

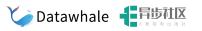


《机器学习公式详解》 (南瓜书)

第9章 聚类

本节主讲: 秦州

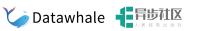
本节大纲



南瓜书对应章节: 9.3, 9.4, 9.5, 9.6

- 1. 距离计算
- 2. k-means (原型聚类)
- 3. DBSCAN (密度聚类)
- 4. AGNES (层次聚类)

聚类和距离计算

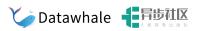


聚类:**物以类聚**。将相似的样本聚集到一起,使得**同一类簇的样本尽可能接近,不同类簇的样本尽可能远离**。

对"距离"的定义:

- 1. 非负性: $\operatorname{dist}\left(\boldsymbol{x}_{i},\boldsymbol{x}_{j}\right)\geqslant0$
- 2. 同一性: $\operatorname{dist}\left(oldsymbol{x}_{i},oldsymbol{x}_{j}
 ight)=0$ 当且仅当 $oldsymbol{x}_{i}=oldsymbol{x}_{j}$
- 3. 对称性: $\operatorname{dist}\left(\boldsymbol{x}_{i},\boldsymbol{x}_{j}\right)=\operatorname{dist}\left(\boldsymbol{x}_{j},\boldsymbol{x}_{i}\right)$
- 4. 直递性: $\operatorname{dist}\left(oldsymbol{x}_i,oldsymbol{x}_j
 ight)\leqslant\operatorname{dist}\left(oldsymbol{x}_i,oldsymbol{x}_k
 ight)+\operatorname{dist}\left(oldsymbol{x}_k,oldsymbol{x}_j
 ight)$

常用的距离度量 - 连续/离散有序



明可夫斯基距离(Minkowski distance)

$$\operatorname{dist}_{\operatorname{mk}}\left(oldsymbol{x}_i,oldsymbol{x}_j
ight) = \left(\sum_{u=1}^n \left|x_{iu} - x_{ju}
ight|^p
ight)^{rac{1}{p}}$$

p=2 退化成欧式距离(Euclidean distance)

$$ext{dist}_{ ext{ed}}\left(oldsymbol{x}_i, oldsymbol{x}_j
ight) = \left\|oldsymbol{x}_i - oldsymbol{x}_j
ight\|_2 = \sqrt{\left|\sum_{u=1}^n \left|x_{iu} - x_{ju}
ight|^2}$$

p=1 退化成曼哈顿距离(Manhattan distance)

$$ext{dist}_{ ext{man}}\left(oldsymbol{x}_i, oldsymbol{x}_j
ight) = \left\|oldsymbol{x}_i - oldsymbol{x}_j
ight\|_1 = \sum_{u=1}^n \left|x_{iu} - x_{ju}
ight|$$

常用的距离度量 - 离散无序

VDM (Value Difference Metric) 度量。

$$ext{VDM}_p(a,b) = \sum_{i=1}^k \left| rac{m_{u,a,i}}{m_{u,a}} - rac{m_{u,b,i}}{m_{u,b}}
ight|^p$$

表 4.1 西瓜数据集 2.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
- 8-	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	 沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	,沉闷	稍糊	稍凹	硬滑	否

kmeans (原型聚类)



原型(prototype)指类结构能通过一组典型的特例刻画。比如男、女类似的。 给定样本集 $D = \{x_1, x_2, \dots, x_m\}$,k均值算法针对聚类所得簇划分 $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$,求解最小化平方误差问题

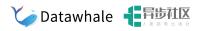
$$E = \sum_{i=1}^k \sum_{oldsymbol{x} \in C_i} \left\| oldsymbol{x} - oldsymbol{\mu}_i
ight\|_2^2$$

其中 $oldsymbol{\mu}_i = rac{1}{|C_i|} \sum_{oldsymbol{x} \in C_i} oldsymbol{x}$ 表示簇 \mathcal{C}_i 的均值向量。

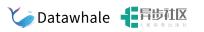
求解改式需要考虑样本集D所有可能的划分,是一个NP-hard问题。一般来说,我们采用 迭代算法求解近似划分。

kmeans 算法和示例

```
输入: 样本集 D = \{x_1, x_2, \ldots, 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, \ldots, \mu_k\}
 2: repeat
 3: \diamondsuit C_i = \varnothing \ (1 \leqslant i \leqslant k)
     for j = 1, 2, ..., m do
        计算样本 x_j 与各均值向量 \mu_i (1 \le i \le k) 的距离: d_{ji} = ||x_j - \mu_i||_2;
         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,...,k\}} d_{ji};
 6:
         将样本 x_i 划入相应的簇: C_{\lambda_i} = C_{\lambda_i} \cup \{x_i\};
 8:
      end for
      for i = 1, 2, ..., k do
        计算新均值向量: \mu'_i = \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x};
10:
        if \mu_i' \neq \mu_i then
11:
            将当前均值向量 \mu_i 更新为 \mu'_i
12:
13:
         else
            保持当前均值向量不变
14:
         end if
15:
      end for
16:
17: until 当前均值向量均未更新
输出: 簇划分 \mathcal{C} = \{C_1, C_2, \dots, C_k\}
```



密度聚类



密度聚类假设聚类结构能够通过样本分布的紧密程度确定。它从样本密度的角度考察样本间的可连接性,并基于可连接样本不断扩展聚类簇得到最终的聚类结果。

DBSCAN是密度聚类的代表之一。它基于一组邻域参数 $(\epsilon, MinPts)$ 刻画样本分布的紧密程度。关于DBSCAN的几个概念如下:

- 1. ϵ -邻域,和样本x距离不超过 ϵ 的样本集合
- 2. 核心对象:如果样本x的 ϵ -邻域内至少包含MinPts个样本,则x是一个核心对象。
- 3. 密度直达: x_j 位于核心对象 x_i 的 ϵ -邻域内,则称 x_i 密度可达 x_j
- 4. 密度可达:若存在样本序列 $x_i, p_1, p_2, ..., p_n, x_j$,其中 p_i 密度直达 p_{i+1} ,则称 x_i 密度可达 x_j
- 5. 密度相连:如上序列中任意两点密度相连

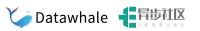
DBSCAN

DBSCN定义的簇为:最大密度相连的 样本集合为一个簇。

- 1. 连接性: 同一个簇内任意两样本 必然密度相连
- 2. 最大性:密度可达的两个样本必定属于同一个簇

```
输入: 样本集 D = \{x_1, x_2, \ldots, 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, ..., m do
       确定样本 x_i 的 \epsilon-邻域 N_{\epsilon}(x_i);
       if |N_{\epsilon}(\boldsymbol{x}_i)| \geqslant MinPts then
           将样本 x_i 加入核心对象集合: \Omega = \Omega \cup \{x_i\}
       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_{\text{old}} = \Gamma;
       随机选取一个核心对象 \mathbf{o} \in \Omega, 初始化队列 Q = \langle \mathbf{o} \rangle;
13: \Gamma = \Gamma \setminus \{\boldsymbol{o}\};
14: while Q \neq \emptyset do
          取出队列 Q 中的首个样本 q;
           if |N_{\epsilon}(q)| \geqslant MinPts then
             \diamondsuit \Delta = N_{\epsilon}(\boldsymbol{q}) \cap \Gamma;
17:
             将 \Delta 中的样本加入队列 Q;
      \Gamma = \Gamma \setminus \Delta;
19:
           end if
21: end while
22: k = k + 1, 生成聚类簇 C_k = \Gamma_{\text{old}} \setminus \Gamma;
23: \Omega = \Omega \setminus C_k
24: end while
输出: 簇划分 \mathcal{C} = \{C_1, C_2, \dots, C_k\}
```

层次聚类



层次聚类试图将数据划分成为不同的层次,因此聚类结果呈现明显的树状结构。 AGNES是一种采用自底向上聚合策略的层次聚类算法。在聚类过程中不断合并距离最近的两个类簇,知道达到预期的聚类簇数目。算法的核心在于如何定义类簇中之间的距离。

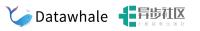
- 1. 最小距离(两个簇最近样本距离): $d_{\min}\left(C_i,C_j
 ight) = \min_{x \in C_i, oldsymbol{z} \in C_j} \operatorname{dist}(oldsymbol{x},oldsymbol{z})$
- 2. 最大距离(两个簇最远样本距离): $d_{\max}\left(C_i,C_j
 ight) = \max_{x \in C_i, oldsymbol{z} \in C_j} \operatorname{dist}(oldsymbol{x},oldsymbol{z})$
- 3. 平均距离(两个簇两两样本距离均值):

$$d_{ ext{avg}}\left(C_{i},C_{j}
ight) = rac{1}{\left|C_{i}
ight|\left|C_{j}
ight|}\sum_{oldsymbol{x}\in C_{i}}\sum_{oldsymbol{z}\in C_{j}}\operatorname{dist}(oldsymbol{x},oldsymbol{z}
ight)$$

AGNES 算法

```
过程:
 1: for j = 1, 2, \ldots, m do
 2: C_i = \{x_i\}
 3: end for
 4: for i = 1, 2, ..., m do
 5: for j = 1, 2, ..., m do
    M(i,j) = d(C_i, C_j);
     M(j,i) = M(i,j)
      end for
 9: end for
10: 设置当前聚类簇个数: q = m
11: while q > k do
      找出距离最近的两个聚类簇 C_{i*} 和 C_{i*};
12:
      合并 C_{i^*} 和 C_{i^*}: C_{i^*} = C_{i^*} \bigcup C_{i^*};
13:
      for j = j^* + 1, j^* + 2, \dots, q do
14:
        将聚类簇 C_i 重编号为 C_{i-1}
15:
      end for
16:
      删除距离矩阵 M 的第 j^* 行与第 j^* 列;
17:
      for j = 1, 2, ..., q - 1 do
18:
      M(i^*, j) = d(C_{i^*}, C_j);
19:
        M(j, i^*) = M(i^*, j)
20:
      end for
21:
22:
      q = q - 1
00 --- -- --- --- --- --- --- ---
```

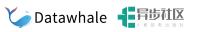
预告



下一节:降维和度量学习

西瓜书对应章节:第10章

结束语



欢迎加入【南瓜书读者交流群】,我们将在群里进行答疑、勘误、本次直播回放、本次直播PPT发放、下次直播通知等最新资源发放和活动通知。加入步骤:

- 1. 关注公众号【Datawhale】,发送【南瓜书】三个字获取机器人"小豚"的微信二维码
- 2. 添加"小豚"为微信好友, 然后对"小豚"发送【南瓜书】三个字即可自动邀请进群

