武汉大学公共基础必修课——机器学习

第八讲 聚类

授课老师：郭迟 教授

guochi@whu.edu.cn

武汉大学测绘学院

2021.12

聚类

1. 聚类任务概述
2. 划分式聚类方法
3. 基于密度的聚类
4. 层次化聚类

第八讲 聚类

聚类任务概述

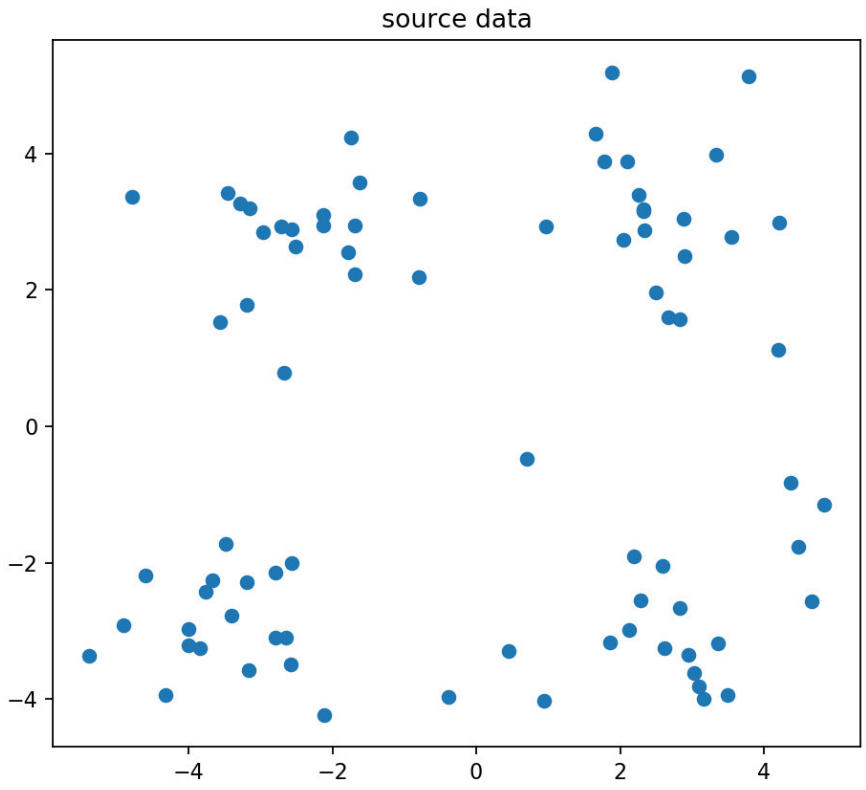
- 聚类（Clustering）是按照某个特定标准（如距离）把一个数据集分割成不同的类或簇，使得同一个簇内的数据对象的相似性尽可能大，同时不在同一个簇中的数据对象的差异性也尽可能地大。聚类后同一类的数据尽可能聚集到一起，不同类数据尽量分离

聚类和分类

- 聚类(Clustering)：是指把相似的数据划分到一起，具体划分的时候并不关心这一类的标签，目标就是把相似的数据聚合到一起，是一种无监督学习(Unsupervised Learning)方法
- 分类(Classification)：是把不同的数据划分开，其过程是通过训练数据集获得一个分类器，再通过分类器去预测未知数据，是一种监督学习(Supervised Learning)方法

聚类任务概述

提问：肉眼观察以下数据可以聚成几类？



聚类任务概述

1. 距离计算

- 对函数$dist(\cdot,\cdot)$，若它是一个“距离度量”，则应满足一些基本性质：
  - 非负性： $dist(x_i, x_j) \geq 0$;
  - 同一性： $dist(x_i, x_j) = 0$当且仅当$x_i = x_j$
  - 对称性： $dist(x_i, x_j) = dist(x_j, x_i)$
  - 直通性： $dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$

给定样本x_i， x_j，最常用的是闵可夫斯基距离：

相似度度量准则	相似度度量函数
Euclidean 距离	$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Manhattan 距离	$d(x,y) = \sum_{i=1}^{n} \|x_i - y_i\|$
Chebyshev 距离	$d(x,y) = \max_{i=1,2,\ldots,n}^{n} \|x_i - y_i\|$
Minkowski 距离	$d(x,y) = [\sum_{i=1}^{n}(x_i - y_i)^p]^{\frac{1}{p}}$

聚类任务概述

1. 距离计算

- 离散属性的距离定义：对无序属性可采用VDM (Value Difference Metric)
  令$m_{u,a}$表示在属性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

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

- 我们基于某种形式的距离来定义“相似度度量”，距离越大，相似度越小。然而，用于相似度度量的距离未必一定要满足距离度量的所有基本性质。而在现实任务中，有必要基于数据样本来确定合适的距离计算式，这可通过“度量学习”（metric learning）来实现

西瓜书P237 《度量学习》

2. 性能度量

- 聚类性能度量亦称为聚类“有效性指标”。一方面，对于聚类结果，我们根据某些性能度量来评估聚类好坏；另一方面，若明确了将使用的性能度量，可直接将其作为优化目标，从而得到符合要求的聚类结果
- 如何将样本划分到不同的簇？直观上来说，我们希望“簇内的样本尽可能相似，簇间的样本尽可能有差异”
- 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”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

2. 性能度量

对数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$, 假定通过聚类给出的簇划分为 $\mathcal{C}=\{C_1, C_2,\ldots,C_k\}$, 参考模型给出的簇划分为 $\mathcal{C}^*=\{C_1^*,C_2^*,\ldots,C_s^*\}$. 相应地, 令 $\boldsymbol{\lambda}$ 与 $\boldsymbol{\lambda}^*$ 分别表示与 $\mathcal{C}$ 和 $\mathcal{C}^*$ 对应的簇标记向量. 我们将样本两两配对考虑, 定义

$$
\begin{aligned}
a &= |SS|, \quad SS=\{(\boldsymbol{x}_i,\boldsymbol{x}_j) \mid \lambda_i=\lambda_j, \lambda_i^*=\lambda_j^*, i<j)\},\\
b &= |SD|, \quad SD=\{(\boldsymbol{x}_i,\boldsymbol{x}_j) \mid \lambda_i=\lambda_j, \lambda_i^*\neq\lambda_j^*, i<j)\},\\
c &= |DS|, \quad DS=\{(\boldsymbol{x}_i,\boldsymbol{x}_j) \mid \lambda_i\neq\lambda_j, \lambda_i^*=\lambda_j^*, i<j)\},\\
d &= |DD|, \quad DD=\{(\boldsymbol{x}_i,\boldsymbol{x}_j) \mid \lambda_i\neq\lambda_j, \lambda_i^*\neq\lambda_j^*, i<j)\},
\end{aligned}
$$

其中集合 SS 包含了在 $\mathcal{C}$ 中隶属于相同簇且在 $\mathcal{C}^*$ 中也隶属于相同簇的样本对, 集合 SD 包含了在 $\mathcal{C}$ 中隶属于相同簇但在 $\mathcal{C}^*$ 中隶属于不同簇的样本对, ……由于每个样本对 $(\boldsymbol{x}_i,\boldsymbol{x}_j)$ $(i<j)$ 仅能出现在一个集合中, 因此有 $a+b+c+d=m(m-1)/2$ 成立.

2. 性能度量

- 聚类分析常用的外部指标：
  - Jaccard 系数（Jaccard Coefficient，JC）

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

  - FM指数（Fowlkes and Mallows Index，FMI）

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

  - Rand指数（Rand Index，RI）

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

上述指标的结果均在[0,1]区间，值越大越好

聚类任务概述

2. 性能度量

- 聚类分析常用的外部指标：

考虑聚类结果的簇划分 $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, 定义

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

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

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

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

其中, $\text{dist}(\cdot,\cdot)$ 用于计算两个样本之间的距离; $\boldsymbol{\mu}$ 代表簇 C 的中心点 $\boldsymbol{\mu} = \frac{1}{|C|} \sum_{1 \leqslant i \leqslant |C|} \boldsymbol{x}_i$. 显然, $\text{avg}(C)$ 对应于簇 C 内样本间的平均距离, $\text{diam}(C)$ 对应于簇 C 内样本间的最远距离, $d_{\min}(C_i, C_j)$ 对应于簇 C_i 与簇 C_j 最近样本间的距离, $d_{\text{cen}}(C_i, C_j)$ 对应于簇 C_i 与簇 C_j 中心点间的距离.

2. 性能度量

- 聚类分析常用的外部指标：

  - DB指数（Davies-Bouldin Index，DBI）

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(\mu_i,\mu_j)}\right)$$

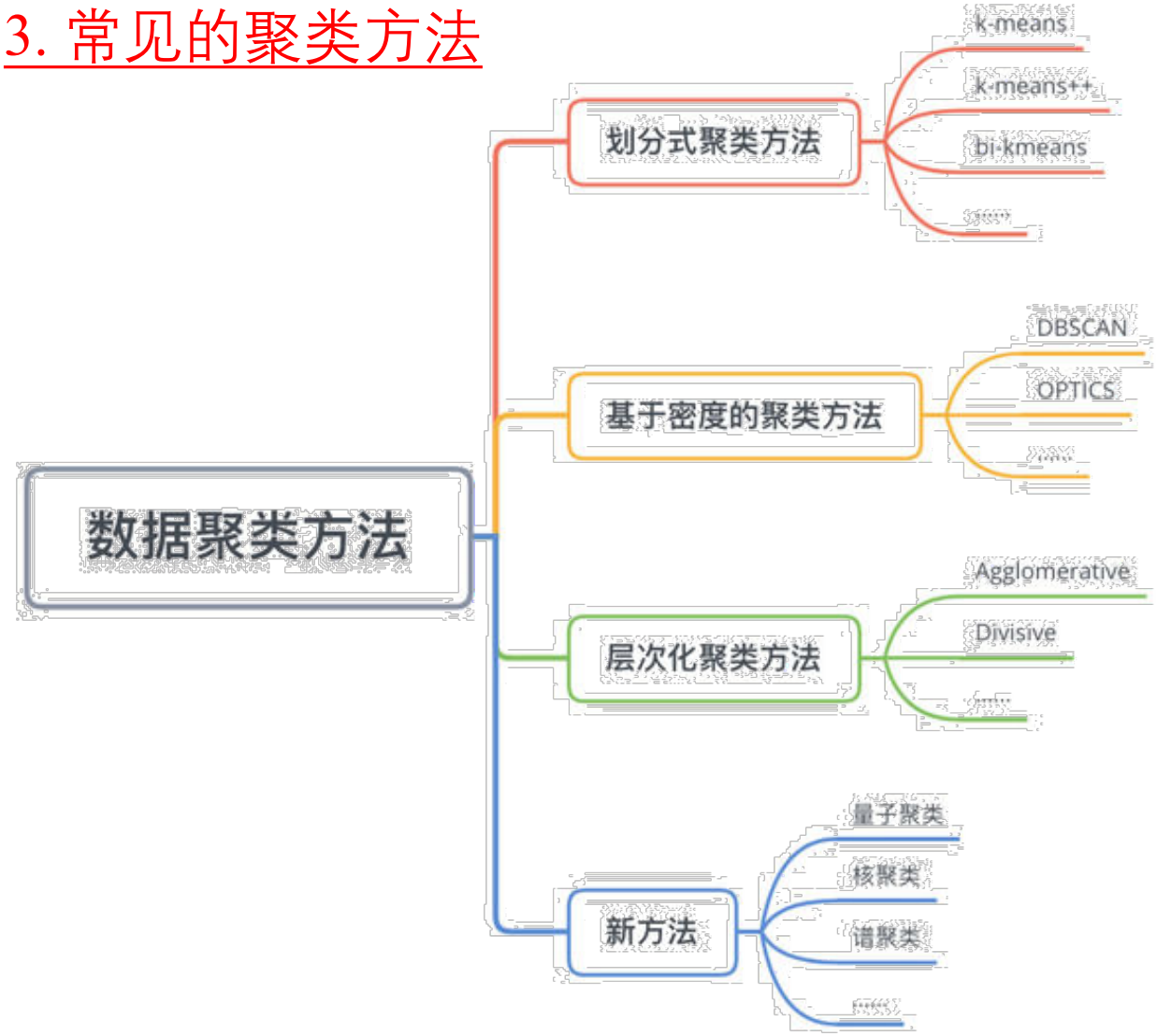
  - Dunn指数（Dunn Index，DI）

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

显然，DBI的值越小越好，而DI的值越大越好

聚类任务概述

3. 常见的聚类方法



聚类

1 聚类任务概述

2 划分式聚类方法（原型聚类）

3 基于密度的聚类

4 层次化聚类

第八讲 聚类

划分式聚类

- 划分式聚类（也叫原型聚类）方法需要事先指定簇类的数目或者聚类中心，通过反复迭代，直至最后达到“簇内的点足够近，簇间的点足够远”的目标。k均值（k-means）方法是一种经典的划分式聚类方法
- k-means算法：给定样本集$D = \{x_1, x_2, \cdots, x_m\}$， 定义$k$-means算法针对聚类所得簇$C = \{C_1, C_2, \cdots C_k\}$的最小化平方误差

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

其中$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$是簇$C_i$的均值向量（又称为参考向量、质心），该指标刻画了簇内样本围绕簇均值向量的紧密程度，E越小簇内样本的相似度越高

- k-means算法采用了贪心策略，先给定一组初始值，再通过迭代来近似求解

划分式聚类

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

过程:

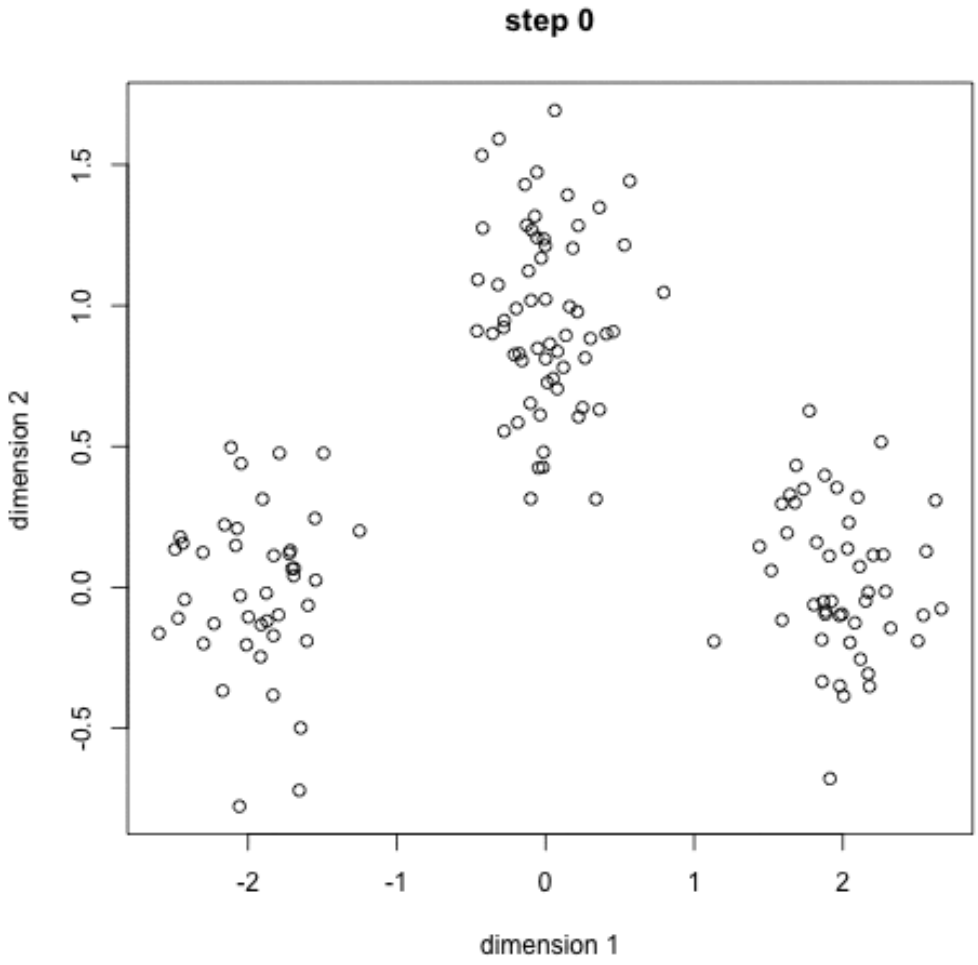
1: 从 D 中随机选择 k 个样本作为初始均值向量 $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$
2: **repeat**
3: 令 $C_i = \varnothing \ (1 \leqslant i \leqslant k)$
4: **for** $j = 1, 2, \ldots, m$ **do**
5: 计算样本 $\boldsymbol{x}_j$ 与各均值向量 $\boldsymbol{\mu}_i \ (1 \leqslant i \leqslant k)$ 的距离: $d_{ji} = \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|_2$;
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

迭代终止条件

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

k均值算法

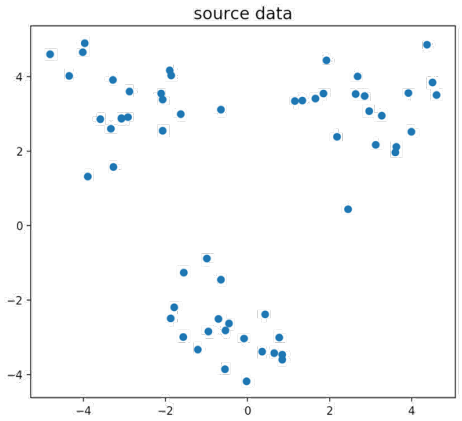
划分式聚类

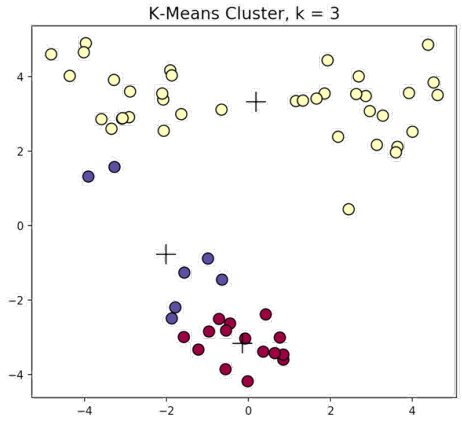


k均值算法的执行过程（动画）

划分式聚类

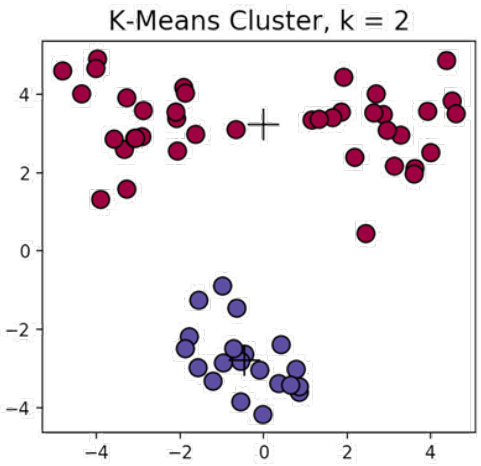
- 经典k-means算法有几个缺点：需要提前确定k值，对初始质心点敏感，对异常数据敏感

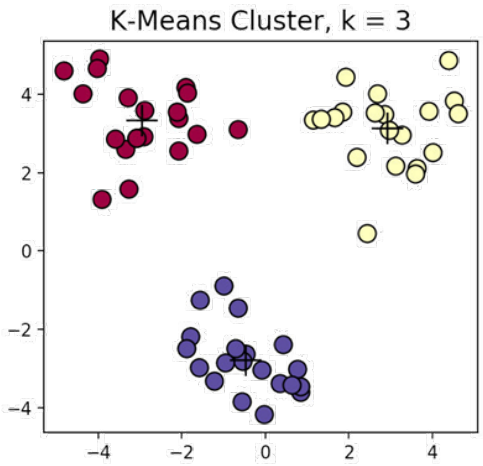


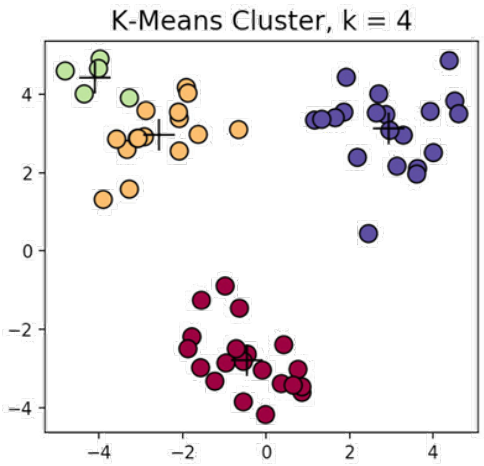


初始质心选取不当

k选取不当







聚类

1 聚类任务概述

2 划分式聚类方法（原型聚类）

3 基于密度的聚类

4 层次化聚类

基于密度的聚类

- DBSCAN是一种基于密度的聚类算法（density-based clustering），这类算法假设聚类结构能通过样本分布的紧密程度确定。DBSCAN基于一组“邻域”参数（ϵ，$MinPts$）来刻画样本分布的紧密程度，给定样本集$D = \{x_1, x_2, \cdots, x_m\}$，定义如下几个概念：

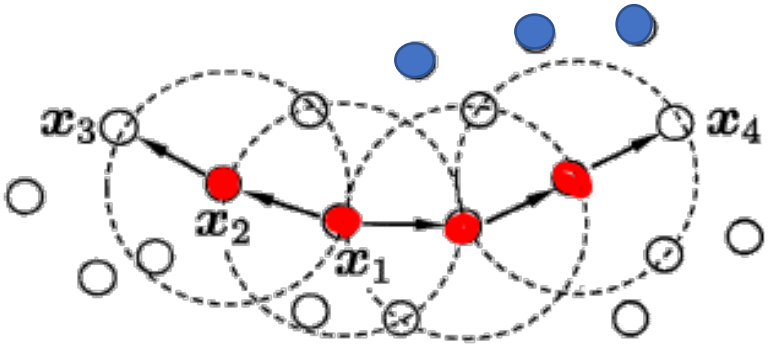
- ϵ-邻域：对$x_j \in D$，其ϵ-邻域包含样本集D中与x_j的距离不大于ϵ的样本
- 核心对象：若x_j的ϵ-邻域至少包含$MinPts$个样本，则x_j是一个核心对象
- 密度直达：若x_j位于x_i的ϵ-邻域中，且x_i是核心对象，则称x_j由x_i密度直达
- 密度可达：对x_i与x_j，若存在样本序列$p_1, p_2, \cdots, p_n$，其中$p_1 = x_i$，$p_n = x_j$且p_{i+1}由p_i密度直达，则称x_j由x_i密度可达
- 密度相连：对x_i与x_j，若存在x_k使得x_i与x_j均由x_k密度可达，则称x_i与x_j密度相连

基于密度的聚类

- DBSCAN是一种基于密度的聚类算法（density-based clustering），这类算法假设聚类结构能通过样本分布的紧密程度确定。DBSCAN基于一组“邻域”参数（ϵ，$MinPts$）来刻画样本分布的紧密程度，给定样本集$D = \{x_1, x_2, \cdots, x_m\}$，定义如下几个概念：

- ϵ-邻域：对$x_j \in D$，其ϵ-邻域包含样本集D中与x_j的距离不大于ϵ的样本
- 核心对象：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
- 密度直达：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：对x_i与x_j，若存在样本序列$p_1, p_2, \cdots, p_n$，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：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

基于密度的聚类



- x_1到x_2是密度直达
- x_1到x_3、x_4是密度可达
- x_3与x_4是密度相连

ϵ描述了某一样本的邻域距离阈值，$MinPts$描述了某一样本的距离为ϵ的邻域中样本个数的阈值

红色点均为核心对象，$MinPts = 3$

蓝色点均为异常点，既不是核心点也不是边界点

- 基于上述概念，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。形式化的说，给定邻域参数（ϵ，$MinPts$），簇$C \subseteq D$是满足以下性质的非空样本子集：
  - 连续性：$x_i \in C$，$x_j \in C \Rightarrow x_i$与$x_j$密度相连
  - 最大性：$x_i \in C$，x_j由x_i密度可达$\Rightarrow x_j \in C$

事实上，若x为核心对象，由x密度可达的所有样本组成的集合记为$X = \{x' \in D | \ x'$由x密度可达$\}$。因此，DBSCAN先任选数据集中的一个核心对象作为“种子”，再由此出发确定相应的簇

- 这个DBSCAN的簇里面可以有一个或者多个核心对象:
  - 如果只有一个核心对象，则簇里其他的非核心对象样本都在这个核心对象的ϵ–邻域里；
  - 如果有多个核心对象，则簇里的任意一个核心对象的ϵ–邻域中一定有一个其他的核心对象，否则这两个核心对象无法密度可达。这些核心对象的ϵ–邻域里所有的样本的集合组成的一个DBSCAN聚类簇

- 那么怎么才能找到这样的簇样本集合呢？DBSCAN使用的方法很简单，它任意选择一个没有类别的核心对象，然后找到所有这个核心对象能够密度可达的样本集合，即为一个聚类簇。接着继续选择另一个没有类别的核心对象去寻找密度可达的样本集合，这样就得到另一个聚类簇。一直运行到所有核心对象都有类别为止

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: $\Omega = \varnothing$
for $j = 1, 2, \ldots, m$ **do**
  确定样本 $\boldsymbol{x}_j$ 的 ϵ-邻域 $N_\epsilon(\boldsymbol{x}_j)$;
  if $|N_\epsilon(\boldsymbol{x}_j)| \geqslant MinPts$ **then**
    将样本 $\boldsymbol{x}_j$ 加入核心对象集合: $\Omega = \Omega \bigcup \{\boldsymbol{x}_j\}$
  end if
end for
初始化聚类簇数: $k = 0$
初始化未访问样本集合: $\Gamma = D$
while $\Omega \neq \varnothing$ **do**
  记录当前未访问样本集合: $\Gamma_{\text{old}} = \Gamma$;
  随机选取一个核心对象 $\boldsymbol{o} \in \Omega$ 初始化队列 $Q = < \boldsymbol{o} >$;
  $\Gamma = \Gamma \setminus \{\boldsymbol{o}\}$;
  while $Q \neq \varnothing$ **do**
    取出队列 Q 中的首个样本 $\boldsymbol{q}$;
    if $|N_\epsilon(\boldsymbol{q})| \geqslant MinPts$ **then**
      令 $\Delta = N_\epsilon(\boldsymbol{q}) \bigcap \Gamma$;
      将 Δ 中的样本加入队列 Q;
      $\Gamma = \Gamma \setminus \Delta$;
    end if
  end while
  $k = k + 1$, 生成聚类簇 $C_k = \Gamma_{\text{old}} \setminus \Gamma$;
  $\Omega = \Omega \setminus C_k$
end while

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

找出所有核心对象

判定是否为核心对象

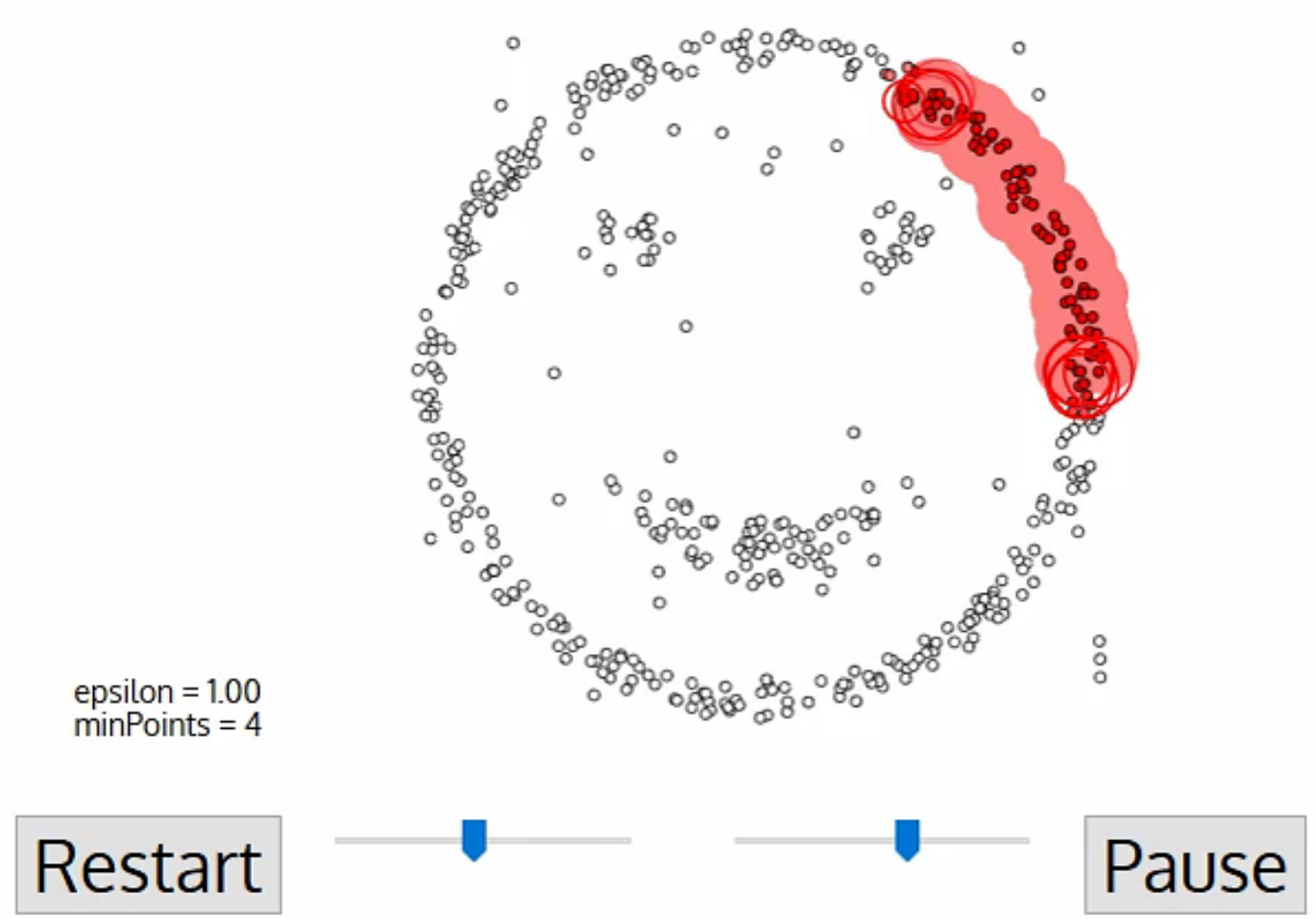
找出其邻域内的样本，即密度直达

核心对象o所有密度可达的样本集合

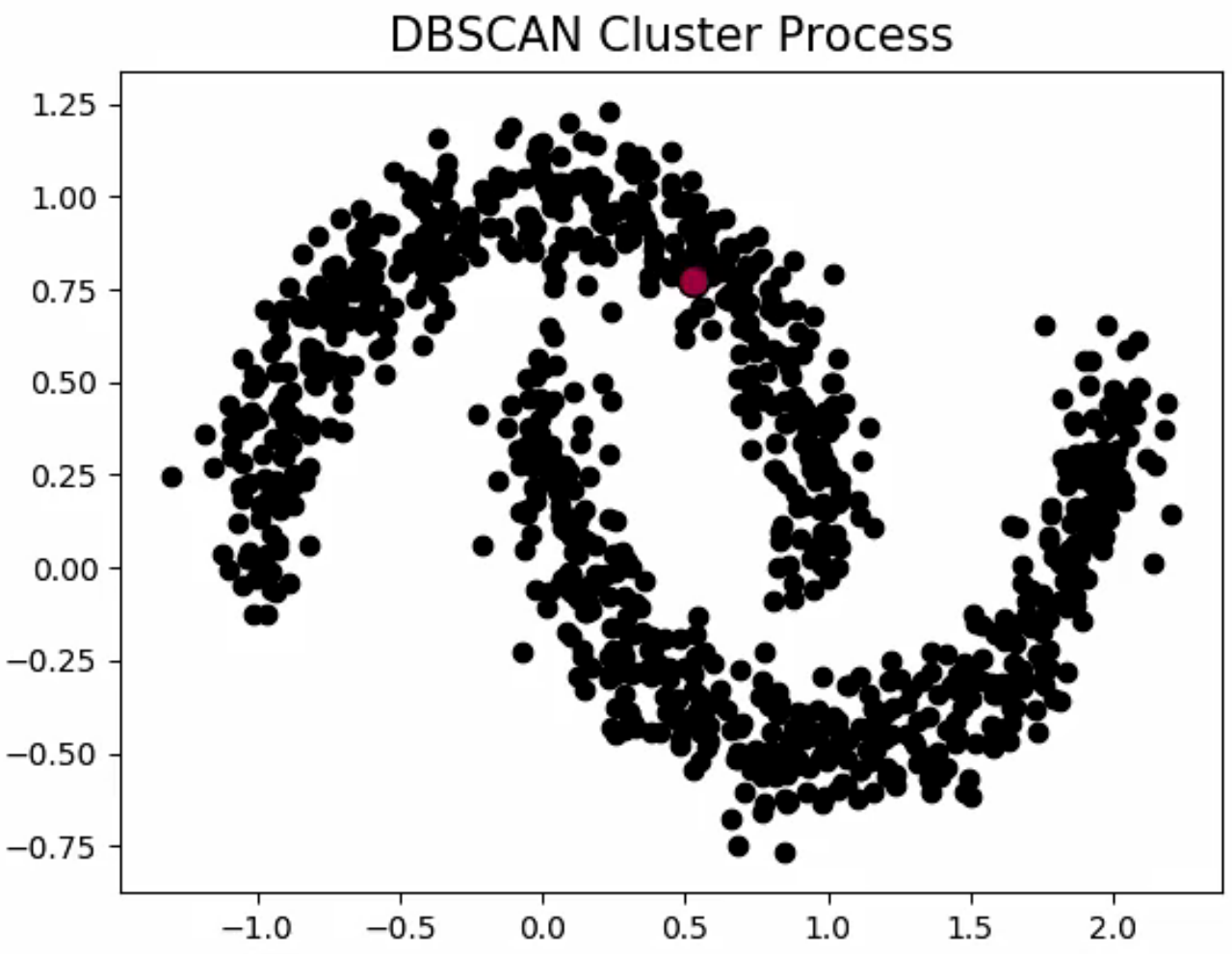
直到遍历完所有核心对象

第八讲 聚类

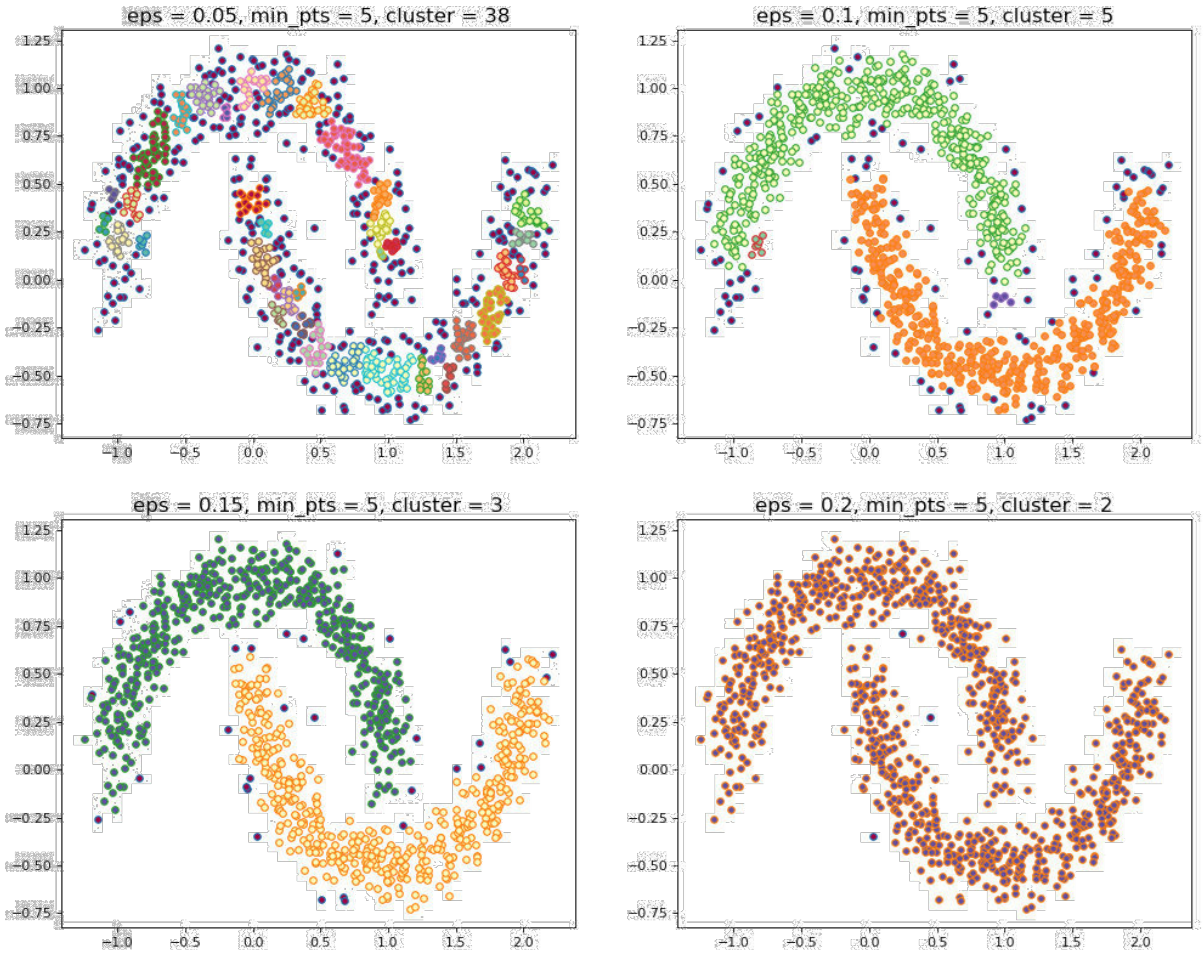
基于密度的聚类



基于密度的聚类

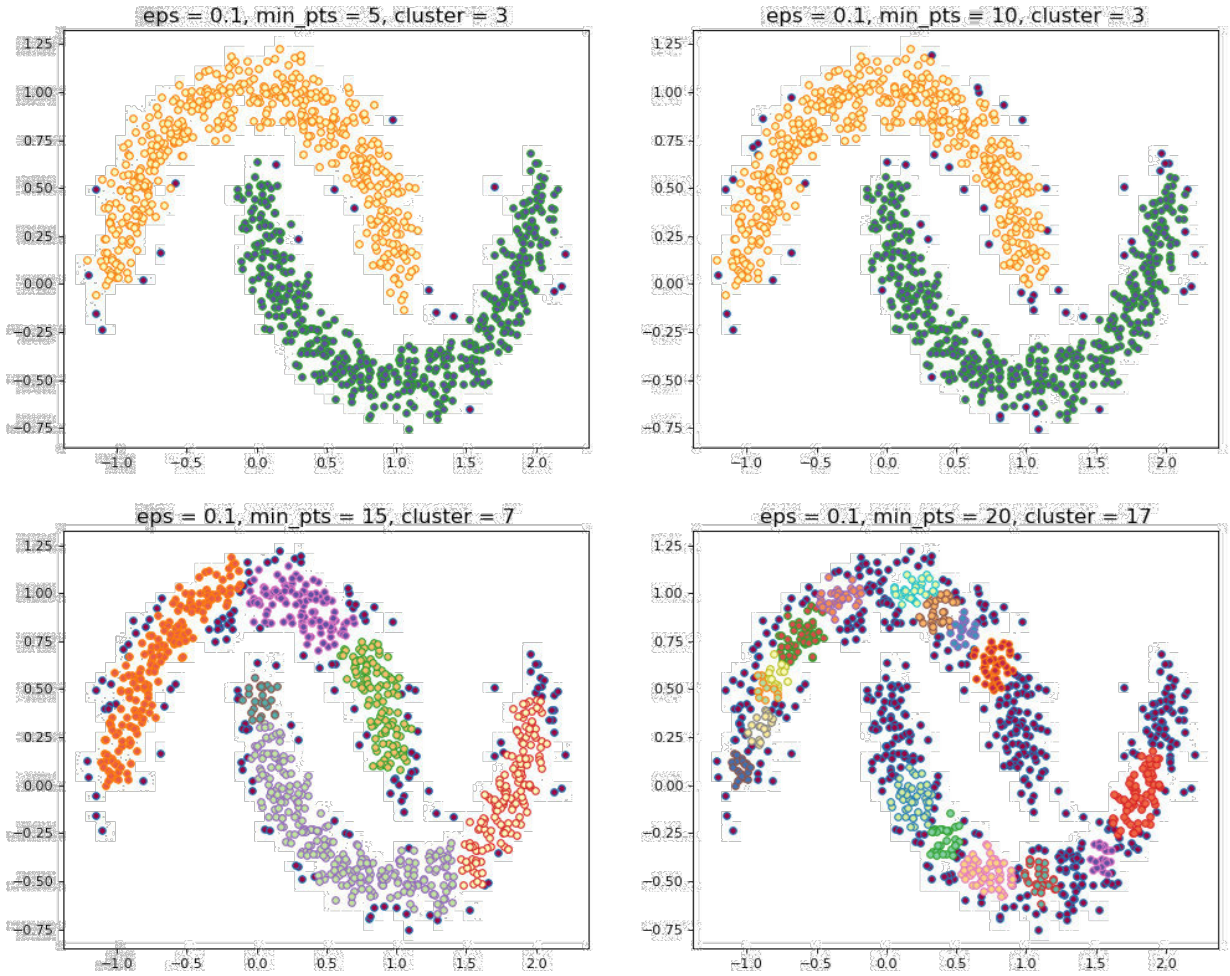

DBSCAN Cluster Process
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

基于密度的聚类



不同ϵ–值的聚类结果

基于密度的聚类

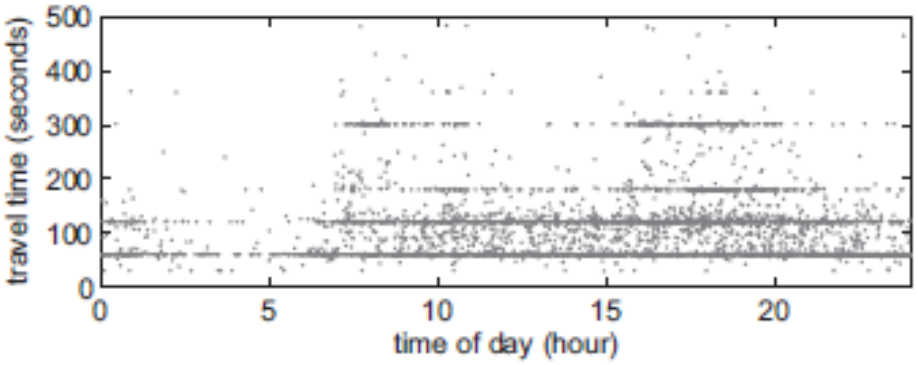


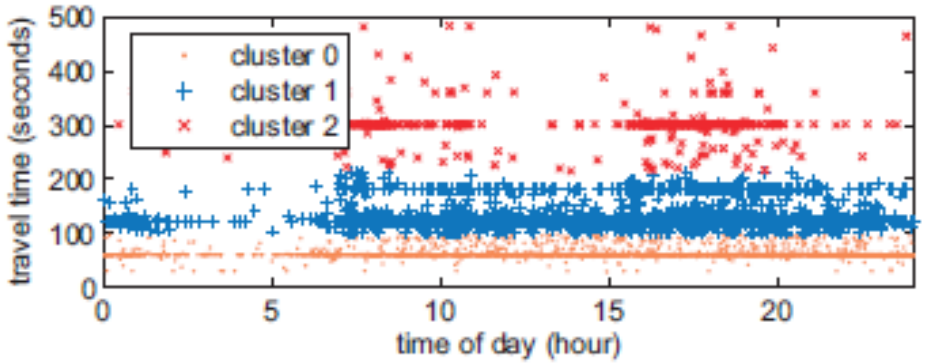
不同MinPts值的聚类结果

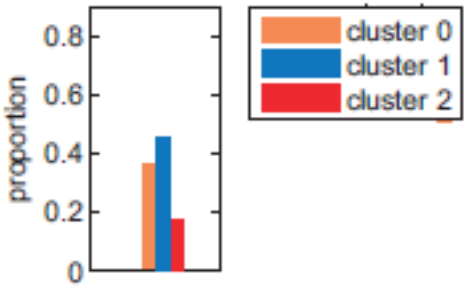
基于密度的聚类

- 一般来说，DBSCAN算法有以下几个特点：
  - 需要提前确定ϵ和MinPts值
  - 不需要提前设置聚类类别的个数
  - 对初值选取敏感，对噪声不敏感
  - 对密度不均的数据聚合效果不好

思考题：这是某路段一天24小时内统计的车辆通行时间（秒），希望对其聚类获得其通行时间的概率分布，如何做？







聚类

1 聚类任务概述

2 划分式聚类方法（原型聚类）

3 基于密度的聚类

4 层次化聚类

第八讲 聚类

层次聚类算法

➢层次聚类（Hierarchical Clustering）算法可以自上而下或自下而上实现。自下而上的算法在一开始就将每个数据点视为一个单一的聚类，然后依次合并（或聚集）类，直到所有类合并成一个包含所有数据点的单一聚类。自上而下的方法则先将所有数据看作一类，然后逐步划分

- 凝聚的（agglomerative）方法（自底向上）

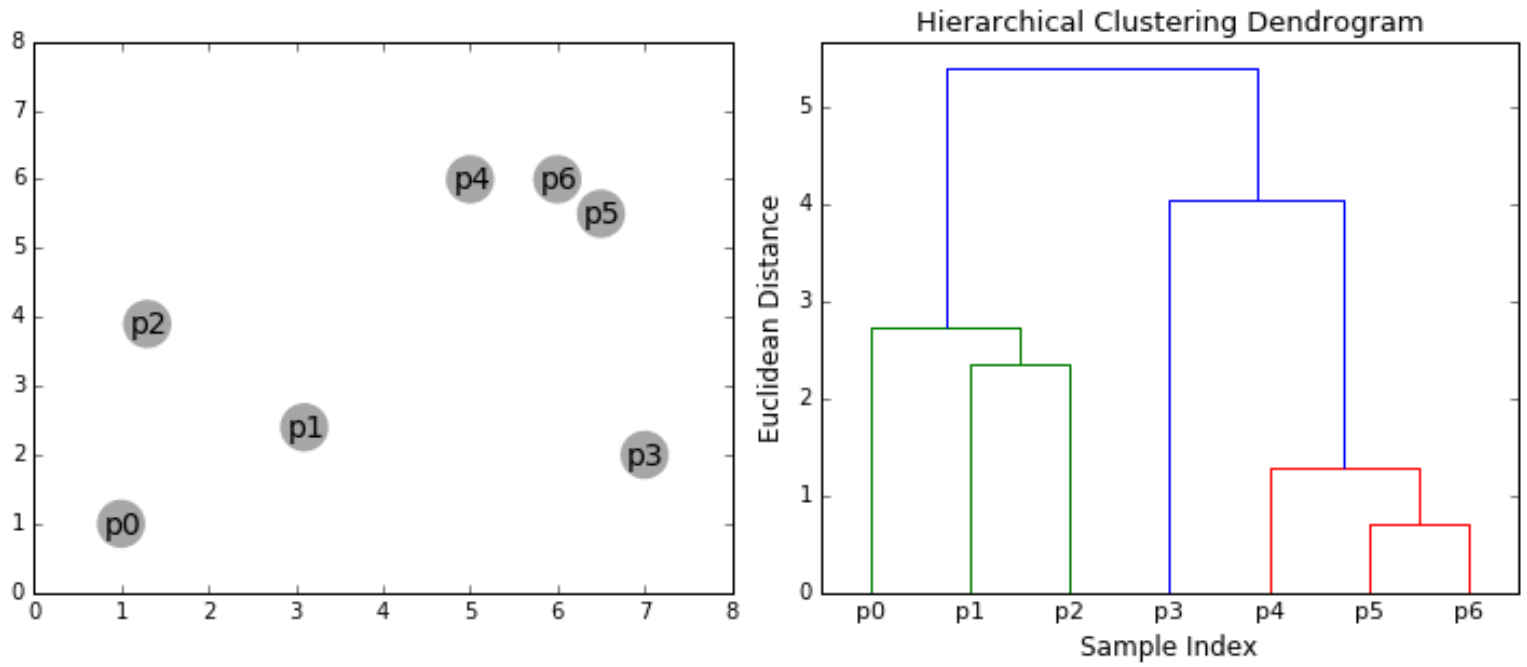
  思想：将每个对象作为单独的一组，然后根据同类相近，异类相异的原则，合并对象，直到所有的组合并成一个，或达到一个终止条件为止

- 分裂的方法（divisive）（自顶向下）

  思想：将所有的对象置于一类，在迭代的每一步中，一个类不断地分为更小的类，直到每个对象在单独的一个类中，或达到一个终止条件

第八讲 聚类

层次聚类算法



层次聚类示意图（动画）

层次聚类算法

➢ **AGNES算法是一种代表性的层次聚类算法，其**基本步骤是：

假设有N个待聚类的样本，

- 初始化-->把每个样本归为一类，计算每两个类之间的距离，也就是样本与样本之间的相似度；
- 寻找各个类之间最近的两个类，把他们归为一类（这样类的总数就少了一个）
- 重新计算新生成的这个**类与各个旧类之间的相似度**；
- 重复2和3直到所有样本点都归为一类，结束

层次聚类算法

- 可以看出其中最关键的一步就是**计算两个类簇的相似度**，这里有多种度量方法：

- 单链接（single-linkage）:取类间最小距离。

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

- 全链接（complete-linkage）:取类间最大距离

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

均链接（average-linkage）:取类间两两的平均距离

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

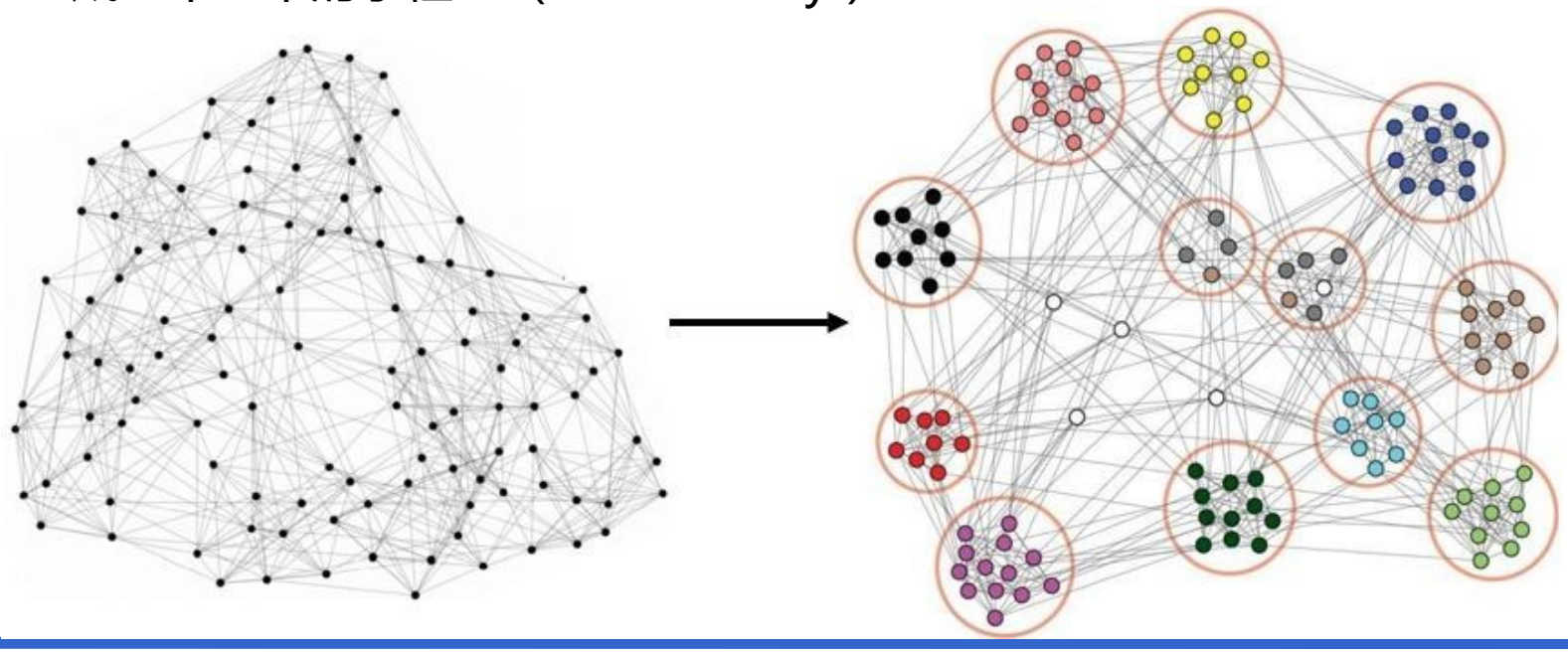
很容易看出：**单链接的包容性极强，全链接则是坚持到底，只要存在缺点就坚决不合并，均连接则是从全局出发顾全大局**

层次聚类算法

- 层次聚类方法的特点：
  - 层次聚类算法不要求我们指定聚类的数量，我们甚至可以选择哪个聚类看起来最好。此外，该算法对距离度量的选择不敏感；
  - 当底层数据具有层次结构时，可以恢复层次结构；
  - 层次聚类的优点是以低效率为代价的，因为它具有$O(n^3)$的时间复杂度，与k-means和高斯混合模型的线性复杂度不同

延申阅读：复杂网络的社区发现

- 近20年来，复杂网络（complex network）研究广泛，是复杂系统的抽象。人们发现复杂网络具有一定的社区结构，即复杂网络并不是一大批性质完全相同的节点随机连接在一起，也不是各种类型的节点之间不相关的随意链接，而是“乱中有序”——相同类型节点之间连接较多，构成一个一个的小社区（Community）



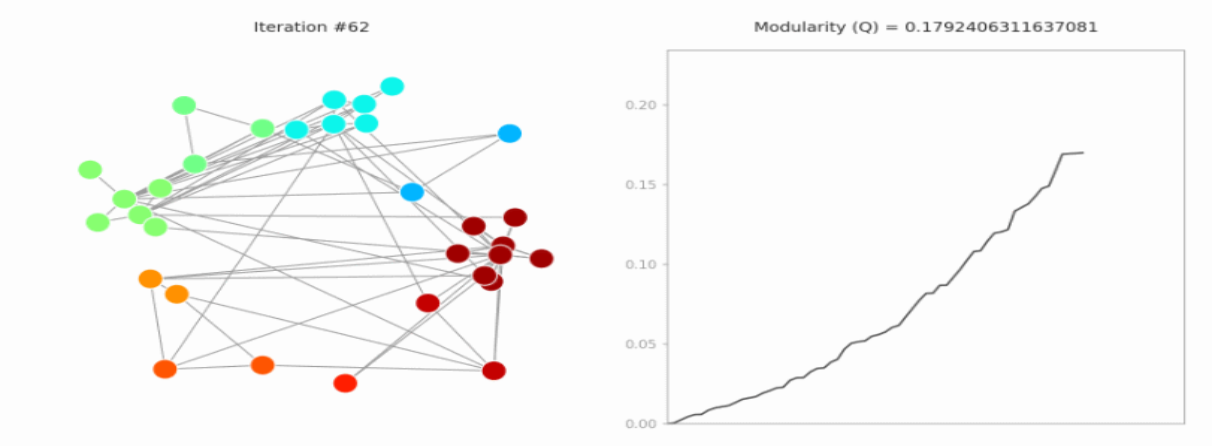
延申阅读：复杂网络的社区发现

- 在大型复杂网络中进行社区发现与搜索算法，具有重要的实际意义
  - 社会关系网络中，能够显示根据兴趣、职业、地域、背景而形成的真实的社会团体。从而可以进行人物分析、职业推荐、圈子推荐、好友推荐、校友发现、以及精准广告投放
  - 引文网络社区中，可以根据主题词、作者、内容、单位进行文章搜索与发掘。具体到应用可以按照用户搜索词进行相关推荐，或者对引用次数及质量分析从而确定影响因子，或者具体到查重算法的设计，都需要网络社区理论做支撑
  - 生物化学网络社区。此类网络中的社区可以是某一类型的功能单元，发现其中社区有助于更加有效地理解开发这些网络。例如生物学领域的食物链分析、人类基因库分析等

层次聚类算法

延申阅读：复杂网络的社区发现

- 层次聚类方法在复杂网络社区发现中被广泛应用。根据网络中结点间的紧密度或者相似度，将网络划分为若干子集。聚类思路依然分为分裂方法(Divisive Method)和凝聚方法(Agglomerative Method) 。GN(Girvan-Newman)算法是比较著名的分裂算法， Newman方法是基于模块相似度的凝聚方法



[1] Girvan M, Newman M E J. Community structure in social and biological networks[J]. PNAS, 2001, 99(12): 7821-7826.
[2] Newman M E J. Fast algorithm for detecting community structure in networks[J]. Physical Review E, 2004, 69(6): 066133.