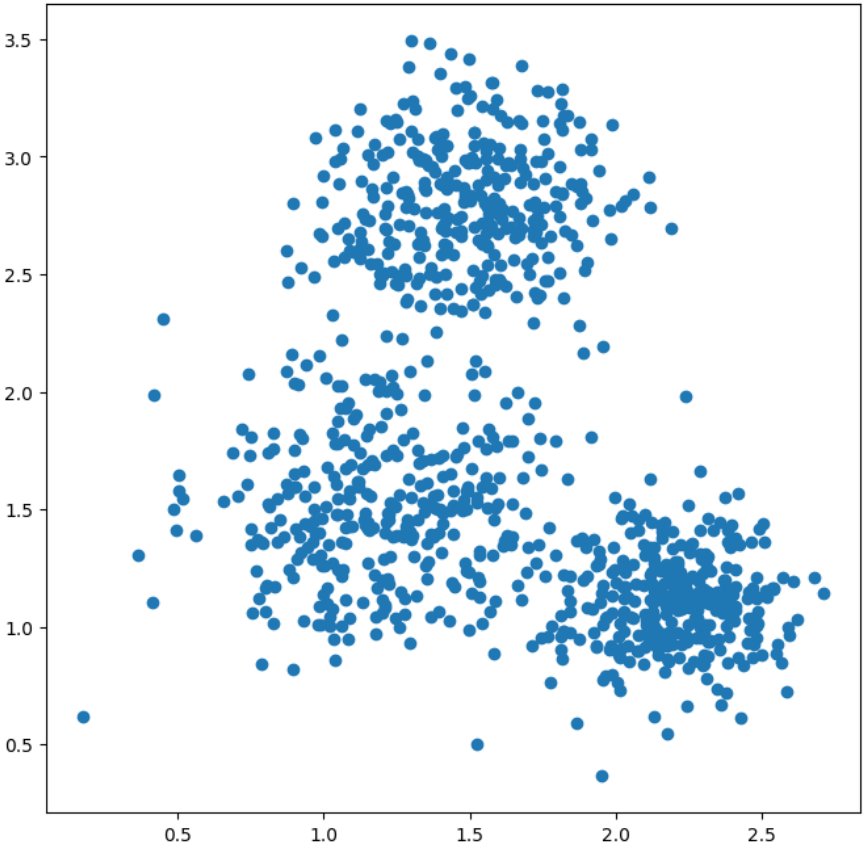
STEP 2

查看剩余点是否在核心点的邻域内，若在，则为边界点，否则为噪声点。

STEP 3

将距离不超过Eps=3 的点相互连接，构成一个簇，核心点邻域内的点也会被加入到这个簇中。

5. DBSCAN——示例



样本容量：1000

类型：某地优势植物类型A、B、C三类

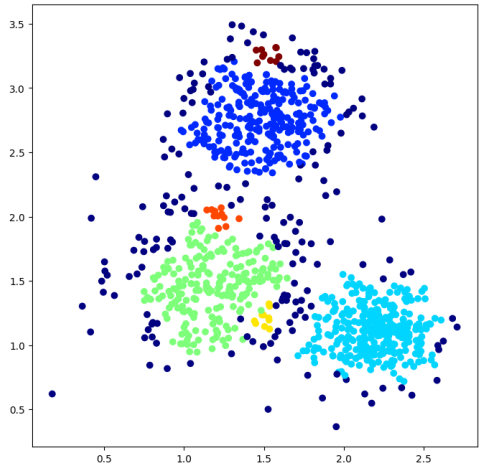
属性：植株高度、冠幅

聚类方法

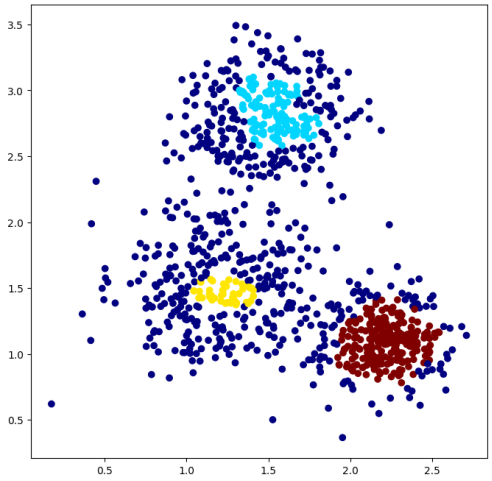
5. DBSCAN——示例

esp = 0.1

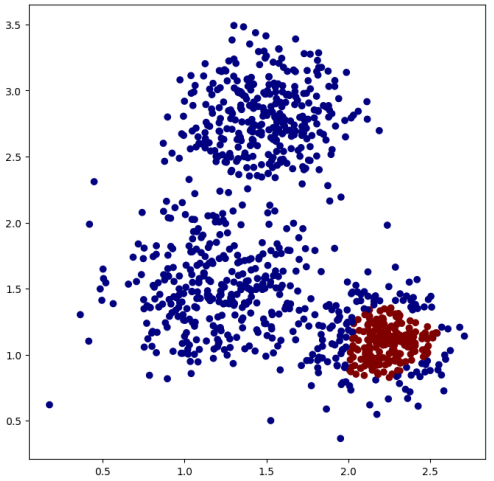
Min_points = 10



Min_points = 20

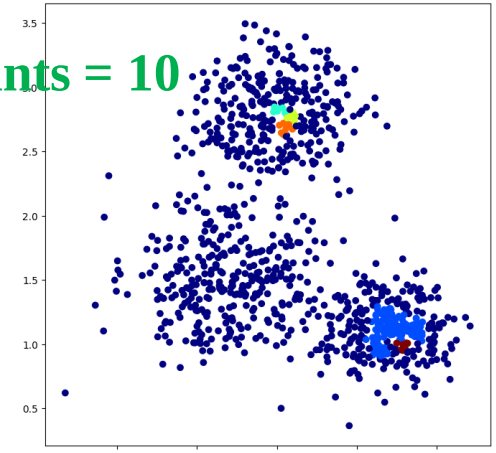


Min_points = 30

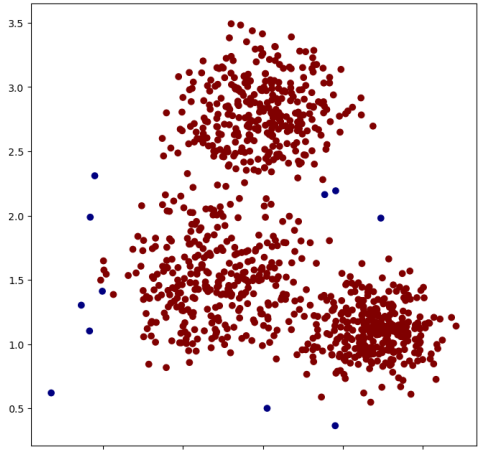


Min_points = 10

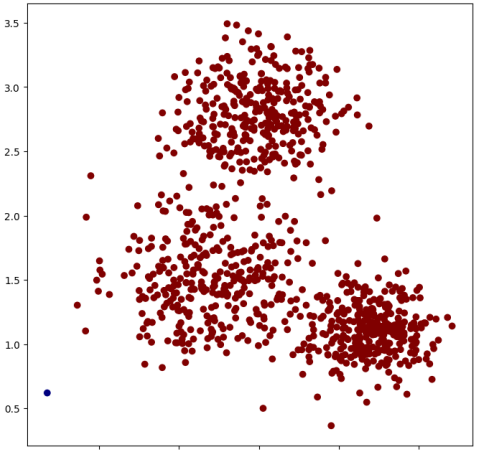
esp = 0.05



esp = 0.2



esp = 0.5



5. DBSCAN

5.4 算法分析

- 不需要输入目标数量，聚类结果没有偏倚，相对地，类似*K*-means 的聚类算法初始值对聚类结果有直接影响；
- 可以对任意形状的稠密数据集进行聚类，相对地，类似*K*-means 聚类算法一般只适用于凸数据集。
- 可以在需要时输入过滤噪声的参数。
- 调参相对于传统的的聚类算法稍复杂，主要需要对距离阈值ε，邻域样本数阈值MinPts 联合调参，不同的参数组合对最后的聚类效果有较大影响。
- ……

聚类方法

□聚类方法小结

方法	一般特点
划分方法	(1) 发现球形互斥的簇 (2) 基于距离 (3) 可以用均值或中心点等代表簇中心 (4) 对中心规模数据集有效
层次方法	(1) 聚类是一个层次分解(即多层) (2) 不能纠正错误的合并或划分 (3) 可以集成其他技术，如微聚类或考虑对象“连接”
DBSCAN	(1) 可以发现任意形状的簇 (2) 簇是对象空间中被低密度区域分隔的稠密区域 (3) 簇密度：每个点的“邻域”内必须具有最少个数的点 (4) 可能过滤离群点
神经网络	对二维、三维数据的可视非常有效，但学习模式较少时，网格的聚类效果取决于输入模式的先后顺序，且网络连接权向量的初始状态对网络的收敛性能有很大影响

聚类分析实验

- **数据：教材271页，习题5。**
- **实验要求：**
  - 根据给定数据进行聚类分析;
  - 分别使用K-means、层次聚类和DBSCAN聚类三种方法中不少于两种方法进行聚类实验;
  - 对比不同聚类方法并分析实验结果。