

PART5 聚类

基本内容:

1. 什么是聚类? 什么是分类? 二者的区别与联系。
2. **样本点之间**欧式距离、马氏距离、切比雪夫距离、曼哈顿距离?
样本点与簇、簇与簇之间的距离计算;
最小距离、最远距离、平均距离法度量簇间距离。
3. 掌握 K-Means Clustering 算法, 影响该算法性能的因素有哪些?
4. 以聚合式系统聚类为例, 掌握系统聚类。
5. 理解基于高斯混合模型进行聚类的基本流程, 了解什么是高斯混合模型?
6. 以 DBSCAN 算法为例, 理解密度聚类实现的基本流程, 掌握有关概念, 哪些因素会影响聚类的效果。
7. 理解聚类算法有关性能的评价指标。

练习:

1. **聚类.** 给定数据集 $D = \{x_i, i = 1, \dots, m\}$, 其中 $x_i \in R^d$. 若采用 K-均值聚类算法将该数据集 D 划分为 K 簇 $\{C_1, \dots, C_K\}$, 请完成如下工作:
 - (1) 写出 K-均值聚类算法的准则函数及其意义, 并给出必要参数说明;
 - (2) 对 K-均值聚类算法的实现过程进行描述.
 - (3) 哪些因素会影响 K-均值聚类的性能?
 - (4) 如何弱化这些因素的不良影响?

答:

- (1) 准则函数---总的簇内误差平方和最小

$$E(\mu_1, \dots, \mu_k, C_1, \dots, C_k) = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

C_i -- 第 i 簇, $i = 1, \dots, k$

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

(2)

输入: 样本集 $D = \{x_1, x_2, \dots, x_m\}$;
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 $\{\mu_1, \mu_2, \dots, \mu_k\}$
- 2: repeat
- 3: 令 $C_i = \emptyset$ ($1 \leq i \leq k$)
- 4: for $j = 1, 2, \dots, m$ do
- 5: 计算样本 x_j 与各均值向量 μ_i ($1 \leq i \leq k$) 的距离: $d_{ji} = \|x_j - \mu_i\|_2$;
- 6: 根据距离最近的均值向量确定 x_j 的簇标记: $\lambda_j = \arg \min_{i \in \{1, 2, \dots, k\}} d_{ji}$;
- 7: 将样本 x_j 划入相应的簇: $C_{\lambda_j} = C_{\lambda_j} \cup \{x_j\}$;
- 8: end for
- 9: for $i = 1, 2, \dots, k$ do
- 10: 计算新均值向量: $\mu'_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$;
- 11: if $\mu'_i \neq \mu_i$ then
- 12: 将当前均值向量 μ_i 更新为 μ'_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 $C = \{C_1, C_2, \dots, C_k\}$

(2) 样本集是否规范化预处理、距离度量方式的不同、K 值的大小、以及初始化方式均会影响 K 均值聚类性能。

2. 若要采用合并式层次聚类(也称: 聚合式系统聚类)将样本集 $D = \{x_i, i = 1, \dots, m\}$

划分为K簇, 其中 $x_i \in R^d$. 请完成如下工作:

(1)请对该聚类算法的实现流程予以描述.

(2)上述聚类过程中, 需要进行不同聚类簇之间的距离计算, 请分别采用最近距离、最远距离, 估算任意两簇 C_j, C_l 之间的距离.

解:

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.逆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 $D = \{x_1, x_2, \dots, x_m\}$;
聚类簇距离度量函数 d ;
聚类簇数 k .

过程：

```

1: for  $j = 1, 2, \dots, m$  do
2:    $C_j = \{x_j\}$ 
3: end for
4: for  $i = 1, 2, \dots, m$  do
5:   for  $j = 1, 2, \dots, m$  do
6:      $M(i, j) = d(C_i, C_j)$ ;
7:      $M(j, i) = M(i, j)$ 
8:   end for
9: end for
10: 设置当前聚类簇个数:  $q = m$ 
11: while  $q > k$  do
12:   找出距离最近的两个聚类簇  $C_{i^*}$  和  $C_{j^*}$ ;
13:   合并  $C_{i^*}$  和  $C_{j^*}$ :  $C_{i^*} = C_{i^*} \cup C_{j^*}$ ;
14:   for  $j = j^* + 1, j^* + 2, \dots, q$  do
15:     将聚类簇  $C_j$  重编号为  $C_{j-1}$ 
16:   end for
17:   删除距离矩阵  $M$  的第  $j^*$  行与第  $j^*$  列;
18:   for  $j = 1, 2, \dots, q - 1$  do
19:      $M(i^*, j) = d(C_{i^*}, C_j)$ ;
20:      $M(j, i^*) = M(i^*, j)$ 
21:   end for
22:    $q = q - 1$ 
23: end while

```

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

(1)

输入：样本集 $D = \{x_1, x_2, \dots, x_m\}$;
聚类簇距离度量函数 d ;
聚类簇数 k .

过程：

```

1: for  $j = 1, 2, \dots, m$  do
2:    $C_j = \{x_j\}$ 
3: end for
4: for  $i = 1, 2, \dots, m$  do
5:   for  $j = 1, 2, \dots, m$  do
6:      $M(i, j) = d(C_i, C_j)$ ;
7:      $M(j, i) = M(i, j)$ 
8:   end for
9: end for
10: 设置当前聚类簇个数:  $q = m$ 
11: while  $q > k$  do
12:   找出距离最近的两个聚类簇  $C_{i^*}$  和  $C_{j^*}$ ;
13:   合并  $C_{i^*}$  和  $C_{j^*}$ :  $C_{i^*} = C_{i^*} \cup C_{j^*}$ ;
14:   for  $j = j^* + 1, j^* + 2, \dots, q$  do
15:     将聚类簇  $C_j$  重编号为  $C_{j-1}$ 
16:   end for
17:   删除距离矩阵  $M$  的第  $j^*$  行与第  $j^*$  列;
18:   for  $j = 1, 2, \dots, q - 1$  do
19:      $M(i^*, j) = d(C_{i^*}, C_j)$ ;
20:      $M(j, i^*) = M(i^*, j)$ 
21:   end for
22:    $q = q - 1$ 
23: end while

```

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

(2)

最近距离 $d_{\min}(C_j, C_l) = \min_{\substack{x \in C_j \\ z \in C_l}} \text{dist}(x, z)$

最远距离 $d_{\max}(C_j, C_l) = \max_{\substack{x \in C_j \\ z \in C_l}} \text{dist}(x, z)$

3. DBSCAN是一种基于密度的聚类算法，若要基于该算法,对观测点集 $D = \{x_i, i = 1, \dots, N\}$ 进行聚类，需要提供两个全局参数 $(\varepsilon, \text{MinPts})$ ，回答如下问题：

(1)两个参数的意义是什么？如何确定核心对象？边界对象？噪声对象？什么是密度直达？什么是密度可达？密度相连？

(2)理解DBSCAN算法的实现流程。

解：

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

ε --邻域最大半径

MinPts--给定样本的 **ε -邻域**内最小样本数.

其中：

ε -邻域 对于 $\forall x_j \in D$, x_j 的 ε -邻域为 $N_\varepsilon(x_j) = \{x_i \in D \mid \text{dist}(x_i, x_j) \leq \varepsilon\}$

(2)**核心对象** (*core object*)

若 $x_j \in D$ 并且 $|N_\varepsilon(x_j)| \geq \text{MinPts}$, 则称 x_j 为一个**核心对象**.

(3)**密度直达** (*directly density-reacheable*)

若 $x_j \in N_\varepsilon(x_i)$, 并且 x_i 为一个核心对象, 则称 x_j 为由 x_i **密度直达**.

(4)**密度可达** (*density-reacheable*)

对于 x_i, x_j , 若存在样本序列 p_1, p_2, \dots, 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**.

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 $D = \{x_1, x_2, \dots, x_m\}$;
邻域参数 $(\epsilon, MinPts)$.

过程:

```
1: 初始化核心对象集合:  $\Omega = \emptyset$ 
2: for  $j = 1, 2, \dots, m$  do
3:   确定样本  $x_j$  的  $\epsilon$ -邻域  $N_\epsilon(x_j)$ ;
4:   if  $|N_\epsilon(x_j)| \geq MinPts$  then
5:     将样本  $x_j$  加入核心对象集合:  $\Omega = \Omega \cup \{x_j\}$ 
6:   end if
7: end for
```

8: 初始化聚类簇数: $k = 0$

9: 初始化未访问样本集合: $\Gamma = D$

```
10: while  $\Omega \neq \emptyset$  do
11:   记录当前未访问样本集合:  $\Gamma_{old} = \Gamma$ ;
12:   随机选取一个核心对象  $o \in \Omega$ , 初始化队列  $Q = \{o\}$ ;
13:    $\Gamma = \Gamma \setminus \{o\}$ ; 更新未访问的样本集合
14:   while  $Q \neq \emptyset$  do
15:     取出队列  $Q$  中的首个样本  $q$ ;
16:     if  $|N_\epsilon(q)| \geq MinPts$  then
17:       令  $\Delta = N_\epsilon(q) \cap \Gamma$ ;
18:       将  $\Delta$  中的样本加入队列  $Q$ ;
19:        $\Gamma = \Gamma \setminus \Delta$ ; 更新未访问的样本集合
20:     end if
21:   end while
22:    $k = k + 1$ , 生成聚类簇  $C_k = \Gamma_{old} \setminus \Gamma$ ;
23:    $\Omega = \Omega \setminus C_k$  更新核心对象集合
24: end while
```

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

4. 给定数据集 $D = \{x_i, i = 1, \dots, m\}$, 其中 $x_i \in R^d$, 若采用高斯混合模型将数据集 D 划分为 K 簇 $\{C_1, \dots, C_K\}$. 按要求完成如下工作:

(1) 写出高斯混合概率密度函数的具体表达形式, 并指出该模型各参数的意义.

(2) 若采用EM算法基于该数据集 D 已经完成了高斯混合模型的参数估计, 如何由高斯混合模型得到关于数据集 D 的最终划分结果?

解:

(1)

$$p_M(\mathbf{x}) = \sum_{j=1}^K \alpha_j p(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \sum_{j=1}^K \frac{\alpha_j}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_j|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) \right]$$

其中

$$\left\{ \begin{array}{l} K \text{--高斯成分的数目} \\ \alpha_j \text{--第} j \text{个高斯成分的混合系数} \quad \sum_{j=1}^K \alpha_j = 1 \\ p(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \text{--第} j \text{个高斯成分的概率密度函数} \\ p(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_j|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) \right] \\ \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j \text{--第} j \text{个高斯成分的期望向量、协方差矩阵} \\ \text{待估计的参数集合: } \{(\alpha_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) | j = 1, \dots, K\} \end{array} \right.$$

(2) 设参数估计结果 $\{(\hat{\alpha}_j, \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j) | j = 1, \dots, K\}$, 则

对于数据集 \mathbf{D} 的观测样本 $\mathbf{x}_i, i = 1, \dots, m$

其相应聚类簇的划分结果为:

$$\lambda_i = \arg \max_{j \in \{1, 2, \dots, K\}} \hat{\alpha}_j p(\mathbf{x}_i | \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j) \quad i = 1, \dots, m$$

(3) 其它: 理解模型学习中的交叉迭代过程:

GMM模型参数估计的交叉迭代

混合高斯模型 $p_M(\mathbf{x}) = \sum_{i=1}^k \alpha_i p(\mathbf{x} | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

(1) 初始化 $\hat{\boldsymbol{\mu}}_i(0), \hat{\boldsymbol{\Sigma}}_i(0), \hat{\alpha}_i(0), i = 1, \dots, k$

(2) 交叉迭代直到满足终止条件:

E-STEP $\gamma_{ji}(n+1) = \frac{\alpha_i(n) p(\mathbf{x}_j | \boldsymbol{\mu}_i(n), \boldsymbol{\Sigma}_i(n))}{\sum_{i=1}^k \alpha_i(n) p(\mathbf{x}_j | \boldsymbol{\mu}_i(n), \boldsymbol{\Sigma}_i(n))} \quad \begin{cases} i = 1, \dots, k \\ j = 1, \dots, m \end{cases}$

$\gamma_{ji} = P_M(z_j = i | \mathbf{x}_j)$

M-STEP $\left\{ \begin{array}{l} \hat{\boldsymbol{\mu}}_i(n+1) = \frac{\sum_{j=1}^m \gamma_{ji}(n+1) \mathbf{x}_j}{\sum_{j=1}^m \gamma_{ji}(n+1)} \quad i = 1, \dots, k \\ \hat{\boldsymbol{\Sigma}}_i(n+1) = \frac{\sum_{j=1}^m \gamma_{ji}(n+1) (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_i(n+1)) (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_i(n+1))^T}{\sum_{j=1}^m \gamma_{ji}(n+1)} \quad i = 1, \dots, k \\ \hat{\alpha}_i(n+1) = \frac{1}{m} \sum_{j=1}^m \gamma_{ji}(n+1) \quad i = 1, \dots, k \end{array} \right.$

$\gamma_{ji} = P_M(z_j = i | \mathbf{x}_j)$