机器学习

李成龙

安徽大学人工智能学院
“多模态认知计算”安徽省重点实验室
合肥综合性国家科学中心人工智能研究院

内容安排

- 什么是机器学习
- 机器如何学习
- 如何让机器学习的更好
- 为什么机器能学习

内容安排

- **机器如何学习**
  - 有监督学习
    - 感知机
    - 支持向量机
    - 朴素贝叶斯分类
    - 决策树
    - 集成学习（Bagging算法与随机森林、Boosting算法）
    - 线性回归
    - 逻辑回归
    - Softmax回归
    - 神经网络与深度学习
  - 无监督学习
    - 聚类
    - 主成分分析

本节目录

本节目录

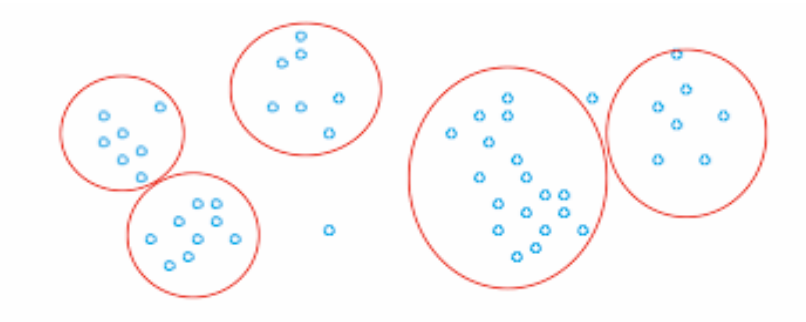
- 划分聚类
- 密度聚类

划分聚类

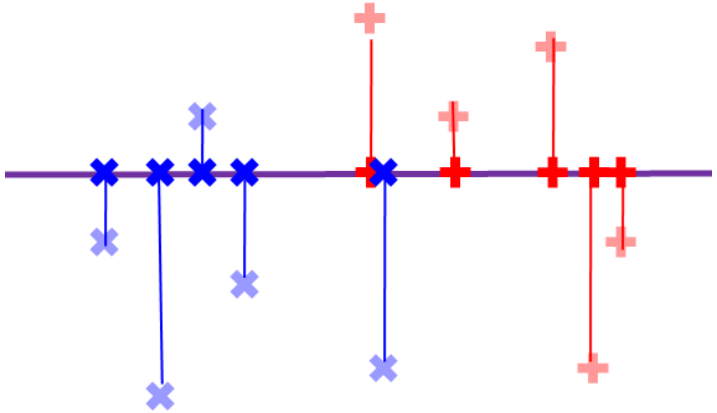
- 机器学习类型

划分聚类

- **无监督学习**
  - **无监督学习**通过比较样本之间的某种联系实现对样本的数据分析。最大特点是学习算法的输入是无标记样本



聚类



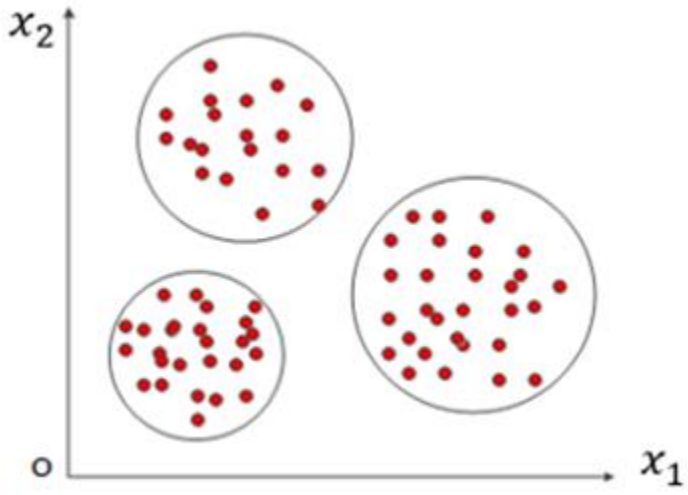
降维

划分聚类

- **基本思想**
  - 对样本数据进行划分，实现对样本数据的聚类分析
    - 确定划分块的个数即聚簇的个数
    - 通过适当方式将样本数据聚集成指定个数的聚簇
  - 常用的划分型聚类算法：k-均值聚类、模糊c-均值聚类
    - 通过使用样本数据的均值确定各聚簇的聚类中心
    - 通过计算各样本数据到各聚簇聚类中心的某种距离实现对样本数据之间的相似性度量

划分聚类

- **k-均值聚类**
  - 基本思想：同类样本在特征空间中应该相距不远
  - 主要方法：将集中在特征空间某一区域内的样本划分为同一个簇
  - 区域位置的界定主要通过样本特征值的均值确定



划分聚类

- **k–均值聚类**
  - 通常用欧式距离（2–范数）或曼哈顿距离（1–范数）等范数度量两个示例样本之间的距离
  - 对于给定的示例样本数据集D：

$$D = \{X_1, X_2, \dots, X_n\}$$

其中每个示例样本分别具有m个特征，即$X_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$，以欧式距离为例，$k$–均值聚类算法对$D$中数据点进行聚类的具体过程如下

划分聚类

- **k–均值聚类**
  - 假设按照某种方式将数据集D中所有示例样本划分为k个簇$C_1, C_2, \dots, C_k$，则与该划分相对应的类内距离$d(C_1, C_2, \dots, C_k)$为：

$$d(C_1, C_2, \dots, C_k) = \sum_{j=1}^{k} \sum_{X_i \in C_j} \left(\sum_{t=1}^{m} (x_{it} - u_{jt})^2 \right)^{\frac{1}{2}}$$

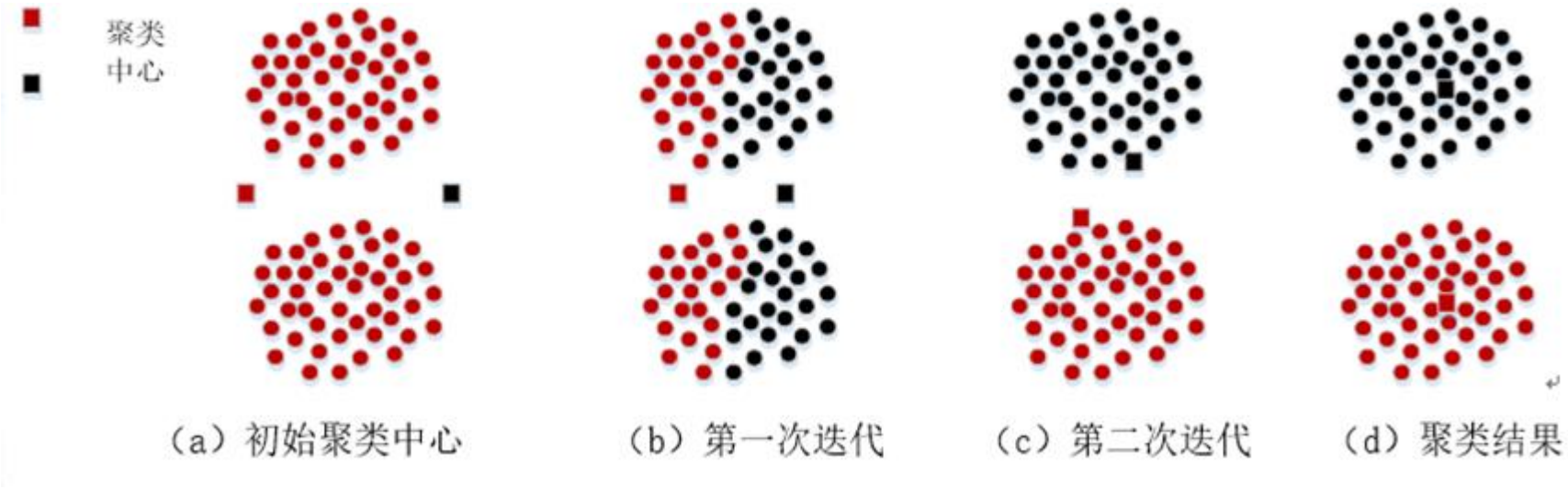
其中u_{jt}表示为第j个簇C_j聚类中心U_j的第t个坐标分量

划分聚类

- k–均值聚类
  - 1：令$s=0$，并从D中随机生成k个作为初始聚类中心的数据点$u_1^0, u_2^0, \ldots, u_k^0$；
  - 2：计算D中各样本与各簇中心之间的距离w，并根据w值将其分别划分到簇中心点与其最近的簇中；
  - 3：分别计算各簇中所有示例样本数据的均值，并分别将每个簇所得到的均值作为该簇新的聚类中心$u_1^{s+1}, u_2^{s+1}, \ldots, u_k^{s+1}$；
  - 4：若$u_j^{s+1}=u_j^s$，则终止算法并输出最终簇，否则令$s=s+1$，并返回步骤2

划分聚类

- **k–均值聚类**
  - 下图展示了k–均值算法从选择初始聚类中心经过迭代到收敛的过程



划分聚类

例题：表为某机构15支足球队在2017-2018年间的积分，各队在各赛事中的水平发挥有所不同。若将球队的水平分为三个不同的层次水平，试用k-均值聚类方法分析哪些队伍的整体水平比较相近

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
赛事1	50	28	17	25	28	50	50	50
赛事2	50	9	15	40	40	50	40	40
赛事3	9	4	3	5	2	1	9	9
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
赛事1	40	50	50	50	40	40	50	
赛事2	40	50	50	50	40	32	50	
赛事3	5	9	5	9	9	17	9	

划分聚类

- 由于各队在各赛事上的发挥水平有所不同，故先对积分数据进行归一化处理，使用最小-最大标准化策略将积分数据映射到[0,1]区间内，具体计算公式为：

$$a_i' = \frac{a_i - \min(a_i)}{\max(a_i) - \min(a_i)}$$

- $\min(a_i)$和$\max(a_i)$分别表示第i个属性值a_i在所有球队中的最小值和最大值。
- 使用上述公式对表数据进行归一化计算，得到下表所示的归一化数据

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
赛事1	1	0.3	0	0.24	0.3	1	1	1
赛事2	1	0	0.15	0.76	0.76	1	0.76	0.76
赛事3	0.5	0.19	0.13	0.25	0.06	0	0.5	0.5
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
赛事1	0.7	1	1	1	0.7	0.7	1	
赛事2	0.76	1	1	1	0.76	0.68	1	
赛事3	0.25	0.5	0.25	0.5	0.5	1	0.5	

划分聚类

- 由于需将球队分为3个层次水平，故取聚类的簇数$k=3$。通过随机采样选择编号为2、11、14的三支队伍所对应数据点作为初始聚类中心，即三个簇的聚类中心分别为：
  $\mu_1=(0.3,0,0.19),\mu_2=(0.7,0,76,0.5),\mu_3=(1,1,0.5)$
- 计算每个数据点到聚类中心的欧氏距离，计算结果如下表所示

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
μ_1	1.2594	0	0.3407	0.7647	0.7710	1.2354	1.0787	1.0787
μ_2	0	0.9131	0.9995	0.5235	0.5946	0.6306	0.3000	0.3000
μ_3	0.3407	1.2594	1.3636	0.8353	0.8609	0.5000	0.2400	0.2400
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
μ_1	0.8609	1.2594	1.2221	1.2594	0.9131	1.1307	1.2594	
μ_2	0.2500	0.3842	0.4584	0.3842	0	0.5064	0.3842	
μ_3	0.4584	0	0.2500	0	0.3842	0.6651	0	

划分聚类

- 分别将每个数据点分配到聚类中心与其距离最近的簇中，得到第一次聚类结果为：

$$C_1 = \{X_2, X_3\};\ C_2 = \{X_1, X_4, X_5, X_9, X_{13}, X_{14}\};$$
$$C_3 = \{X_6, X_7, X_8, X_{10}, X_{11}, X_{12}, X_{15}\}$$

- 根据上述第一次聚类结果，对聚类中心做调整。对于C_1，有：

$$\mu_1' = \left(\frac{0.3+0}{2}, \frac{0.15+0}{2}, \frac{0.19+0.13}{2}\right) = (0.15, 0.075, 0.16)$$

- 同理可将第二个簇C_2和第三个簇C_3的聚类中心进行调整，分别得到

$$\mu_2' = (0.528, 0.744, 0.412), \mu_3' = (1, 0.94, 0.40625)$$

- 计算各数据点与更新后的聚类中心的距离，得到如下表所示计算结果

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
μ_1'	1.3014	0.1704	0.1704	0.6967	0.7083	1.2664	1.1434	1.1434
μ_2'	0.5441	0.8092	0.8443	0.3308	0.4197	0.6768	0.4804	0.4804
μ_3'	0.1113	1.1918	1.3040	0.7965	0.8014	0.4107	0.2030	0.2030
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
μ_1'	0.8831	1.3014	1.2595	1.3014	0.9420	1.1722	1.3014	
μ_2'	0.2368	0.5441	0.5609	0.5441	0.1939	0.6160	0.5441	
μ_3'	0.3832	0.1113	0.1674	0.1113	0.3622	0.7142	0.1113	

划分聚类

- 根据表4-4可得到第二次聚类结果如下：$C_1 = \{X_2, X_3\}$；
  $C_2 = \{X_4, X_5, X_9, X_{13}, X_{14}\}$；$C_3 = \{X_1, X_6, X_7, X_8, X_{10}, X_{11}, X_{12}, X_{15}\}$
- 聚类结果并未发生变化，故聚类中心收敛，停止迭代
- 由上述聚类结果可知，X_2、X_3两支球队的整体水平比较相近，
  $X_4, X_5, X_9, X_{13}, X_{14}$的整体水平比较相近，其余球队的整体水平比较相近

划分聚类

- k–均值聚类
  - 算法要求每个样本数据点在一次迭代过程中只能被划分到某个特定的簇中
  - 样本数据并非都满足这种非此即彼的刚性划分

划分聚类

- **模糊c-均值聚类**
  - 基本思想：使用模糊数学中属于$[0,1]$区间的隶属度指标度量单个样本隶属于各个簇的程度
  - 规定每个样本到所有簇的隶属度之和均为1，若某个样本到某个簇的隶属度为1，则表示该样本完全隶属于该簇

划分聚类

- **模糊c–均值聚类**
  - 给定示例样本数据集$D=\{X_1,X_2,\ldots,X_n\}$，假设对数据集D进行模糊聚类得到c个簇$C_1,C_2,\ldots,C_c$，D中任意给定单个样本X_i对于第j个簇C_j的隶属度为α_{ij}，则可使用如下加权欧式距离w_{ij}度量样本X_i与簇C_j之间的相关性：

$$w_{ij}=\alpha_{ij}\left(\sum_{t=1}^{m}\left(x_{it}-u_{jt}\right)^2\right)^{1/2}$$

其中u_{jt}表示第j个簇C_j的聚类中心U_j第t个坐标分量

划分聚类

- **模糊c-均值聚类**
  - 依据上述加权欧式距离w_{ij}计算公式可得所有簇内加权距离之和为：

$$d(\alpha_{ij}) = \sum_{j=1}^{c}\sum_{i=1}^{n}\alpha_{ij}\left(\sum_{t=1}^{m}(x_{it}-u_{jt})^2\right)^{1/2}$$

  - 为控制隶属度对聚类最终效果的影响并简化计算，可将上述加权距离之和$d(\alpha_{ij})$改写为如下形式：

$$J(\alpha_{ij}) = \sum_{j=1}^{c}\sum_{i=1}^{n}\alpha_{ij}^{p}\sum_{t=1}^{m}(x_{it}-u_{jt})^2$$

其中p为控制隶属度影响的参数，通常取$p=2$。 p值越大，则隶属度对最终的聚类效果影响就越大

划分聚类

- **模糊c-均值聚类**
  - 上述关于α_{ij}的函数$J(\alpha_{ij})$既包含所有簇内加权总距离，又包含该聚类算法边界划分的模糊程度，故可将其作为目标函数将样本数据集D的模糊聚类问题转化为$J(\alpha_{ij})$的最小值优化问题，即：

$$\arg_{\alpha_{ij}} \min J(\alpha_{ij});\ \text{s.t.} \sum_{j=1}^{c} \alpha_{ij} = 1$$

  - 可用拉格朗日乘数法求解上述条件优化问题。令拉格朗日函数为：

$$\hat{J}(\alpha_{ij}) = \sum_{j=1}^{c}\sum_{i=1}^{n} \alpha_{ij}^{p} \sum_{t=1}^{m} (x_{it} - u_{jt})^2 + \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{c} \alpha_{ij} - 1 \right)$$

划分聚类

- **模糊c-均值聚类**
  - 分别令$\hat{J}(\alpha_{ij})$对α_{ij}的偏导数令为0，则有：

$$\frac{\partial \hat{J}}{\partial \alpha_{ij}} = \sum_{j=1}^{c}\sum_{i=1}^{n} p\alpha_{ij}^{p-1}\sum_{t=1}^{m}(x_{it}-u_{jt})^2 + \sum_{i=1}^{n} c\lambda_i = 0$$

  - 解得：$\alpha_{ij} = [\frac{-\lambda_i}{p\sum_{t=1}^{m}(x_{it}-u_{jt})^2}]^{\frac{1}{p-1}}$
  - 结合隶属度约束条件可消去未知参数λ_i，得到隶属度α_{ij}的计算公式：

$$\alpha_{ij} = \left(\sum_{k=1}^{c}\frac{\sum_{t=1}^{m}(x_{it}-u_{jt})^2}{\sum_{t=1}^{m}(x_{it}-u_{kt})^2}\right)^{\frac{1}{1-p}}$$

  - 上式表明第i个样本到第j个簇的最佳隶属度α_{ij}取决于该样本点到第j个簇心的距离与其到所有簇心距离的比值之和

划分聚类

- **模糊c–均值聚类**
  - 还可以将目标函数$\hat{J}(\alpha_{ij})$看成是聚类中心u_{jt}的函数，即$\hat{J}(u_{jt})$，并由此通过对目标函数$\hat{J}(u_{jt})$作最小值优化计算进一步得到各簇最优聚类中心坐标U_j。为此，分别令$\hat{J}(u_{jt})$关于u_{jt}的偏导数为0，可得到如下方程：

$$-2\sum_{j=1}^{c}\sum_{i=1}^{n}\alpha_{ij}^{p}\sum_{t=1}^{m}(x_{it}-u_{jt})=0$$

  - 即有：$\sum_{i=1}^{n}\alpha_{ij}^{p}\,x_{it}=\sum_{i=1}^{n}\alpha_{ij}^{p}\,u_{jt},\ \ t=1,2,\ldots,m$
  - 由此可得如下聚类中心计算公式：

$$U_j=\sum_{i=1}^{n}\alpha_{ij}^{p}\,X_i\Big/\sum_{i=1}^{n}\alpha_{ij}^{p}$$

划分聚类

- **模糊c-均值聚类**
  - 模糊c-均值聚类算法依据上述隶属度和聚类中心计算公式，算法具体步骤如下：
  - 设定簇的数目c和阈值ε，并令$s = 0$。随机初始化所有样本对所有簇的隶属度，并将其记录在隶属度矩阵Q中，即：

$$Q^0 = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1c} \\ \cdots & \cdots & \cdots \\ \alpha_{n1} & \cdots & \alpha_{nc} \end{pmatrix}$$

其中元素α_{ij}为非负实数且满足隶属度约束条件$\sum_{j=1}^{c} \alpha_{ij} = 1$

划分聚类

- **模糊c–均值聚类**
  - 使用隶属度矩阵Q计算各簇的聚类中心$u_j^q, j = 1,2,\ldots,c$，计算目标函数值J^t
  - 若$J^s \geq \varepsilon$或$|J^s - J^{s-1}| \geq \varepsilon$，则更新隶属度矩阵$Q$，令$s = s + 1$并返回步骤2；否则，依据隶属度矩阵$Q^s$得到聚类结果并结束算法

划分聚类

例题：现假设在二维平面中有6个点，如表所示，试使用模糊c-均值聚类算法对数据集进行模糊二均值聚类，当每个聚类中心相邻两次迭代的变化均小于10^{-4}时停止聚类过程并算出相应的聚类中心和隶属度矩阵结果

	X_1	X_2	X_3	X_4	X_5	X_6
x_{i1}	3	4	9	14	18	21
x_{i2}	3	10	6	8	11	7

划分聚类

	X_1	X_2	X_3	X_4	X_5	X_6
x_{i1}	3	4	9	14	18	21
x_{i2}	3	10	6	8	11	7

- 将表中数据点二维空间分布情况如图所示。取$p=2$，并令：

$$d_{ij}=\left(\sum_{t=1}^{m}\left(x_{it}-u_{jt}\right)^2\right)^{\frac{1}{2}},\quad d_{ik}=\left(\sum_{t=1}^{m}\left(x_{it}-u_{kt}\right)^2\right)^{\frac{1}{2}}$$

则可将隶属度计算公式表示为：$\alpha_{ij}=\left(\sum_{k=1}^{2}\frac{d_{ij}^2}{d_{ik}^2}\right)^{-1}$

划分聚类

	X_1	X_2	X_3	X_4	X_5	X_6
x_{i1}	3	4	9	14	18	21
x_{i2}	3	10	6	8	11	7

- 随机选择的初始聚类中心分别为$c_1 = a, c_2 = b$，使用该聚类中心确定第一次划分：
- 对于任意点X_i，使用α_{ij}计算公式计算隶属度：对于点X_1，由于其为第一个簇C_1的聚类中心，故有$\alpha_{11} = 1$，$\alpha_{12} = 0$，同理，对于点X_2，有$\alpha_{21} = 0$，$\alpha_{22} = 1$。对于点X_3，将数据带入隶属度公式计算，得到$\alpha_{31} = 0.48, \alpha_{32} = 0.52$。同理计算得其他数据点所对应的隶属度，根据隶属度数据将聚类中心更新为：

$$c_j' = \left(\frac{\sum_{\mathrm{i}} \alpha_{ij}^2 x_{i1}}{\sum_{\mathrm{i}} \alpha_{ij}^2}, \frac{\sum_{i} \alpha_{ij}^2 x_{i2}}{\sum_{\mathrm{i}} \alpha_{ij}^2}\right)$$

划分聚类

	X_1	X_2	X_3	X_4	X_5	X_6
x_{i1}	3	4	9	14	18	21
x_{i2}	3	10	6	8	11	7

- 带入数据可得第一次迭代时的聚类中心为：

$$c_1' = (8.47,5.12)，\ c_2' = (10.42,8.99)$$

隶属度矩阵为：

$$M_1^T = \begin{bmatrix} 1 & 0 & 0.48 & 0.42 & 0.41 & 0.47 \\ 0 & 1 & 0.52 & 0.58 & 0.59 & 0.53 \end{bmatrix}$$

迭代执行上述步骤可得第二次迭代时，聚类中心为：

$$c_1' = (8.51,6.11)，\ c_2' = (14.42,8.69)$$

隶属度矩阵为：

$$M_2^T = \begin{bmatrix} 0.73 & 0.49 & 0.91 & 0.26 & 0.33 & 0.42 \\ 0.27 & 0.51 & 0.09 & 0.74 & 0.67 & 0.58 \end{bmatrix}$$

划分聚类

	X_1	X_2	X_3	X_4	X_5	X_6
x_{i1}	3	4	9	14	18	21
x_{i2}	3	10	6	8	11	7

- 第三次迭代时的聚类中心为：

$$c_1' = (6.40,6.24),\quad c_2' = (16.55,8.64)$$

隶属度矩阵为：

$$M_3^T = \begin{bmatrix} 0.80 & 0.76 & 0.99 & 0.02 & 0.14 & 0.23 \\ 0.20 & 0.24 & 0.01 & 0.98 & 0.86 & 0.77 \end{bmatrix}$$

重复上述过程，经过8次迭代得到如下聚类中心为和隶属度矩阵：

$$c_1' = (5.24,6.34),\quad c_2' = (17.84,8.73)$$

$$M_8^T = \begin{bmatrix} 0.94 & 0.93 & 0.86 & 0.16 & 0.03 & 0.05 \\ 0.06 & 0.07 & 0.14 & 0.84 & 0.97 & 0.95 \end{bmatrix}$$

此时每个聚类中心相邻两次迭代的变化均小于10^{-4}

本节目录

- 划分聚类
- **密度聚类**

密度聚类

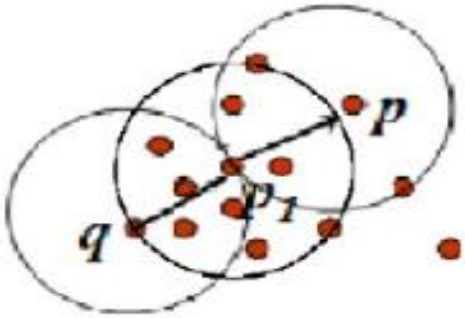
- **概述**
  - 基于划分的聚类算法主要通过样本数据之间的距离进行聚类操作，主要适合于对类圆形聚簇的聚类，如果将其用于对具有任意形状的聚簇进行聚类则有时不能获得满意的效果
  - **密度聚类算法**：将聚簇看作是数据空间中被稀疏区域分开的稠密区域，由此得到以密度为度量标准的样本数据聚类方法
  - 三种代表性密度聚类算法：DBSCAN算法、OPTICS算法和DENCLUE算法

密度聚类

- **DBSCAN密度聚类**
  - 两个参数ε和$MinPts$
    - ε：领域半径
    - $MinPts$：在 以 ε 为半径的领域内最少包含点的个数（密度阈值）
  - 核心对象：一个对象的ε-邻域至少包含$MinPts$个对象
  - 边界对象：不是核心点 ，但落在某个核心点的ε邻域内的对象
  - 噪声对象：不属于任何簇的对象

密度聚类

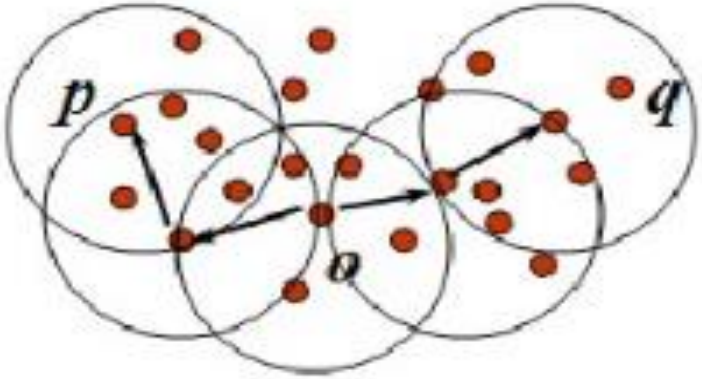
- **DBSCAN密度聚类**
  - 密度直达(Directly density reachable, DDR):如果q是一个核心对象，p1属于q的邻域，那么称q密度直达p1



  - 密度可达的(density reachable):点 p 关于ε和$MinPts$是从 q密度可达的, 如果存在一个节点链 p1, …, pn, p1 = q, pn = p，pi密度直达pi+1，则称p密度可达q

密度聚类

- **DBSCAN密度聚类**
  - 密度相连的：点p关于 ε和$MinPts$与点 q是密度相连的，如果 存在点o使得，p和q都是关于ε和$MinPts$是从o密度可达的(如果存在o，o密度可达q和p，则称p和q是密度相连的)



密度聚类

- **DBSCAN密度聚类**
  - 任意选取一个点p
  - 得到所有从p关于ε和$MinPts$密度可达的点
  - 如果p是一个核心点，则找到一个聚类
  - 如果p是一个边界点，没有从p密度可达的点，DBSCAN将访问数据库中的下一个点
  - 继续这一过程，直到数据库中的所有点都被处理

密度聚类

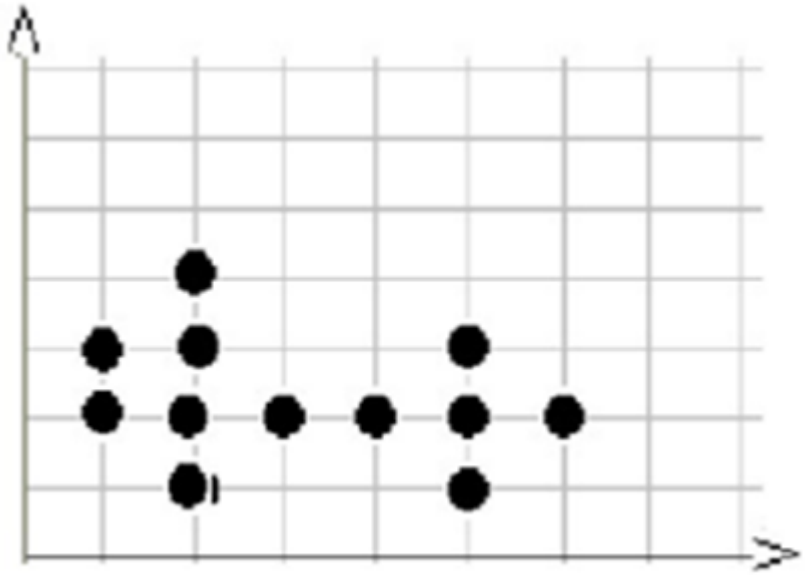
- **例题：**已知表所示的某数据集D，试用DBSCAN算法对其进行密度聚类分析，取$\varepsilon = 1$、$MinPts = 4$、$n = 12$

序号	1	2	3	4	5	6	7	8	9	10	11	12
属性A	2	5	1	2	3	4	5	6	1	2	5	2
属性B	1	1	2	2	2	2	2	2	3	3	3	4

密度聚类

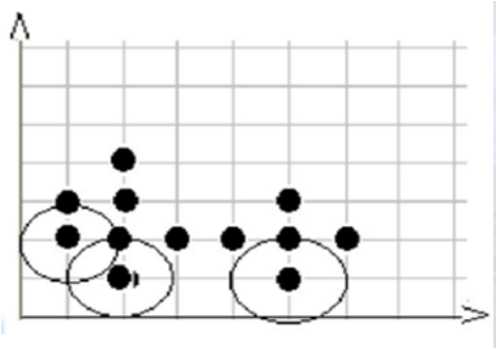
序号	1	2	3	4	5	6	7	8	9	10	11	12
属性A	2	5	1	2	3	4	5	6	1	2	5	2
属性B	1	1	2	2	2	2	2	2	3	3	3	4

- 首先可视化数据集D，将其绘制在如图所示平面直角坐标系中。然后，使用DBSCAN算法对其进行密度聚类

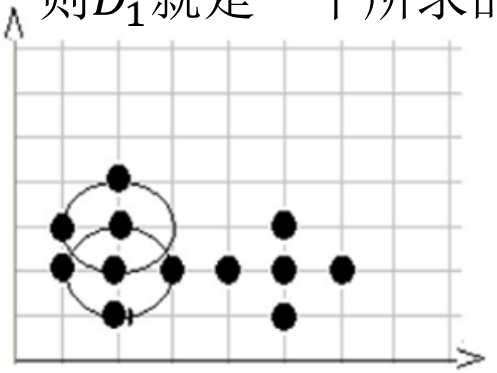


密度聚类

- 第一步，在数据集D中任意选择一个样本点首先选择1号样本点，由于以1号样本点为圆心且半径为1的圆中只包含2个样本点，小于4个，故1号样本点不是核心对象点。同理可得第2号、第3号样本点均不是核心样本点，如图所示

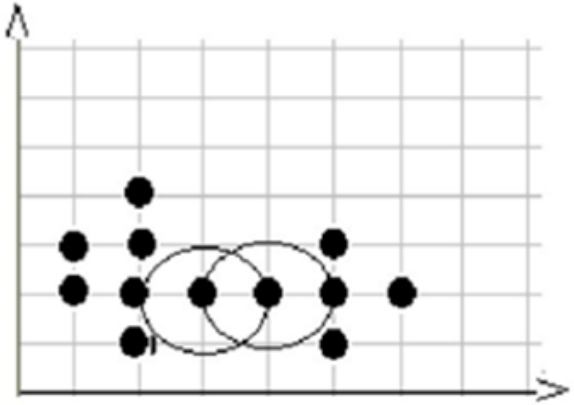


- 第二步，易知4号样本点是一个核心对象点。从4号样本点出发寻找所有与其具有可达关系的其余样本点，可以找到4个直接可达样本点、3个间接可达样本点，将这7个样本点组成一个样本子集合$D_1 = \{1, 3, 4, 5, 9, 10, 12\}$，则$D_1$就是一个所求的聚簇，如图所示

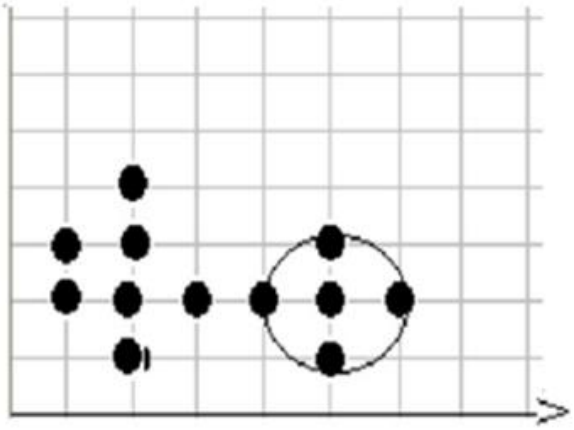


密度聚类

- 第三步，选择第5号样本点，因为第5号样本点已在簇D_1内，故选择下个样本点，选择第6号样本点，易知第6号样本点不是核心对象点，如图所示

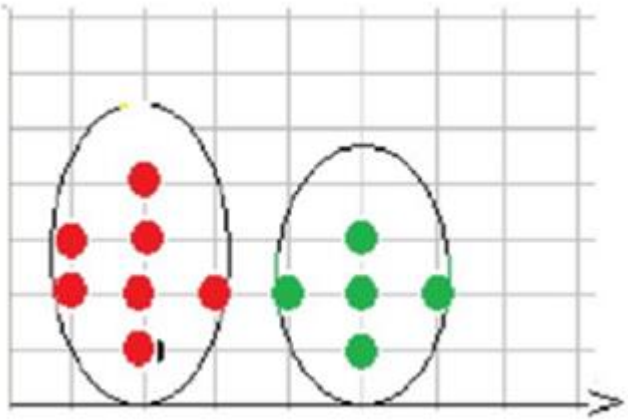


- 第四步，选择第7号样本点，易知第7号样本点是核心对象点。与第二步同理获得一个新的聚簇$D_2 = \{2,6,7,8,11\}$，如图所示



密度聚类

- 第五步，在数据集D选择第8号样本点，此样本点已经在簇2里面，故选择下一个样本点；同理发现样本点9、10和12已在聚簇D_1内，样本点11已在聚簇D_2内。此时已完成对数据集D中所有样本点的聚类分析，结束聚类过程并输出聚簇D_1和D_2，图显示了最终聚类结果



密度聚类

- **DBSCAN密度聚类**
  - 两个初始参数ε（邻域半径）和$MinPts$（ε邻域最小点数）需要用户手动设置输入，聚类的类簇结果对这两个参数的取值非常敏感
  - 在样本点密度分布不够均匀的场合，使用DBSCAN算法则难以获得满意的效果
  - 为避免DBSCAN算法在使用全局固定参数方面的局限，可以使用OPTICS密度聚类算法

密度聚类

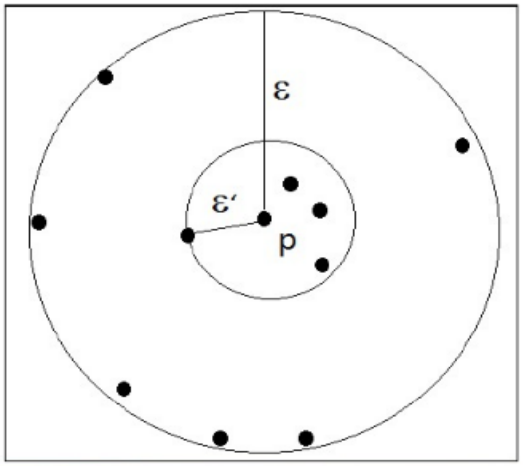
- **OPTICS密度聚类**
  - OPTICS算法并不显示的产生结果类簇，而是为聚类分析生成一个增广的簇排序，这个排序代表了各样本点基于密度的聚类结构
  - 从这个排序中可以得到基于参数ε和$MinPts$在任意取值下的聚类结果
  - 在同时构建不同的聚类时,以特定的顺序来处理对象，优先选择最小的ε值密度可达的对象，以便高密度的聚类能被首先完成

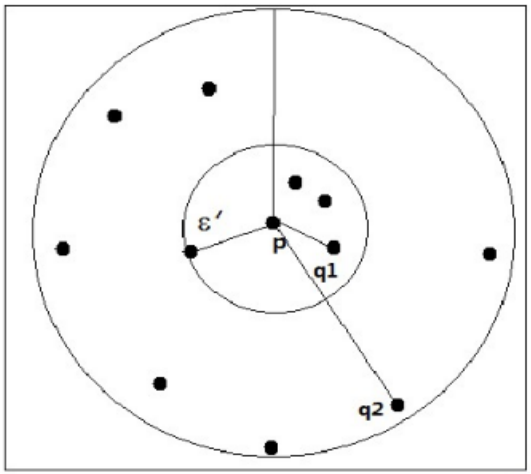
密度聚类

- **OPTICS密度聚类**
  - 每个对象需要存储两个值
    - 对象p的核心距离(core-distance)是使得p成为核心对象的最小ε。如果p不是核心对象，p的核心距离没有定义
    - 对象q关于另一个对象p的可达距离(reachability-distance)是p的核心距离和p与q的欧几里得距离之间的较大值。如果p不是一个核心对象，p和q之间的可达距离没有定义

密度聚类

- **OPTICS密度聚类**
  - 例：设ε=6(mm)，MinPts=5
  - p的核心距离是p与四个最近的数据对象之间的距离ε’
  - q1关于p的可达距离是p的核心距离(即ε’=3mm)，因为它比从p到q1的欧几里得距离要大
  - q2关于p的可达距离是从p到q2的欧几里得距离，它大于p的核心距离





密度聚类

- **OPTICS密度聚类**
  - 计算样本数据集D中每个样本的对象的核心距离和相应的可达距离，并根据样本对象到与其最近核心对象之间的可达距离实现对D中所有样本对象的排序，生成一个有序的线性表
  - OPTICS算法的具体计算过程如下：
  - （1）创建两个队列，有序队列和结果队列。有序队列用来存储核心对象及其直接可达对象，并按可达距离升序进行排列，结果队列用来存储样本点的输出次序
  - （2）如果样本集D中所有样本点都处理完毕，则算法结束。否则，任意选择一个未处理样本点，找到其所有直接密度可达样本点，若该样本点不在于结果队列中，则将其放入有序队列并按可达距离排序

密度聚类

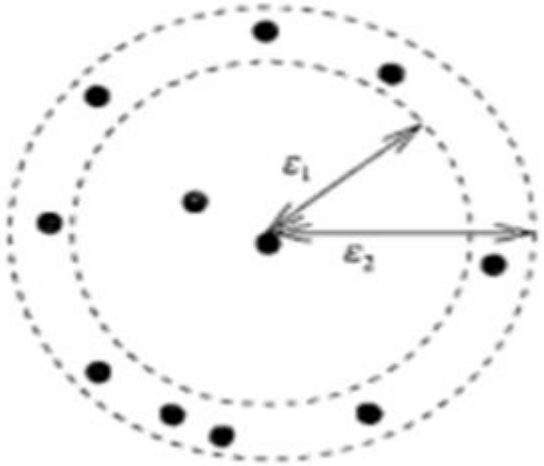
- OPTICS密度聚类
  - （3）如果有序队列为空，则跳至步骤2，重新选取处理数据；否则，从有序队列中取出第一个样本点，即可达距离最小的样本点进行拓展，并将取出的样本点保存至结果队列中
  - （4）判断该拓展点是否是核心对象，若该点是核心对象，则找到该拓展点所有的直接密度可达点；若不是，则跳至步骤3，取可达距离倒数第二小的样本点。再判断该直接密度可达样本点是否已经存在结果队列，是则不处理，否则下一步

密度聚类

- **OPTICS密度聚类**
  - （5）将所有的直接密度可达点放入有序队列，且将有序队列中的点按照可达距离重新排序，如果该点已经在有序队列中且新的可达距离较小，则更新该点的可达距离；如果有序队列中不存在该直接密度可达样本点，则插入该点并对有序队列进行重新排序
  - （6）迭代上述步骤2至5，直至有序队列为空，此时输出结果队列中样本点序列

密度聚类

- **OPTICS密度聚类**
  - DBSCAN算法和OPTICS算法的缺点中，样本数据对象的分布密度都是通过统计被半径参数ε所界定邻域中的样本对象个数进行计算
  - 这种密度估计方法有时对半径值的变化非常敏感

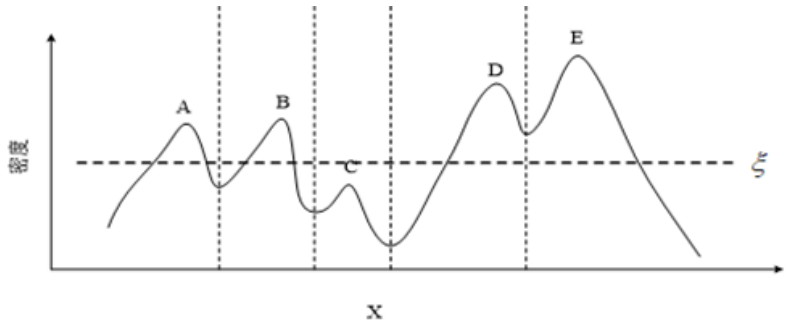


密度聚类

- **DENCLUE密度聚类**
  - DENCLUE 算法通过样本点的分布估计密度函数，用与每个点相关联的影响函数之和对数据集进行总密度建模，最终得到一个在属性空间中的用来描述数据集总密度的密度函数
  - 总密度函数会有局部尖峰，即局部密度极大值和局部低谷，即局部密度极小值，每一个尖峰对应了一个簇质心，而簇与簇之间通过低谷来分离
  - 尖峰也被称为局部吸引点或密度吸引点

密度聚类

- **DENCLUE密度聚类**
  - 例如，图4-17表示某一维数据集基于DENCLUE算法的聚类结果，A~E是该数据集总密度函数的尖峰，ξ是图中所示的密度阈值。图中作为尖峰的A、B、D和E点也是局部吸引点或密度吸引点，它们的局部密度影响区域会形成聚簇，故A和B各自形成一个单独的簇；D和E是两个邻近的尖峰，同时处于密度阈值ξ之上且它们之间每个样本点处的密度均高于密度阈值，故可将它们所代表的影响区域合并为一个簇
  - 尖峰C所代表的影响区域由于处在ξ之下，故其所代表的簇被定义为噪声



密度聚类

- **DENCLUE密度聚类**
  - DENCLUE算法通过引入核密度估计方法而不是通过统计由半径参数ε所定义邻域中样本对象个数来计算密度
  - 核密度估计的目标是用函数描述样本数据点的分布，并使用某种特定的核函数表示每个样本点对总密度函数的贡献，把总密度函数看成是与每个样本点相关联的核函数之和

密度聚类

- **DENCLUE密度聚类**
  - DENCLUE算法的原理：设$X_1,\cdots,X_n$是随机变量f的独立同分布样本，对于样本空间中任意给定的某个样本点X，在点X处的样本分布密度通常与所有样本点$X_1,\cdots,X_n$相关，可根据核密度的基本思想构造如下关于概率密度函数$f(X)$的估计量$\hat{f}_h(X)$，即有：

$$\hat{f}_h(X)=\frac{1}{nh}\sum_{i=1}^{n}K(\frac{X-X_i}{h})$$

其中$K(t)$为函数核，h是用作光滑参数的带宽。

DENCLUE算法通常取均值为0，方差为1的标准高斯函数作为核函数，即有：

$$K(t)=\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}$$

密度聚类

- **DENCLUE密度聚类**
  - 样本数据集D中任意给定的样本点X，如果X是密度函数估计量$\hat{f}_h(X)$的一个局部极大点，则称该样本点X为一个密度吸引点，记为X^*
  - DENCLUE算法使用一个噪声阈值ξ约束极大点的函数取值，即仅考虑满足$\hat{f}(X^*) \geq \xi$的密度吸引点X^*，并将这些满足约束条件的非平凡密度吸引点作为聚簇中心进行聚类
  - 对于样本数据集D中任意一个样本点，DENCLUE算法从该样本点出发通过一个步进式爬山过程寻找与该样本点对应的密度吸引点
  - 并将该样本点分配到所找到密度吸引点生成的聚簇中，完成对该样本点的聚类

密度聚类

- **DENCLUE密度聚类**
  - 具体地说，对于数据集D中任意一个样本点X，爬山过程从X出发并被函数$\hat{f}_h(X)$的梯度所指导通过迭代计算寻找密度吸引点X^*，即有：

$$X^{j+1} = X^j + \delta \frac{\nabla \hat{f}(X^j)}{\left|\nabla \hat{f}(X^j)\right|}$$

其中$X^0 = X$，δ为迭代计算的松弛因子，主要用于控制收敛速度，且有：

$$\nabla \hat{f}(X) = \frac{1}{h^{d+2} n \sum_{i=1}^{n} K\left(\frac{X - X_i}{h}\right)(X_i - X)}$$

密度聚类

- **DENCLUE密度聚类**
  - 如果在爬山过程中出现 $\hat{f}(x^{k+1}) < \hat{f}(x^k)$ ，则停止迭代，并将样本点X^k作为样本点X所对应的密度吸引点X^*，即令$X^* = X^k$，将X分配给X^k所生成的聚簇，完成对X的聚类；否则，如果爬山过程收敛于某个满足$\hat{f}(X^*) < \varepsilon$的平凡局部最大点$X^*$，则认为样本点$X$是一个噪声点，无需对其进行聚类

密度聚类

- **DENCLUE密度聚类**
  - DENCLUE算法具体步骤如下：
  - 对样本点占据的邻域空间构造密度函数
  - 通过沿密度变化最大方向，即梯度方向移动，识别密度函数的最大局部点即局部吸引点，将每个点关联到一个密度吸引点
  - 定义与特定的密度吸引点相关联的样本点构成的簇
  - 丢弃密度吸引点的密度小于用户指定阈值ξ的簇
  - 若两个密度吸引点之间存在密度大于或者等于 ε 的路径，则合并由它们路径连接所形成的聚簇。对所有密度吸引点重复此过程，直到不会产生新的合并时算法终止

密度聚类

- **DENCLUE密度聚类**
  - OPTICS算法并不显示的产生结果类簇，而是为聚类分析生成一个增广的簇排序，这个排序代表了各样本点基于密度的聚类结构
  - 从这个排序中可以得到基于参数ε和$MinPts$在任意取值下的聚类结果
  - 在同时构建不同的聚类时,以特定的顺序来处理对象，优先选择最小的ε值密度可达的对象，以便高密度的聚类能被首先完成

总结

- **划分聚类**
  - K均值聚类
  - 模糊k-均值聚类
- **密度聚类**
  - DBSCAN密度聚类
  - OPTICS密度聚类
  - DENCLUE密度聚类

思考题

- 对于k-均值聚类算法，如何使得该算法对于离群点更具有鲁棒性
- 试分析划分聚类算法和密度聚类算法所能得到的簇边界有何区别

练习题

- 试编程实现k均值聚类算法，设置三组不同的k值、三组不同的初始中心点，在西瓜数据集4.0上进行实验比较，并讨论什么样的初始中心有利于取得好结果