机器学习

第八章 聚类

鲍鹏
北京交通大学

本章目录

本章目录

无监督学习方法概述

- **监督学习和无监督学习的区别**

监督学习

在一个典型的监督学习中，训练集有标签y，我们的目标是找到能够区分正样本和负样本的决策边界，需要据此拟合一个假设函数。

无监督学习

与此不同的是，在无监督学习中，我们的数据没有附带任何标签y，无监督学习主要分为聚类、降维、关联规则、推荐系统等方面。

无监督学习方法概述

- **主要的无监督学习方法**

聚类（Clustering）

如何将教室里的学生按爱好、身高划分为5类？

降维（ Dimensionality Reduction ）

如何将将原高维空间中的数据点映射到低维度的空间中？

关联规则（ Association Rules）

很多买尿布的男顾客，同时买了啤酒，可以从中找出什么规律来提高超市销售额？

推荐系统（ Recommender systems）

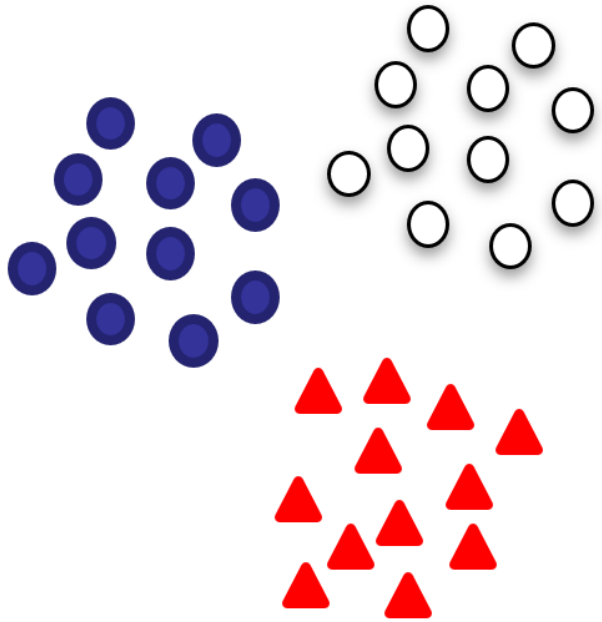
很多客户经常上网购物，根据他们的浏览商品的习惯，给他们推荐什么商品呢？

本章目录

聚类任务概述

- **聚类的背景知识--基本思想**

图中的数据可以分成三个分开的**点集**(称为**簇**)，一个能够分出这些点集的算法，就被称为**聚类算法**。



聚类算法示例

聚类任务概述

- **聚类的背景知识--形式化描述**

假定样本集$D = \{x_1, x_2, \cdots, x_m\}$包含$m$个无标记样本，每个样本$x_i = (x_{i1}; x_{i2}; \cdots; x_{in})$是一个$n$维的特征向量，聚类算法将样本集$D$划分成$k$个不相交的簇$\{C_l \mid l = 1,2, \ldots, k\}$。其中$C_{l'} \cap C_l = \phi$，且$D = \cup_{l=1}^{k} C_l$。

相应地，用$\lambda_j \in \{1,2, \ldots, k\}$表示样本$x_j$的“簇标记”（即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$\lambda = \{\lambda_1; \lambda_2; \cdots; \lambda_m\}$表示。

聚类任务概述

- **聚类**

主要算法
K-means、密度聚类、层次聚类

主要应用

市场细分、文档聚类、图像分割、图像压缩、聚类分析、特征学习或者词典学习、确定犯罪易发地区、保险欺诈检测、公共交通数据分析、IT资产集群、客户细分、识别癌症数据、搜索引擎应用、医疗应用、药物活性预测……

聚类任务概述

- **聚类案例**

1.医疗

医生可以使用聚类算法来发现疾病。以甲状腺疾病为例。当我们对包含甲状腺疾病和非甲状腺疾病的数据集应用无监督学习时，可以使用聚类算法来识别甲状腺疾病数据集。

聚类任务概述

- **聚类案例**

2.市场细分

为了吸引更多的客户，每家公司都在开发易于使用的功能和技术。为了了解客户，公司可以使用聚类算法。这将帮助公司了解用户群，然后对每个客户进行归类。

聚类任务概述

- **聚类案例**

3.金融业

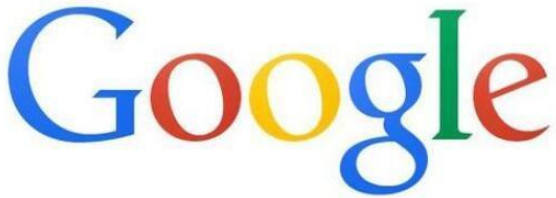
银行可以观察到金融欺诈行为，就此向客户发出警告。在聚类算法的帮助下，保险公司可以发现某些客户的欺诈行为，并调查类似客户的保单是否有欺诈行为。

聚类任务概述

- **聚类案例**

4.搜索引擎

百度是人们使用的搜索引擎之一。举个例子，当我们搜索一些信息，如在某地的超市，百度将为我们提供不同的超市的选择。这是聚类的结果，提供给你的结果就是聚类的相似结果。



聚类任务概述

- **聚类案例**

5.社交网络

比如在社交网络的分析上。已知你朋友的信息，比如经常发email的联系人，或是你的微博好友、微信的朋友圈，我们可运用聚类方法自动地给朋友进行分组，做到让每组里的人们彼此都熟识。

本章目录

聚类的基本概念

- **相似度或距离**

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)，有多种相似度或距离定义。因为相似度直接影响聚类的结果，所以其选择是聚类的根本问题。

聚类的基本概念

- **闵可夫斯基距离**

闵可夫斯基距离越大相似度越小，距离越小相似度越大。

给定样本集合X,X是m维实数向量空间R^m中点的集合，其中

$$x_i, x_j \in X, x_i = (x_{1i}, x_{2i}, \cdots, x_{mi})^{\mathrm{T}}, x_j = (x_{1j}, x_{2j}, \cdots, x_{mj})^{\mathrm{T}}$$

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

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

聚类的基本概念

- **闵可夫斯基距离**

当$p=2$时称为欧氏距离（Euclidean distance)　　$d_{ij}=(\sum_{k=1}^{m}|x_{ki}-x_{kj}|^2)^{\frac{1}{2}}$

当$p=1$时称为曼哈顿距离（Manhattan distance)　　$d_{ij}=\sum_{k=1}^{m}|x_{ki}-x_{kj}|$

当$p=\infty$时称为切比雪夫距离（Chebyshev distance）　　$d_{ij}=\max_{k}|x_{ki}-x_{kj}|$

聚类的基本概念

- **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）之间的相关性并与各个分量的尺度无关。

马哈拉诺比斯距离越大相似度越小，距离越小相似度越大。

给定一个样本集合X,$X = [x_{ij}]_{m\times n}$，其协方差矩阵记作S。样本x_i与样本x_j之间的马哈拉诺比斯距离d_{ij}定义为

$$d_{ij} = \left[\left(x_i - x_j\right)^{\mathrm{T}} S^{-1}\left(x_i - x_j\right)\right]^{\frac{1}{2}}$$

$$x_i = \left(x_{1i}, x_{2i}, \cdots, x_{mi}\right)^{\mathrm{T}}, x_j = \left(x_{1j}, x_{2j}, \cdots, x_{mj}\right)^{\mathrm{T}}$$

聚类的基本概念

- **相关系数**

样本之间的相似度也可以用相关系数（correlation coefficient）来表示。

相关系数的绝对值越接近于1，表示样本越相似

越接近于0，表示样本越不相似。

样本x_i与样本x_j之间的相关系数定义为

$$r_{ij}=\frac{\sum_{k=1}^{m}\left(x_{ki}-\bar{x}_i\right)\left(x_{kj}-\bar{x}_j\right)}{\left[\sum_{k=1}^{m}\left(x_{ki}-\bar{x}_i\right)^2\sum_{k=1}^{m}\left(x_{kj}-\bar{x}_j\right)^2\right]^{\frac{1}{2}}}\qquad \bar{x}_i=\frac{1}{m}\sum_{k=1}^{m}x_{ki},\bar{x}_j=\frac{1}{m}\sum_{k=1}^{m}x_{kj}$$

聚类的基本概念

- **夹角余弦**

样本之间的相似度也可以用夹角余弦（cosine）来表示。

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。

样本x_i与样本x_j之间的夹角余弦定义为

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

聚类的基本概念

- **相似度**

用距离度量相似度时，距离越小样本越相似

用相关系数时，相关系数越大样本越相似

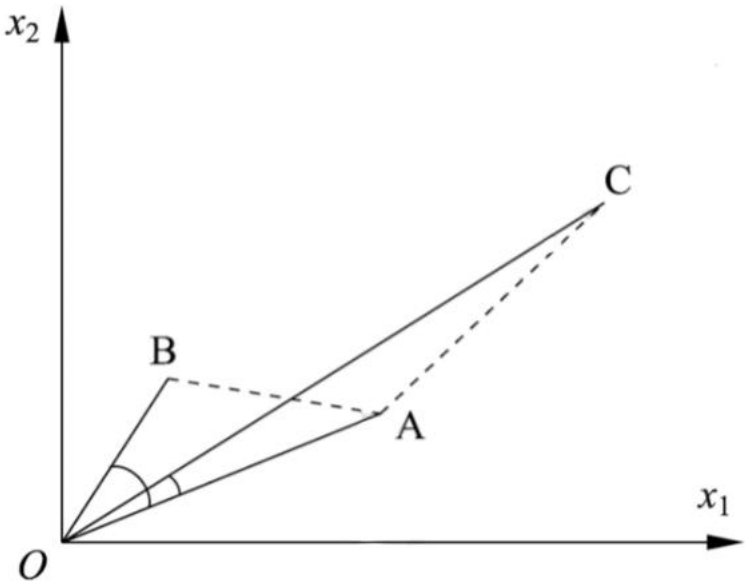
注意不同相似度度量得到的结果并不一定一致。

从右图可以看出，如果从距离的角度看，

A和B比A和C更相似

但从相关系数的角度看，

A和C比A和B更相似。



聚类的基本概念

- **类或簇**

通过聚类得到的类或簇，本质是样本的子集。

如果一个聚类方法假定一个样本只能属于一个类，或类的交集为空集，那么该方法称为硬聚类（hard clustering）方法。

如果一个样本可以属于多个类，或类的交集不为空集，那么该方法称为软聚类（soft clustering）方法。

聚类的基本概念

- **类或簇**

用G表示类或簇（cluster)，用x_i, x_j表示类中的样本，用n_G表示G中样本的个数，用d_{ij}表示样本x_i与样本x_j之间的距离。

类或簇有多种定义，下面给出几个常见的定义。

聚类的基本概念

- **类或簇**

设T为给定的正数，若集合G中任意两个样本x_i, x_j ，有

$$d_{ij} \leq T$$

则称G为一个类或簇。

聚类的基本概念

- **类或簇**

设T为给定的正数，若对集合G的任意样本x_i，一定存在G中的另一个样本x_j，使得

$$d_{ij} \leq T$$

则称G为一个类或簇。

聚类的基本概念

- **类或簇**

设T为给定的正数，若对集合G中任意一个样本x_i，G中的另一个样本x_j满足：

$$\frac{1}{n_G - 1}\sum_{x_j \in G} d_{ij} \leqslant T$$

其中n_G为G中样本的个数，则称G为一个类或簇。

聚类的基本概念

- **类或簇**

设T和V为给定的两个正数，如果集合G中任意两个样本x_i, x_j的距离d_{ij}满足

$$\frac{1}{n_G(n_G-1)}\sum_{x_i\in G}\sum_{x_j\in G} d_{ij} \leqslant T \ , d_{ij} \leqslant V$$

则称G为一个类或簇。

以上四个定义，第一个定义最常用，并且由它可推出其他三个定义。

聚类的基本概念

- **类或簇**

类的特征可以通过不同角度来刻画，常用的特征有下面三种：

（1）类的均值$\bar{x}_G$，又称为类的中心

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

式中n_G是类G的样本个数。

聚类的基本概念

- **类或簇**

类的特征可以通过不同角度来刻画，常用的特征有下面三种：

（2）类的直径(diameter) D_G

类的直径D_G是类中任意两个样本之间的最大距离，即

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

聚类的基本概念

- **类或簇**

类的特征可以通过不同角度来刻画，常用的特征有下面三种：

（3）类的样本散布矩阵(scatter matrix) A_G与样本协方差矩阵(covariance matrix) S_G

类的样本散布矩阵A_G为 $$A_G = \sum_{i=1}^{n_G} (x_i - \bar{x}_G)(x_i - \bar{x}_G)^{\mathrm{T}}$$

样本的协方差矩阵S_G为 $$S_G = \frac{1}{m-1} A_G$$

$$= \frac{1}{m-1} \sum_{i=1}^{n_G} (x_i - \bar{x}_G)(x_i - \bar{x}_G)^{\mathrm{T}}$$

m为样本的维数

聚类的基本概念

- **类与类之间的距离**

下面考虑类G_{p}与类G_q之间的距离$D(p,q)$，也称为连接（linkage)。类与类之间的距离也有多种定义。

设类G_{p}包含n_{p}个样本，G_q包含n_q个样本，分别用$\bar{x}_p$和$\bar{x}_q$表示G_{p}和G_q的均值，即类的中心。

聚类的基本概念

- **类与类之间的距离**

最短距离或单连接（ single linkage)

定义类G_p的样本与G_q的样本之间的最短距离为两类之间的距离

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

聚类的基本概念

- **类与类之间的距离**

最长距离或完全连接（complete linkage)

定义类G_{p}的样本与G_q的样本之间的最长距离为两类之间的距离

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

聚类的基本概念

- **类与类之间的距离**

中心距离

定义类G_p与G_q的中心$\bar{x}_p$与$\bar{x}_q$之间的距离为两类之间的距离

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

聚类的基本概念

- **类与类之间的距离**

平均距离

定义类G_p与G_q任意两个样本之间距离的平均值为两类之间的距离

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

本章目录

原型聚类

- **K-均值算法(K-means)算法概述**

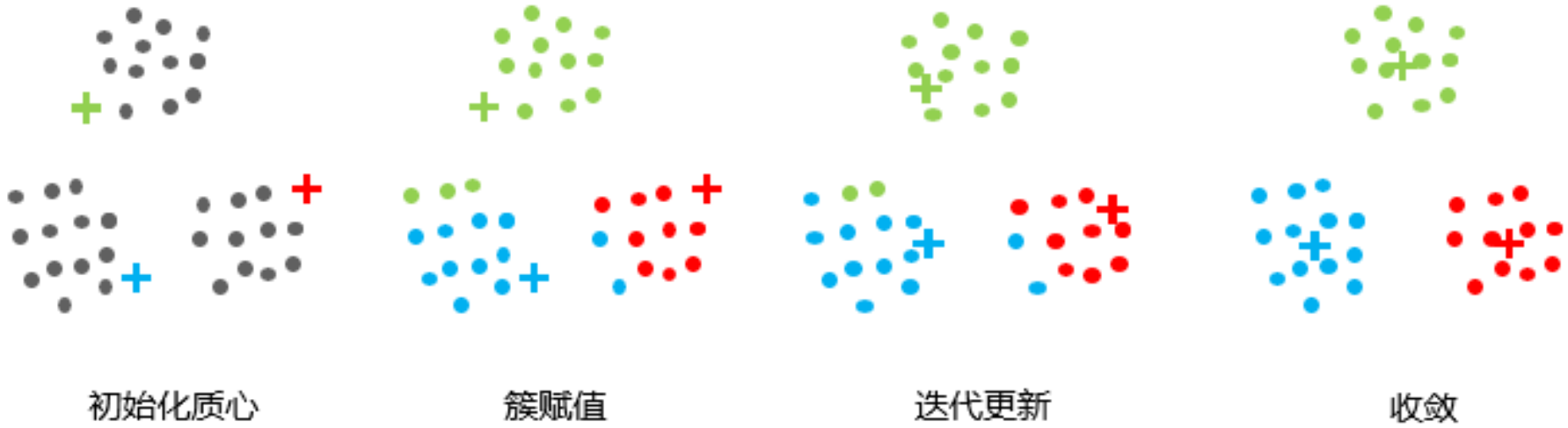
K-means算法是一种**无监督学习**方法，是最普及的聚类算法，算法使用一个**没有标签**的数据集，然后将数据聚类成不同的组。

K-means算法具有一个迭代过程，在这个过程中，数据集被分组成若干个预定义的不重叠的聚类或子组，使簇的内部点尽可能相似，同时试图保持簇在不同的空间，它将数据点分配给簇，以便**簇的质心和数据点之间的平方距离之和最小**，在这个位置，簇的质心是簇中数据点的算术平均值。

原型聚类

- **K-means算法流程**

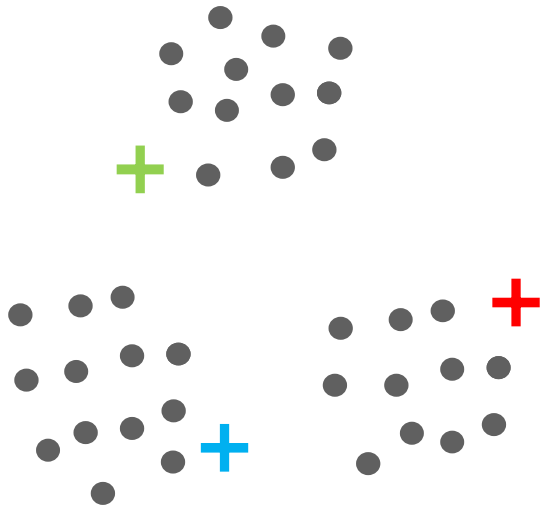
1. 选择K个点作为初始质心。
2. 将每个点指派到最近的质心，形成K个簇。
3. 对于上一步聚类的结果，进行平均计算，得出该簇的新的聚类中心。
4. 重复上述两步/直到迭代结束：质心不发生变化。



原型聚类

- **K-means算法流程**

首先，初始化称为簇质心的任意点。初始化时，必须注意簇的质心必须小于训练数据点的数目。因为该算法是一种迭代算法，接下来的两个步骤是迭代执行的。

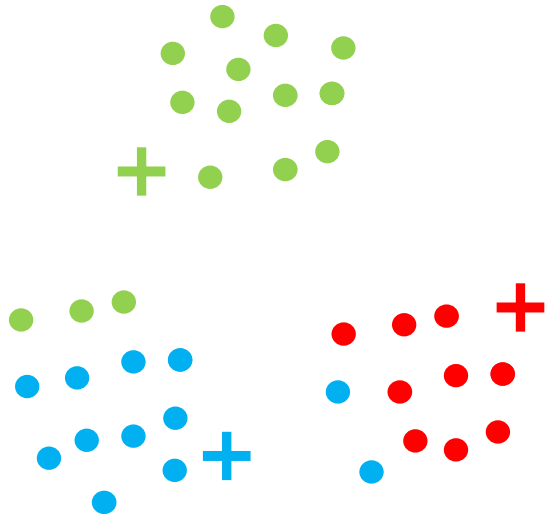


初始化质心

原型聚类

- **K-means算法流程**

第二步：初始化后，遍历所有数据点，计算所有质心与数据点之间的距离。现在，这些簇将根据与质心的最小距离而形成。在本例中，数据分为3个簇($K = 3$)。

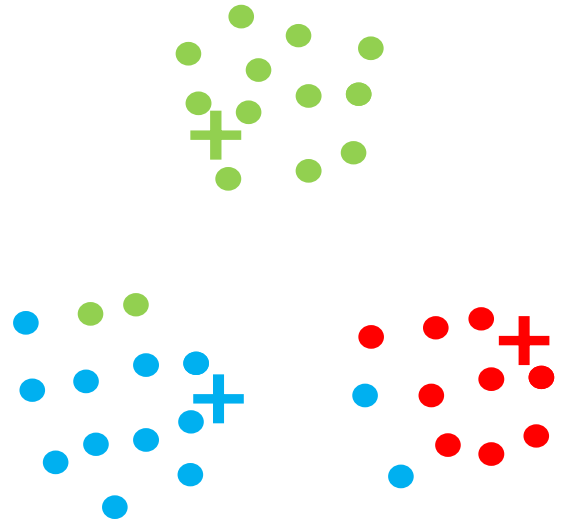


簇赋值

原型聚类

- **K-means算法流程**

第三步：移动质心，因为上面步骤中形成的簇没有优化，所以需要形成优化的簇。为此，我们需要迭代地将质心移动到一个新位置。取一个簇的数据点，计算它们的平均值，然后将该簇的质心移动到这个新位置。对所有其他簇重复相同的步骤。



迭代更新

原型聚类

- **K-means算法流程**

优化

上述两个步骤是迭代进行的，直到质心停止移动，即它们不再改变自己的位置，并且成为静态的。一旦这样做，k-均值算法被称为收敛。

原型聚类

- **K-means算法流程**

设训练集为：$\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$，簇划分$C = \{C_1, C_2, \cdots, C_K\}$，用$\mu_1, \mu_2, \ldots, \mu_K$来表示聚类中心

其中$\mu_{c^{(i)}}$代表与$x^{(i)}$最近的聚类中心点。

我们的的优化目标便是找出使得代价函数最小的 $c^{(1)}$,$c^{(2)}$,...,$c^{(m)}$和 $\mu_1, \mu_2, \ldots, \mu_K$ 。

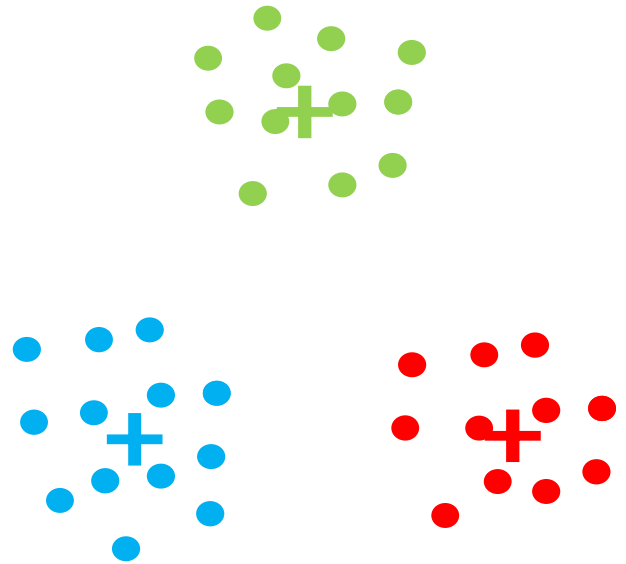
K-均值的代价函数（又称**畸变函数 Distortion function**）为：

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \left\| X^{(i)} - \mu_{c^{(i)}} \right\|^2$$

原型聚类

- **K-means算法流程**

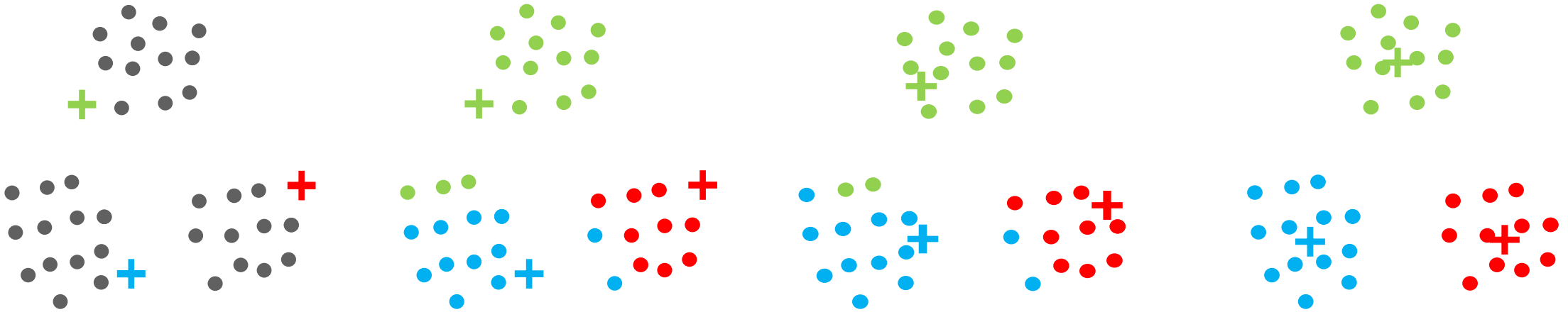
现在，这个算法已经收敛，形成了清晰可见的不同簇。该算法可以根据簇在第一步中的初始化方式给出不同的结果。



收敛

原型聚类

- K-means算法流程总结

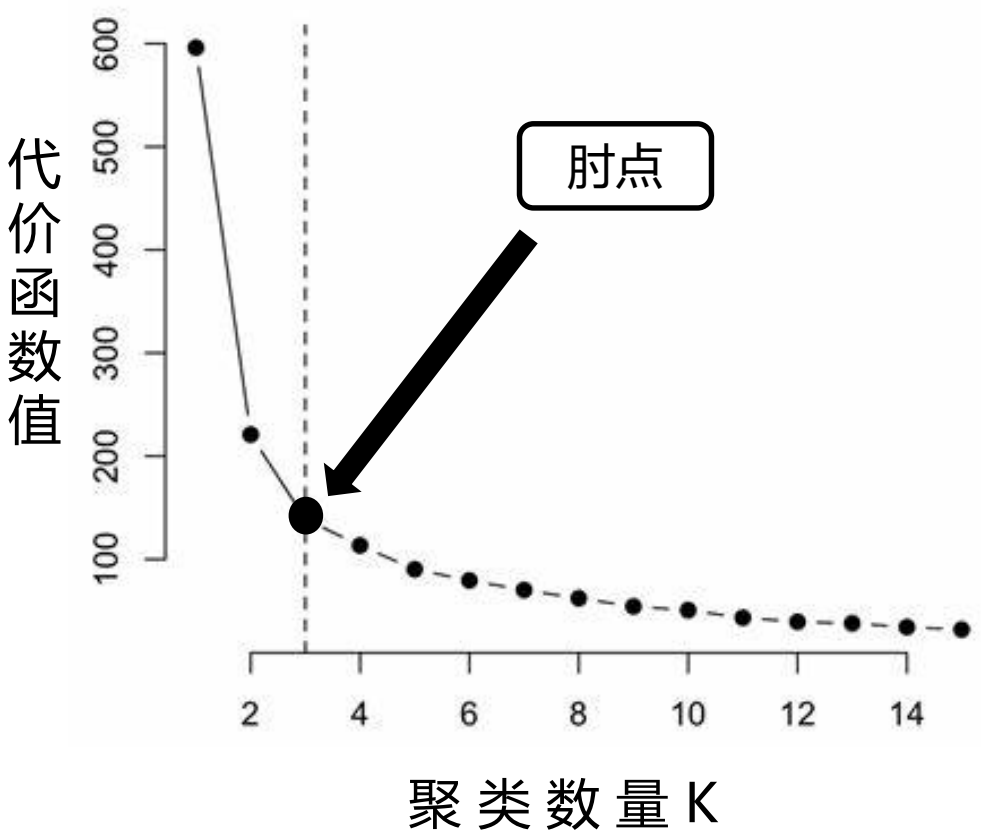


初始化质心　　簇赋值　　迭代更新　　收敛

原型聚类

- **K值的选择**

现在我们需要找到簇的数量。通常通过“肘部法则”进行计算。我们可能会得到一条类似于人的肘部的曲线。右图中，代价函数的值会迅速下降，在$K = 3$的时候达到一个肘点。在此之后，代价函数的值会就下降得非常慢，所以，我们选择$K = 3$。这个方法叫“肘部法则”。



K-均值的一个问题在于，它有可能会停留在一个局部最小值处，而这取决于初始化的情况。

为了解决这个问题，我们通常需要多次运行**K-均值**算法，每一次都重新进行随机初始化，最后再比较多次运行**K-均值**的结果，选择代价函数最小的结果。

原型聚类

- **K-means的优点**

  原理比较简单，实现也是很容易，收敛速度快。

  聚类效果较优。

  算法的可解释度比较强。

  主要需要调参的参数仅仅是簇数K。

原型聚类

- **K-means的缺点**
  - 需要预先指定簇的数量；
  - 如果有两个高度重叠的数据，那么它就不能被区分，也不能判断有两个簇；
  - 欧几里德距离可以不平等的权重因素，限制了能处理的数据变量的类型；
  - 有时随机选择质心并不能带来理想的结果；
  - 无法处理异常值和噪声数据；
  - 不适用于非线性数据集；
  - 对特征尺度敏感；
  - 如果遇到非常大的数据集，那么计算机可能会崩溃。

原型聚类

- **K-means算法例题**

给定含有5个样本的集合

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

试用k均值聚类算法将样本聚到2个类中。

原型聚类

- **K-means算法例题**

(1) 选择两个样本点作为类的中心，假设选择$m_1^{(0)} = x_1 = (0,2)^{\mathrm{T}}$, $m_2^{(0)} = x_2 = (0,0)^{\mathrm{T}}$

(2) 以$m_1^{(0)}$, $m_2^{(0)}$为类$G_1^{(0)}$, $G_2^{(0)}$的中心，计算$x_3 = (1,0)^{\mathrm{T}}$, $x_4 = (5,0)^{\mathrm{T}}$, $x_4 = (5,2)^{\mathrm{T}}$与$m_1^{(0)} = (0,2)^{\mathrm{T}}$， $m_2^{(0)} = (0,0)^{\mathrm{T}}$的欧式距离平方。

对$x_3 = (1,0)^{\mathrm{T}}, d\left(x_3, m_1^{(0)}\right) = 5, d\left(x_3, m_2^{(0)}\right) = 1$，将$x_3$分到类$G_2^{(0)}$。

对$x_4 = (5,0)^{\mathrm{T}}, d\left(x_4, m_1^{(0)}\right) = 29, d\left(x_4, m_2^{(0)}\right) = 25$，将$x_4$分到类$G_2^{(0)}$。

对$x_5 = (5,2)^{\mathrm{T}}, d\left(x_5, m_1^{(0)}\right) = 25, d\left(x_5, m_2^{(0)}\right) = 29$，将$x_5$分到类$G_1^{(0)}$。

原型聚类

- **K-means算法例题**

(3) 得到新的类$G_1^{(1)}=\{x_1,x_5\}$，$G_2^{(1)}=\{x_2,x_3,x_4\}$，计算类的中心$m_1^{(1)}$，$m_2^{(1)}$：

$$m_1^{(1)}=(2.5,2.0)^{\mathrm{T}}，\ m_2^{(1)}=(2,0)^{\mathrm{T}}$$

(4) 重复步骤(2)和步骤(3)。

将x_1分到类$G_1^{(1)}$，将x_2分到类$G_2^{(1)}$，x_3分到类$G_2^{(1)}$，x_4分到类$G_2^{(1)}$，x_5分到类$G_1^{(1)}$

得到新的类$G_1^{(2)}=\{x_1,x_5\}$，$G_2^{(2)}=\{x_2,x_3,x_4\}$。

由于得到的新的类没有改变，聚类停止。得到聚类结果：

$$G_1^{*}=\{x_1,x_5\}，\ G_2^{*}=\{x_2,x_3,x_4\}$$

原型聚类

- **学习向量量化**

与一般聚类算法不同的是，学习向量量化(LVQ)假设数据样本带有类别标记，学习过程中利用样本的这些监督信息来辅助聚类。

给定样本集$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$，$y_j \in \mathcal{Y}$是样本$x_j$的类别标记。LVQ的目标是学得一组$n$维原型向量$\{p_1, p_2, \dots, p_q\}$，每个原型向量代表一个聚类簇，簇标记$t_i \in \mathcal{Y}$。

原型聚类

- **学习向量量化-算法伪代码：**

输入： 样本集$D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$;
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$\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_q\}$
2: **repeat**
3: 从样本集D随机选取样本$(\boldsymbol{x}_j, y_j)$;
4: 计算样本$\boldsymbol{x}_j$与$\boldsymbol{p}_i$ $(1 \leq i \leq q)$的距离：$d_{ji} = ||\boldsymbol{x}_j - \boldsymbol{p}_i||_2$;
5: 找出与$\boldsymbol{x}_j$距离最近的原型向量; $i^* = \arg\min_{i \in \{1,2,\ldots,q\}} d_{ji}$;
6: **if** $y_j = t_{i^*}$ **then**
7: $\boldsymbol{p}' = \boldsymbol{p}_{i^*} + \eta \cdot (\boldsymbol{x}_j - \boldsymbol{p}_{i^*})$
8: **else**
9: $\boldsymbol{p}' = \boldsymbol{p}_{i^*} - \eta \cdot (\boldsymbol{x}_j - \boldsymbol{p}_{i^*})$
10: **end if**
11: 将原型向量$\boldsymbol{p}_{i^*}$更新为$\boldsymbol{p}'$
12: **until** 满足停止条件
13: **return** 当前原型向量

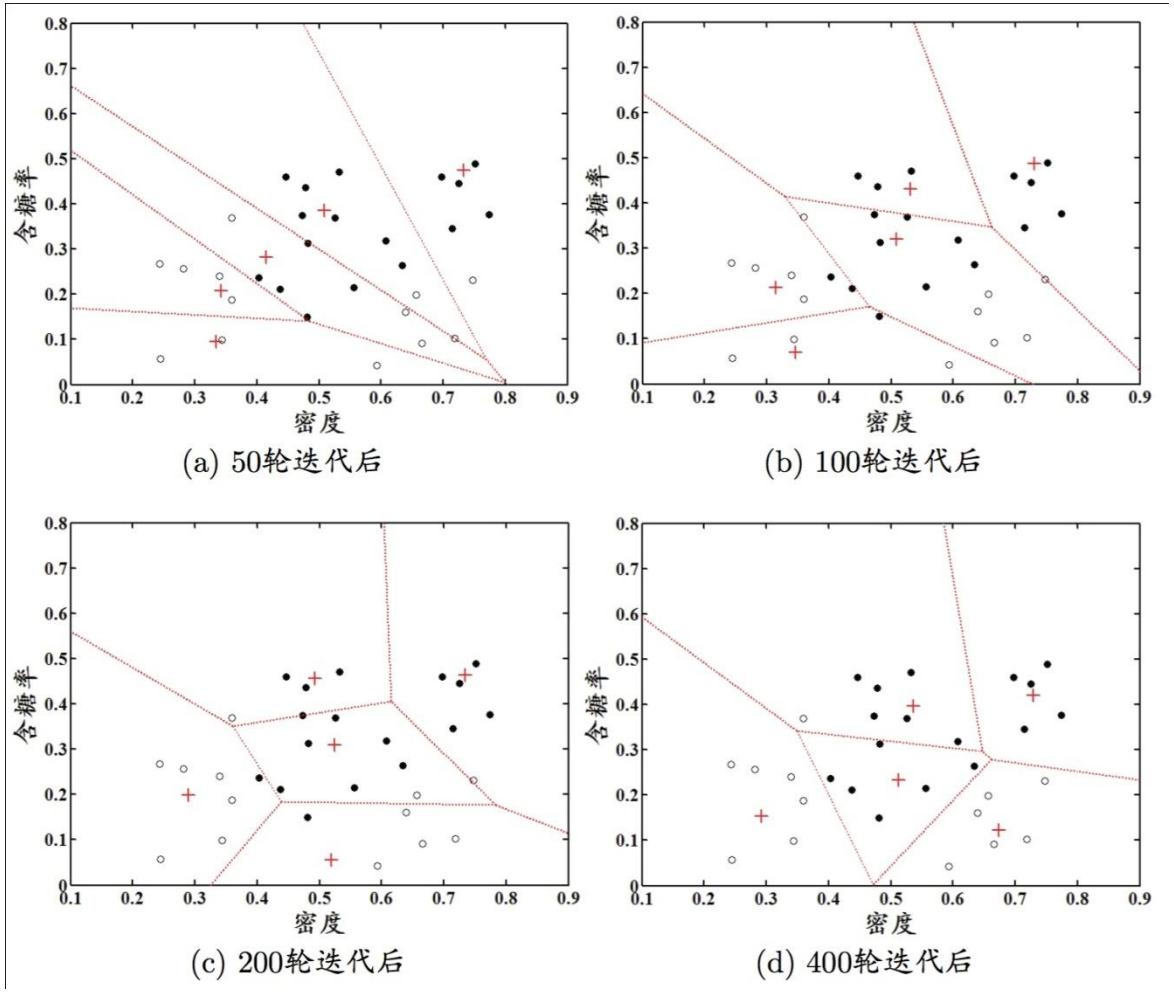
输出： 原型向量$\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_q\}$

算法第2~12行对原型向量进行迭代优化。在每一次迭代中，算法随机选取一个有标记训练样本，找出与其距离最近的原型向量，并根据两者的类别标记是否一致来对原型向量进行相应的更新。

在第12行中，若算法的停止条件已满足（例如以达到最大迭代数，或原型向量更新很小甚至不再更新），则将当前原型向量作为最终结果返回。

原型聚类

- **学习向量量化-聚类效果：**



“+”代表原型向量，红色虚线代表原型向量对样本空间的簇划分，划分区域中每个样本与原型向量的距离不大于样本与其它原型向量的距离。

原型聚类

- **高斯混合聚类**

与k均值、LVQ用原型向量来刻画聚类结构不同，高斯混合聚类（Mixture-of-Gaussian）采用概率模型来表达聚类原型：

回顾多元高斯分布的定义：

对n维样本空间中的随机向量x，若x服从高斯分布，其概率密度函数为

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

其中μ是n维均值向量，Σ是$n \times n$的协方差矩阵。也可将概率密度函数记作$p(x|\mu,\Sigma)$。

原型聚类

- **高斯混合聚类**

我们可定义**高斯混合分布**：

$$p_M(x) = \sum_{i=1}^{k} \alpha_i p(x \mid \mu_i, \Sigma_i)$$

该分布由k个混合分布组成，每个混合成分对应一个高斯分布。其中，μ_i与Σ_i是第i个高斯混合成分的参数。而$\alpha_i > 0$为相应的“混合系数”，$\sum_{i=1}^{k} \alpha_i = 1$。

假设样本的生成过程由高斯混合分布给出：首先，根据$\alpha_1, \alpha_2, \ldots, \alpha_k$定义的先验分布选择高斯混合成分，$\alpha_i$为选择第$i$个混合成分的概率；然后，根据被选择的混合成分的概率密度函数进行采样，从而生成相应的样本。

原型聚类

- **高斯混合聚类**

若训练集$D = \{x_1, x_2, \ldots, x_m\}$由上述过程生成，令随机变量$z_j \in \{1,2, \ldots, k\}$表示生成样本的$x_j$的高斯混合成分，其取值未知。显然，$z_j$的先验概率$\mathrm{P}(z_j = i)$对应于$\alpha_i (\mathrm{i} = 1,2, \ldots, \mathrm{k})$。根据贝叶斯定理，$z_j$的后验分布对应于

$$p_M(z_j = i \mid x_j) = \frac{P(z_j = i) \cdot p_M(x_j \mid z_j = i)}{p_M(x_j)}$$

$$= \frac{\alpha_i \cdot p(x_j | \mu_i, \Sigma_i)}{\sum_{\mathrm{l}=1}^{k} \alpha_l \cdot p(x_j | \mu_l, \Sigma_l)}$$

原型聚类

- **高斯混合聚类**

换言之，$p_M(z_j = i \mid x_j)$给出了样本x_j由第i个高斯混合成分生成的后验概率。为了方便叙述，将其简记为$\gamma_{ji}(i = 1,2,\ldots,k)$.

当高斯混合分布已知时，高斯混合聚类将样本集D划分为k个簇$C = \{C_i, C_2, \ldots, C_k\}$，每个样本$x_j$的簇标记为$\lambda_j$如下确定：

$$\lambda_j = \underset{i \in \{1,2,\ldots,k\}}{\arg\max}\ \gamma_{ji}$$

原型聚类

- **高斯混合聚类-模型求解**

那么对于高斯混合分布$p_M(x) = \sum_{i=1}^{k} \alpha_i p(x \mid \mu_i, \Sigma_i)$，模型参数$\{(\alpha_i, \mu_i, \Sigma_i) | 1 \leq i \leq k\}$如何求解？显然，给定样本集$D$，可用极大似然估计：

$$LL(D) = \ln\left(\prod_{j=1}^{m} p_M(x_j)\right)$$

$$= \sum_{j=1}^{m} \ln\left(\sum_{i=1}^{k} \alpha_i \cdot p(x_j \mid \mu_i, \Sigma_i)\right)$$

令：$\dfrac{\partial LL(D)}{\partial \mu_i} = 0$ ⟶ $\mu_i = \dfrac{\sum_{j=1}^{m} \gamma_{ji} x_j}{\sum_{j=1}^{m} \gamma_{ji}}$

原型聚类

- **高斯混合聚类-模型求解**

令：$\frac{\partial LL(D)}{\partial \Sigma_i} = 0$ ⟶ $\Sigma_i = \frac{\sum_{j=1}^{m} \gamma_{ji}\left(x_j - \mu_i\right)\left(x_j - \mu_i\right)^T}{\sum_{j=1}^{m} \gamma_{ji}}$

对于混合系数α_i，除了要最大化$LL(D)$，还需满足$\alpha_i \geq 0$，$\sum_{i=1}^{k} \alpha_i = 1$。

考虑$LL(D)$的拉格朗日形式：

$$LL(D) + \lambda\left(\sum_{i=1}^{k} \alpha_i - 1\right)$$

其中λ 为拉格朗日乘子，上式对α_i 的导数为0，求得：$\alpha_i = \frac{1}{m}\sum_{j=1}^{m} \gamma_{ji}$

即每个高斯成分的混合系数由样本属于该成分的平均后验概率确定。

原型聚类

- **高斯混合聚类-算法伪代码：**

输入： 样本集$D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$;
高斯混合成分个数k.

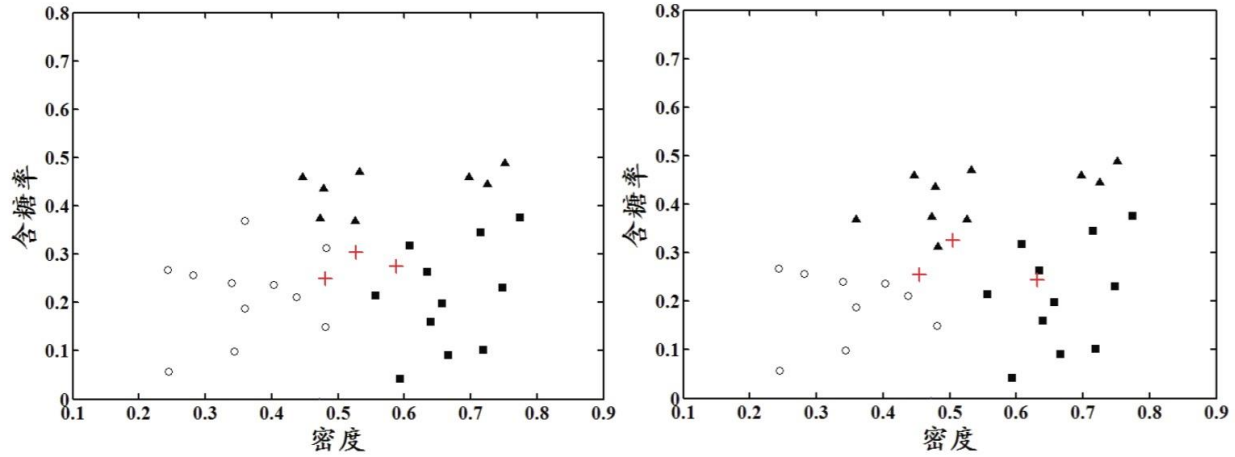
过程：

初始化高斯混合分布的模型参数$\{(\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\mid 1\leq i\leq k\}$
repeat
  for $j=1,\ldots,m$ **do**
    根据(9.30)计算$\boldsymbol{x}_j$由各混合成分生成的后验概率, 即
    $\gamma_{ji}=p_{\mathcal{M}}(z_j=i\mid\boldsymbol{x}_j)\ (1\leq i\leq k)$
  end for
  for $i=1,\ldots,k$ **do**
    计算新均值向量：$\boldsymbol{\mu}_i'=\frac{\sum_{j=1}^{m}\gamma_{ji}\boldsymbol{x}_j}{\sum_{j=1}^{m}\gamma_{ji}}$;
    计算新协方差矩阵：$\boldsymbol{\Sigma}_i'=\frac{\sum_{j=1}^{m}\gamma_{ji}(\boldsymbol{x}_j-\boldsymbol{\mu}_i')(\boldsymbol{x}_j-\boldsymbol{\mu}_i')^{\top}}{\sum_{j=1}^{m}\gamma_{ji}}$;
    计算新混合系数：$\alpha_i'=\frac{\sum_{j=1}^{m}\gamma_{ji}}{m}$;
  end for
  将模型参数$\{(\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\mid 1\leq i\leq k\}$更新为$\{(\alpha_i',\boldsymbol{\mu}_i',\boldsymbol{\Sigma}_i')\mid 1\leq i\leq k\}$
until 满足停止条件
$C_i=\emptyset\ (1\leq i\leq k)$
for $j=1,\ldots,m$ **do**
  根据(9.31)确定$\boldsymbol{x}_j$的簇标记λ_j;
  将$\boldsymbol{x}_j$划入相应的簇：$C_{\lambda_j}=C_{\lambda_j}\bigcup\{\boldsymbol{x}_j\}$
end for
return 簇划分结果

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

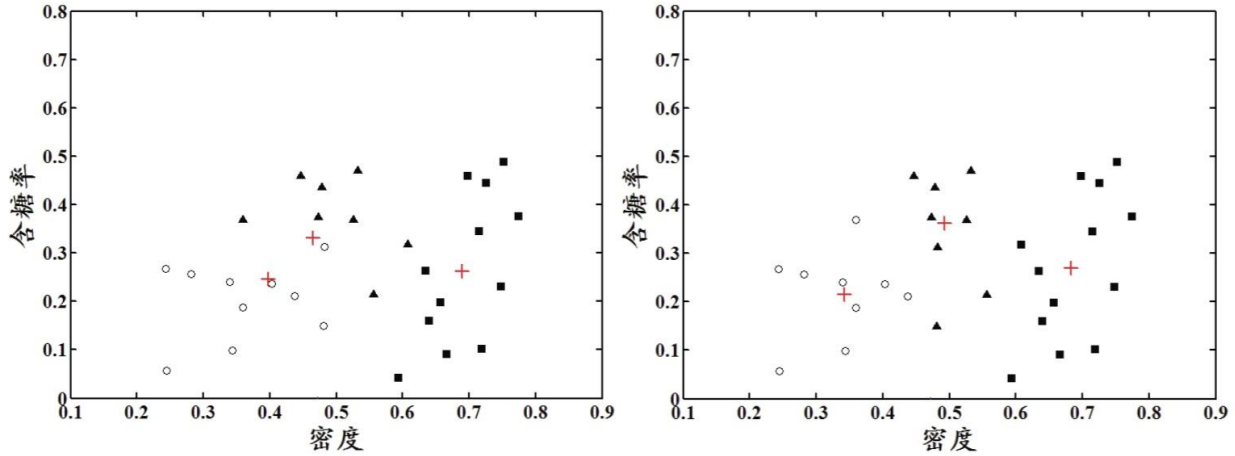
原型聚类

- **聚类效果：**



(a) 5轮迭代后

(b) 10轮迭代后

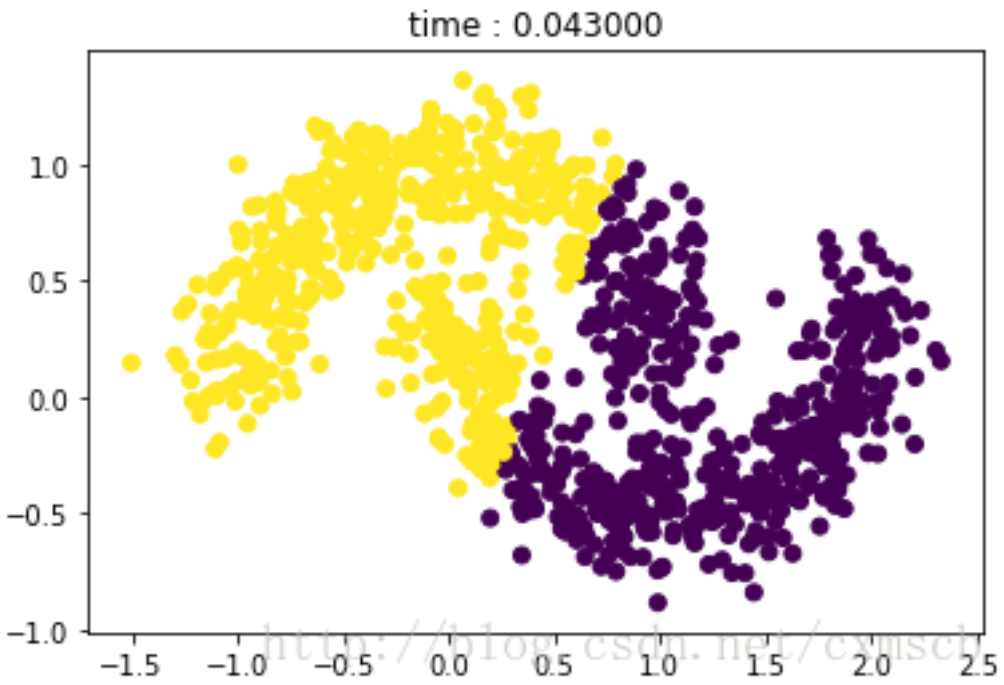


(c) 20轮迭代后

(d) 50轮迭代后

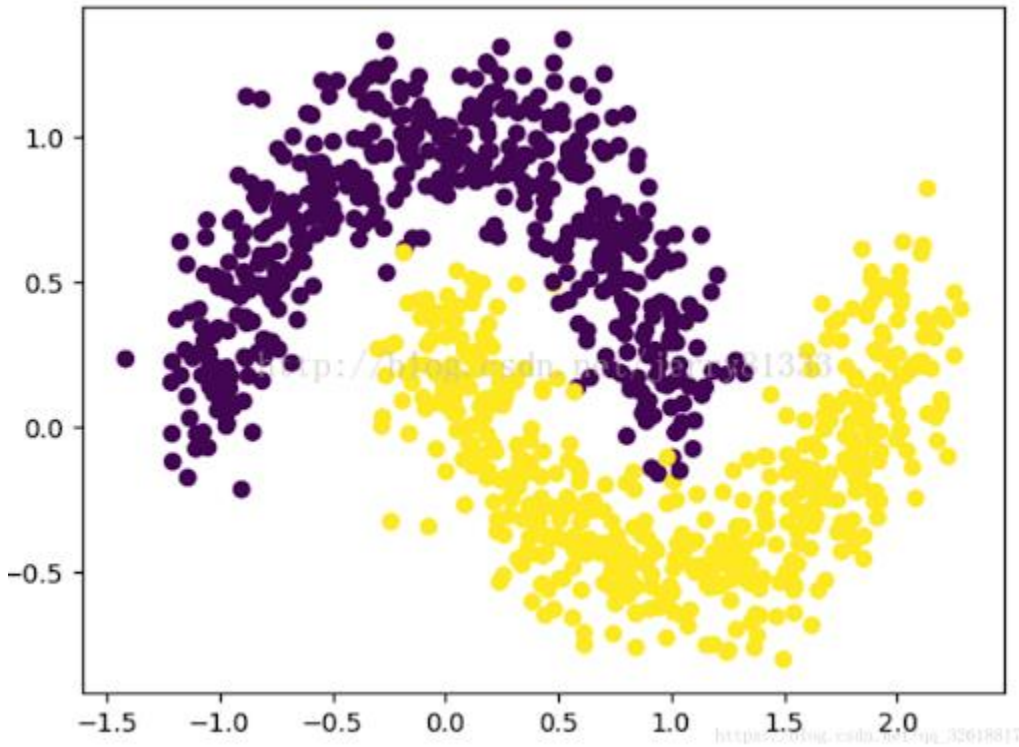
本章目录

密度聚类



使用k均值方法对下图数据进行聚类，基于原型的聚类方法难以发掘到密度连接的信息，导致聚类结果同直观差异较大：

密度聚类



基于密度的聚类方法：从样本密度的角度考察样本的连接性，使密度相连的样本归结到一个簇，更符合直观认知：

密度聚类

- **密度聚类定义：**

密度聚类也称为“基于密度的聚类” (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来确定。

通常情况下，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来获得最终的聚类结果。

接下来介绍DBSCAN这一密度聚类算法。

密度聚类

- **DBSCAN密度聚类**

与划分和层次聚类方法不同，DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一个比较有代表性的基于密度的聚类算法。它将簇定义为密度相连的点的最大集合，能够把具有**足够高密度的区域划分为簇**，并可**在噪声的空间数据库中发现任意形状的聚类**。

密度：空间中任意一点的密度是以该点为圆心，以**扫描半径**构成的圆区域内包含的点数目。

密度聚类

- **DBSCAN密度聚类**

DBSCAN使用**两个超参数**：

扫描半径 (eps)和最小包含点数(minPts)来获得簇的数量，而不是猜测簇的数目。

Ø **扫描半径 (eps)：**

用于定位点/检查任何点附近密度的距离度量，即扫描半径。

Ø **最小包含点数(minPts) ：**

聚集在一起的最小点数（阈值），该区域被认为是稠密的。

密度聚类

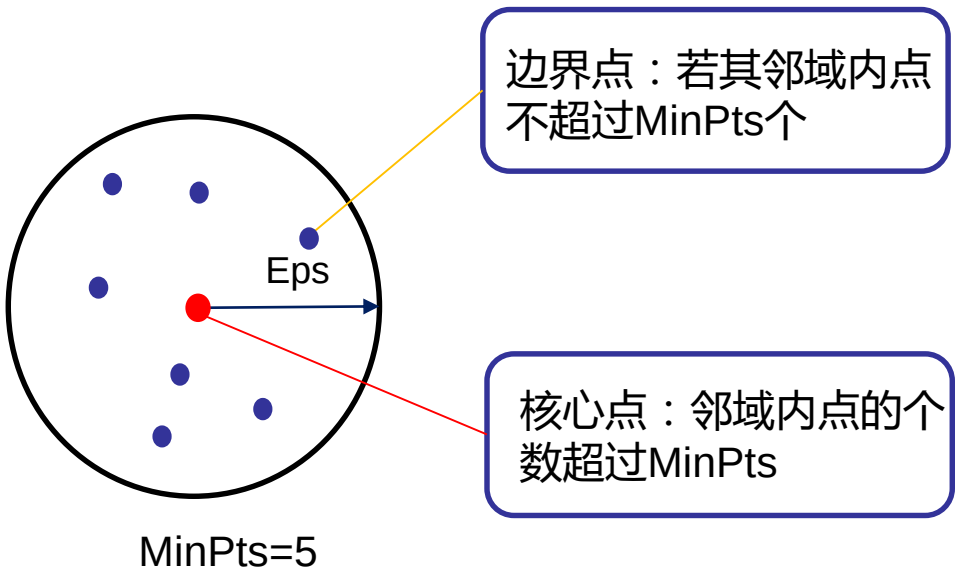
- **DBSCAN密度聚类**

DBSCAN算法将数据点分为三类：

1. **核心点**：在半径Eps内含有超过MinPts数目的点。

2.**边界点**：在半径Eps内点的数量小于MinPts,但是落在核心点的邻域内的点。

3.**噪音点**：既不是核心点也不是边界点的点。



密度聚类

- **DBSCAN密度聚类的算法流程**

1. 将所有点标记为核心点、边界点或噪声点；
2. 如果选择的点是核心点，则找出所有从该点出发的密度可达对象形成簇；
3. 如果该点是非核心点，将其指派到一个与之关联的核心点的簇中；
4. 重复以上步骤，直到所点都被处理过

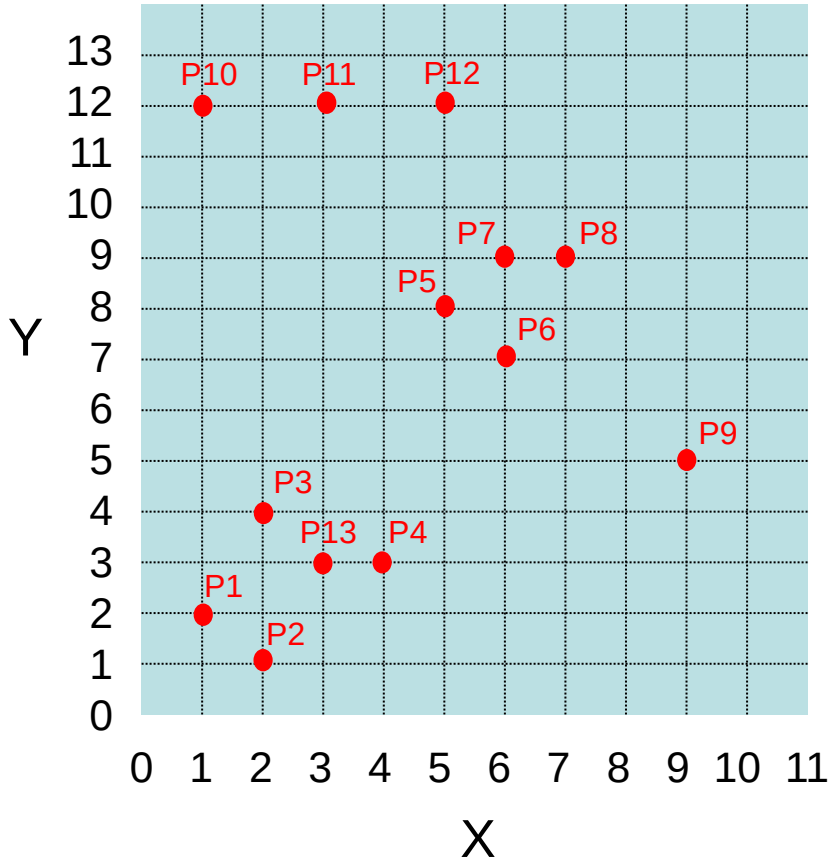
举例：有如下13个样本点，使用DBSCAN进行聚类

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
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

密度聚类

- **DBSCAN密度聚类的算法流程**

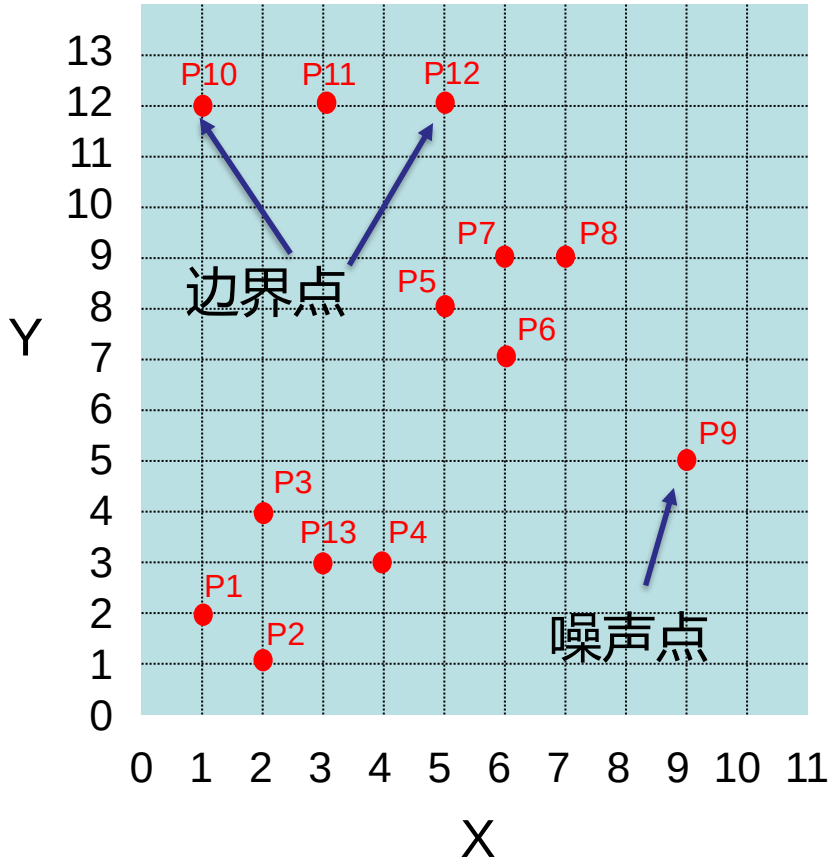
  - 对每个点计算其邻域Eps=3内的点的集合。
  - 集合内点的个数超过MinPts=3的点为核心点。



密度聚类

- **DBSCAN密度聚类的算法流程**

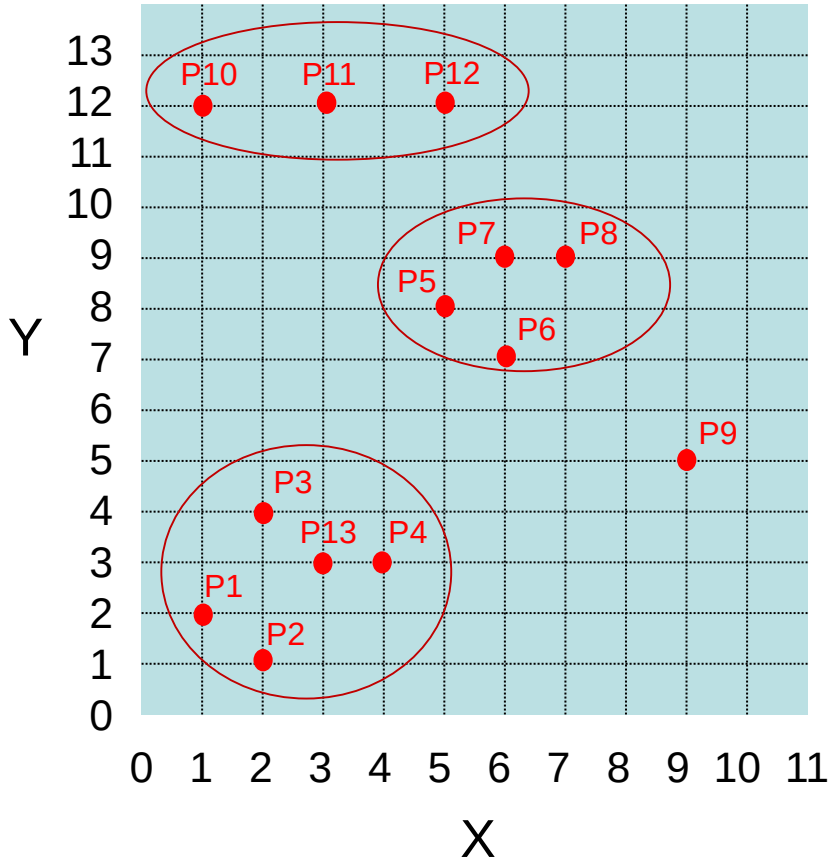
  - 查看剩余点是否在核点的邻域内，若在，则为边界点，否则为噪声点。



密度聚类

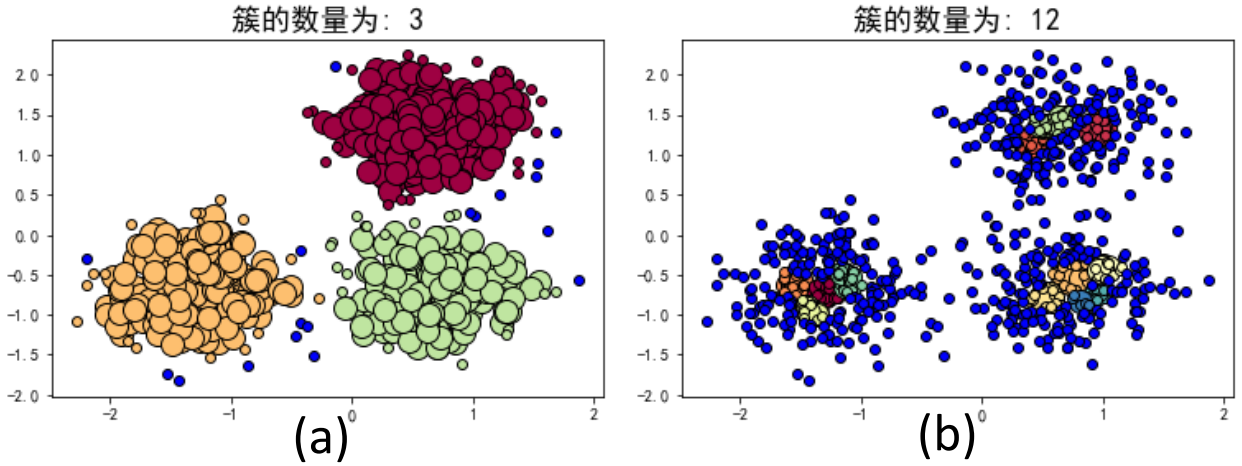
- **DBSCAN密度聚类的算法流程**

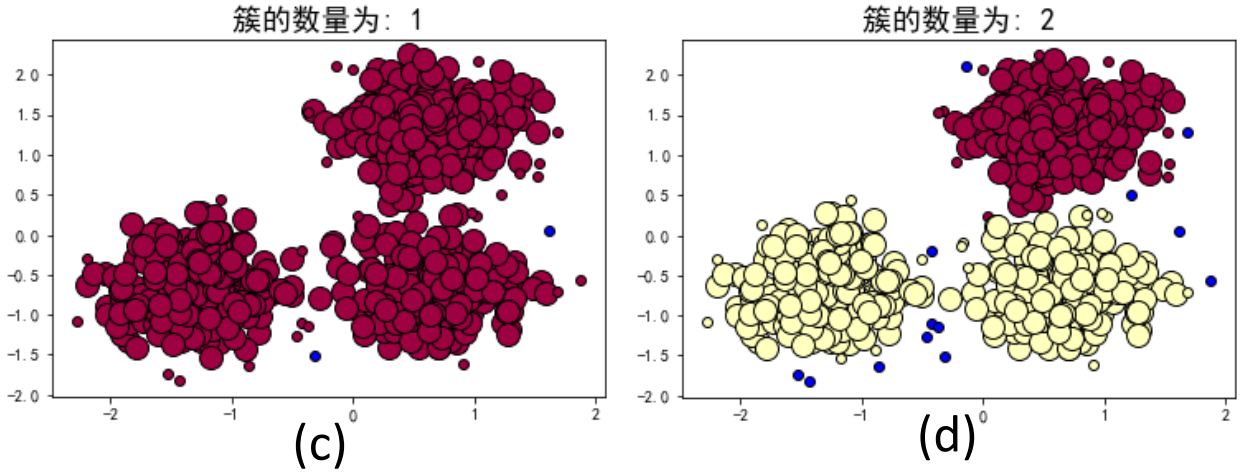
  - 将距离不超过Eps=3的点相互连接，构成一个簇，核心点邻域内的点也会被加入到这个簇中。



密度聚类

- **DBSCAN的超参数**





DBSCAN超参数案例

图片编号		(a)	(b)	(c)	(d)
评价指标	超参数	eps=0.3 minPts=10	eps=0.1 minPts=10	eps=0.4 minPts=10	eps=0.3 minPts=6
估计的簇的数量		3	12	1	2
估计的噪声点		18	516	2	13
同一性		0.9530	0.3128	0.0010	0.5365
完整性		0.8832	0.2489	0.0586	0.8623
V-measure		0.9170	0.0237	0.0020	0.6510
ARI		0.9517	0.2673	0	0.5414
轮廓系数		0.6255	-0.3659	0.0611	0.3845

这个案例中，当：

eps=0.3，minPts=10的时候，

DBSCAN达到最优效果。

本章目录

层次聚类

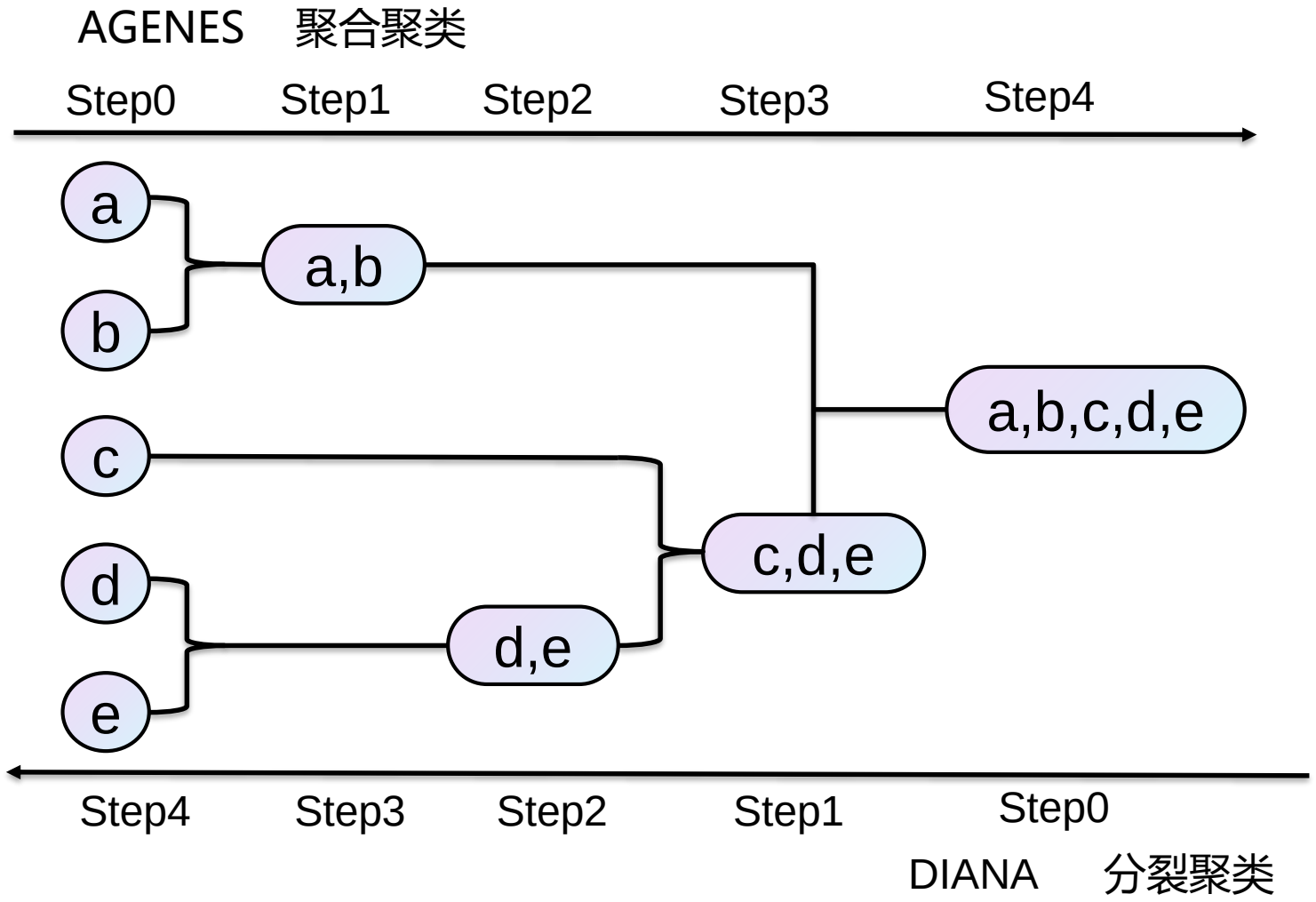
- **层次聚类**

  - 层次聚类假设簇之间存在层次结构，将样本聚到层次化的簇中。
  - 层次聚类又有聚合聚类（自下而上）、分裂聚类（自上而下）两种方法。
  - 因为每个样本只属于一个簇，所以层次聚类属于硬聚类。

背景知识：如果一个聚类方法假定一个样本只能属于一个簇，或簇的交集为空集，那么该方法称为硬聚类方法。

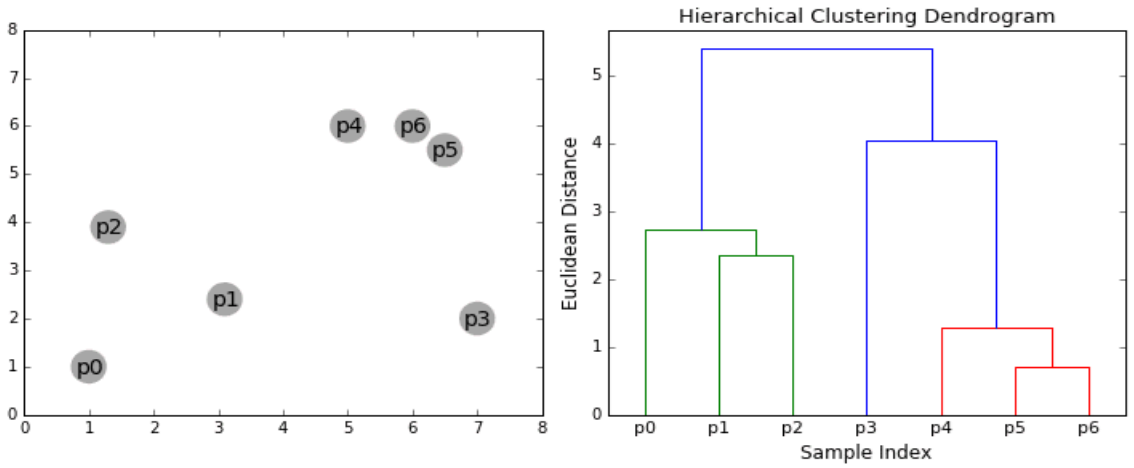
如果一个样本可以属于多个簇，或簇的交集不为空集，那么该方法称为软聚类方法。

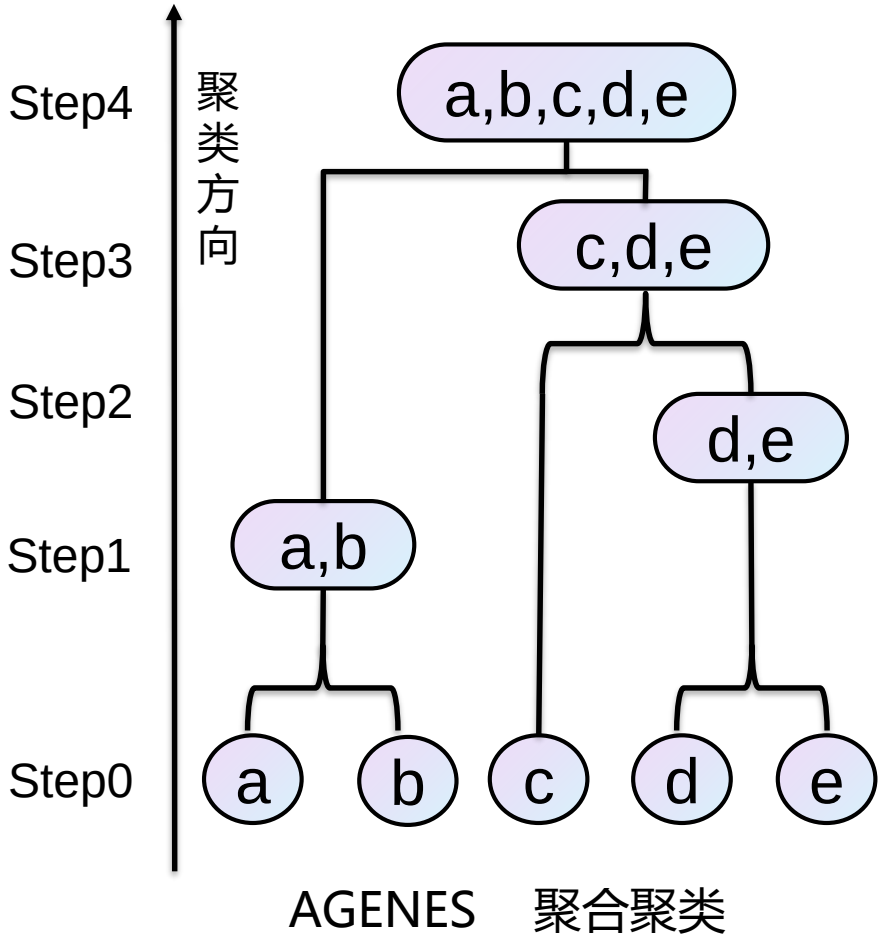
层次聚类


AGENES 聚合聚类
Step0
Step1
Step2
Step3
Step4
a
b
a,b
c
d
e
d,e
c,d,e
a,b,c,d,e
Step4
Step3
Step2
Step1
Step0
DIANA 分裂聚类

层次聚类

- **聚合聚类**
  - 开始将每个样本各自分到一个簇；
  - 之后将相距最近的两簇合并，建立一个新的簇；
  - 重复此操作直到满足停止条件；
  - 得到层次化的类别。





层次聚类

- **聚合聚类例题**

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

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

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。

层次聚类

- **聚合聚类例题**

(1) 首先用5个样本构建5个类， $G_i = \{x_i\}, i = 1,2, \dots, 5$ ，样本之间的距离也就变成类之间的距离，所以5个类之间的距离矩阵亦为D 。

(2) 由矩阵D可以看出， $D_{35} = D_{53} = 1$为最小，所以把G_3和G_5 合并为一 个新类，记作$G_6 = \{x_3, x_5\}$

层次聚类

- **聚合聚类例题**

(3) 计算G_6与G_1,G_2,G_4之间的最短距离，有

$$G_{61} = 2, G_{62} = 5, G_{64} = 5$$

又注意到其余两类之间的距离是

$$D_{12} = 7, D_{14} = 9, D_{24} = 4$$

显然， $G_{61} = 2$ 最小，所以将G_1与G_6合并成一个新类，记作

$$G_7 = \{x_1, x_3, x_5\}$$

层次聚类

- **聚合聚类例题**

(4) 计算G_7与G_2, G_4 之间的最短距离，

$$G_{72} = 5, G_{74} = 5$$

又注意到

$$D_{24} = 4$$

显然，其中D_{24}=4最小，所以将G_2与G_4合并成一个新类，记作

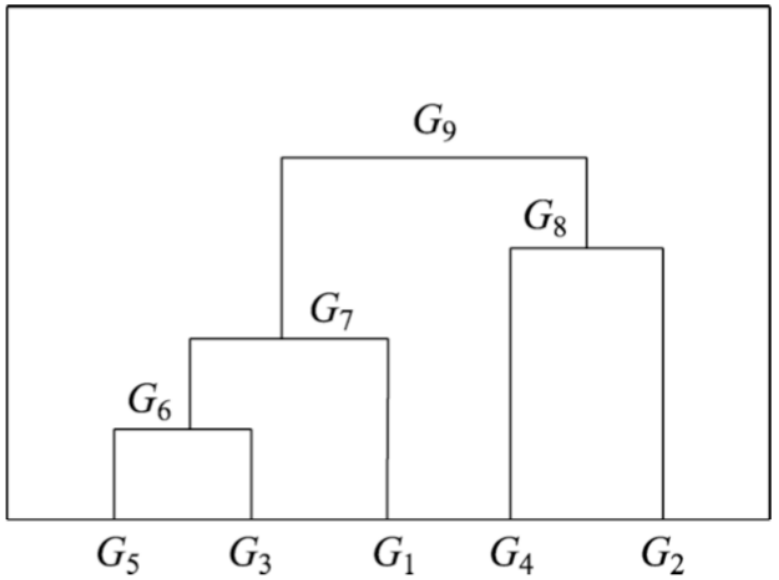
$$G_8 = \{x_2, x_4\}$$

层次聚类

- **聚合聚类例题**

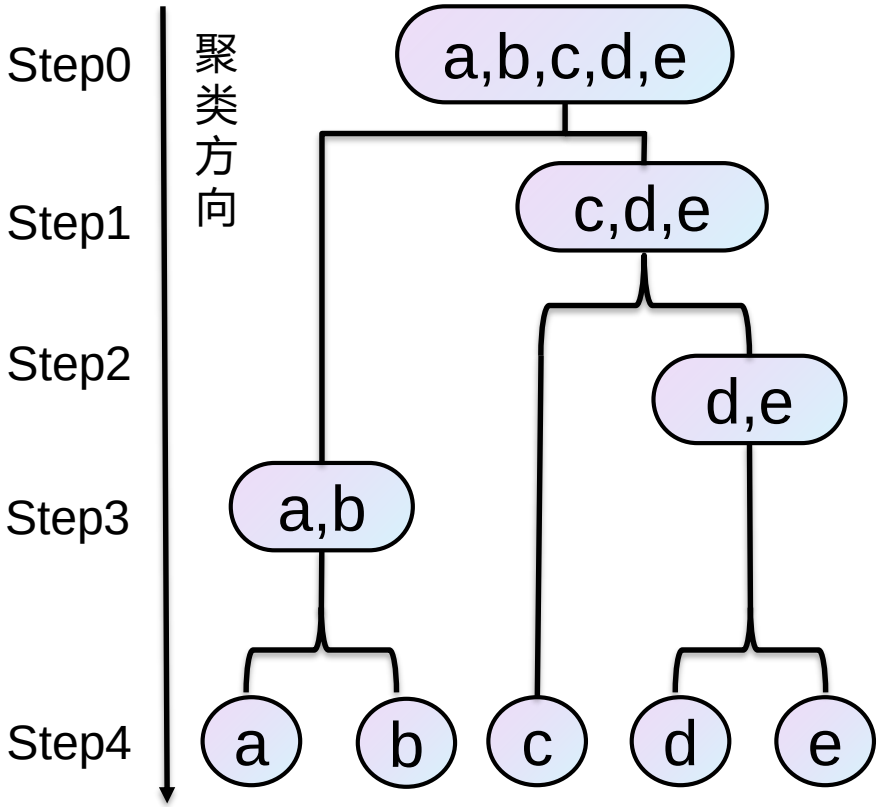
(5) 将G_7与G_8合并成一个新类，记作$G_9 = \{x_1, x_2, x_3, x_4, x_5\}$

即将全部样本聚成1类，聚类终止



层次聚类

- **分裂聚类**
  - 开始将所有样本分到一个簇；
  - 之后将已有类中相距最远的样本分到两个新的簇；
  - 重复此操作直到满足停止条件；
  - 得到层次化的类别。



DIANA 分裂聚类

参考文献

[1] Wong J A H A . Algorithm AS 136: A K-Means Clustering Algorithm[J]. Journal of the Royal Statistical Society, 1979, 28(1):100-108.

[2] Ester M . A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[J]. Proc.int.conf.knowledg Discovery & Data Mining, 1996.

[3] Andrew Ng. Machine Learning[EB/OL]. Stanford University,2014. https://www.coursera.org/course/ml

[4] 李航. 统计学习方法[M]. 清华大学出版社,2019.

[5] 周志华. 机器学习[M]. 清华大学出版社,2016.

[6] Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning[M]. Springer, New York, NY, 2001.

参考文献

[7] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191):1492.

[8] CHRISTOPHER M. BISHOP. Pattern Recognition and Machine Learning[M]. Springer,2006.

[9] Rosenberg A, Hirschberg J. V-Measure: A conditional entropy-based external cluster evaluation[C]// Conference on Emnlp-conll. DBLP, 2007.

[10] Campello R J G B, Moulavi D, Zimek A, et al. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection[J]. Acm Transactions on Knowledge Discovery from Data, 2015.

[11] 彭涛.人工智能概论[EB/OL].北京联合大学,2020. https://www.icourse163.org/course/BUU-1461546165

谢 谢！