控制与决策
Control and Decision

考虑边界样本邻域归属信息的粗糙K-means增量聚类算法

马福民, 孙静勇, 张腾飞

引用本文:
马福民,孙静勇,张腾飞. 考虑边界样本邻域归属信息的粗糙K-means增量聚类算法[J]. 控制与决策, 2022, 37(11): 2968-2976.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.0624

考虑边界样本邻域归属信息的粗糙K-means增量聚类算法

马福民[1†], 孙静勇[1], 张腾飞[2]

(1. 南京财经大学 信息工程学院，南京 210023；2. 南京邮电大学 自动化学院、人工智能学院，南京 210023)

摘 要: 在原有数据聚类结果的基础上,如何对新增数据进行归属度量分析是提高增量式聚类质量的关键,现有增量式聚类算法更多地是考虑新增数据的位置分布,忽略其邻域数据点的归属信息. 在粗糙K-means聚类算法的基础上,针对边界区域新增数据点的不确定性信息处理,提出一种基于邻域归属信息的粗糙K-means增量式聚类算法. 该算法综合考虑边界区域新增数据样本的位置分布及其邻域数据点的类簇归属信息,使得新增数据点与各类簇的归属度量更为合理;此外,在增量式聚类过程中,根据新增数据点所导致的类簇结构的变化,对类簇进行相应的合并或分裂操作,使类簇划分可以自适应调整. 在人工数据集和UCI标准数据集上的对比实验结果验证了算法的有效性.

关键词: 粗糙K-means聚类；增量聚类；邻域归属信息；类簇结构

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.0624 **开放科学(资源服务)标识码(OSID):**

引用格式: 马福民, 孙静勇, 张腾飞. 考虑边界样本邻域归属信息的粗糙K-means增量聚类算法[J]. 控制与决策, 2022, 37(11): 2968-2976.

Rough K-means incremental clustering algorithm considering neighborhood belonging information of boundary samples

MA Fu-min[1†], *SUN Jing-yong*[1], *ZHANG Teng-fei*[2]

(1. College of Information Engineering，Nanjing University of Finance and Economics，Nanjing 210023，China；2. College of Automation & College of Artificial Intelligence，Nanjing University of Posts and Telecommunications，Nanjing 210023，China)

Abstract: The key to improve the quality of incremental clustering is how to assign the new data to different clusters on the basis of original data clustering results. The existing incremental clustering algorithms mostly consider the location distribution of the newly added data point, and ignore the belonging information of the neighbor points around the new data point. To deal with the uncertain information of new data points that fall into boundary regions of original clusters, based on the rough K-means clustering, a rough K-means incremental clustering algorithm is developed. In this algorithm, focusing on the assignment of the newly added data in the boundary region, the neighborhood belonging information of the new data is taken into consideration, so that the hybrid measure of the new data point belonging to different clusters is more reasonable. Furthermore, the clusters will be merged or split to make the new divided clusters becoming more reasonable according to the cluster structure changes caused by the new data. The validity of the proposed algorithm is demonstrated by the experimental results on the artificial data sets and UCI standard data sets.

Keywords: rough K-means clustering；incremental clustering；neighborhood belonging information；cluster structure

0 引 言

聚类是一种无监督的分类算法,通过相似性度量将相似的数据划分到同一类簇、相异的数据划分到不同的类簇[1-2]. 聚类算法大致可以划分为基于划分的聚类[3]、基于层次的聚类[4]、基于密度的聚类[5]和基于网格的聚类[6]. K-means是最为经典的基于划分的聚类算法,在数据挖掘分析领域得到了广泛应用[7-8]. 针对K-means算法无法有效处理含有模糊不确定信息、类簇交叉的数据聚类分析问题,Lingras等[9-10]将粗糙集理论融入K-means算法,将具有不确定性归属关系的数据划分到多个类簇的边界区域,聚类结果更为客观. 如何对交叉边界区域的数据对

收稿日期: 2021-04-12；**录用日期:** 2021-07-29.

基金项目: 国家自然科学基金项目(61973151, 62073173)；江苏省自然科学基金项目(BK20191406, BK20191376).

责任编委: 阳春华.

†通讯作者. E-mail: fmmatj@126.com.

象进行度量与处理,成为近些年粗糙K-means(Rough K-means)及其衍生算法研究的焦点[11-13].

此外,现有粗糙K-means相关算法大多只是针对静态数据的处理,当不断有新的数据加入时,需要将数据重新进行聚类,严重浪费计算资源[14]. 增量聚类是指利用数据更新前已获取的聚类结果,对新增数据逐个或逐批次地进行更新聚类的过程[15],在各领域泛在智能感知快速发展的时代背景下,增量聚类更具有现实意义,也已引起国内外学者的广泛关注. Ester等[16]较早提出基于密度的增量聚类算法(incremental-DBSCAN),结合密度度量实现新增数据的类簇划分. 文献[17]引入相对密度度量,提出了基于相对密度的混合属性数据增量聚类算法. 文献[18]针对高维数据,采用马氏距离对新增数据的位置分布进行度量,并将DBSCAN与SVM相结合,提出了一种DBSCAN-SVM增量聚类算法类. 文献[19]根据新增数据的位置分布动态调整类簇的均值向量和协方差矩阵,提出了基于贝叶斯自适应共振理论的增量聚类算法. 基于模糊隶属度对新增数据进行度量也是较为常用的方法,文献[20-21]提出了模糊K-means增量聚类算法,首先针对离线数据计算得到初始类簇中心,再针对在线数据运行增量模糊聚类,可以更新现有的类簇也可以产生新类簇,但相关的控制参数需事先主观设置以控制离群点的辨识、新类簇的生成以及空类簇的删除. 文献[22]提出了一种结合密度和模糊度量的增量聚类算法,在局部类簇分配过程中引入模糊局部聚类,为提高聚类质量、简化参数选择过程,采用改进的山谷搜索算法自适应地确定离群值以生成最终的聚类.

上述增量聚类算法更多是考虑新增数据的位置分布,即依据新增数据点相对于类簇中心的距离或所在位置的密度来度量其类簇归属,并没有区分关注新增数据点是否落入现有类簇的边界区域,也忽略了其邻域数据点的归属信息. 实际上,落入类簇交叉边界区域的新增数据点含有更多的不确定性信息,需要重点分析处理;且边界区域新增数据点的类簇归属不仅与其所处的位置分布有关,邻域范围内原有数据点的归属信息,即邻域归属信息,也是非常重要的参考度量. 此外,新增数据点的不断加入,类簇的结构也会随之发生变化,需要实时更新调整. 鉴于此,本文提出一种考虑边界样本邻域归属信息的粗糙K-means增量聚类算法,聚焦边界区域的新增数据点,综合考虑其位置分布和邻域数据点归属信息,并根据类簇结构的变化对相应的类簇进行合并或分裂操作,以提高增量聚类的质量.

1　粗糙K-means及增量操作

1.1　粗糙K-means聚类

在K-means聚类算法中,通过引入粗糙集上下近似的概念,将每一个类簇划分为具有明确归属关系的下近似区域和具有不确定性归属关系的边界区域,对数据样本的划分更为客观. 将第i个类簇表示为C_i,类簇中心为$v_i, i=1,2,\ldots,k$. 为便于表述,将类簇C_i的上、下近似区域分别表示为$\underline{C}_i$和$\overline{C}_i$,边界区域为$\hat{C}_i=\underline{C}_i-\overline{C}_i$. 在粗糙$K$-means聚类算法中,数据对象$x_n$具有如下特性[9]:

1) 数据对象x_n只能被划分到至多一个类簇的下近似;

2) 若数据对象x_n不属于任何一个类簇的下近似,则至少属于两个或两个以上类簇的边界区域.

在Lingras等[9]最初的粗糙K-means聚类算法中,类簇中心的迭代计算公式如下:

$$v_i=\begin{cases} w_l\dfrac{\sum\limits_{x_n\in\underline{C}_i}x_n}{|\underline{C}_i|}+w_b\dfrac{\sum\limits_{x_n\in\hat{C}_i}x_n}{|\hat{C}_i|}, & \hat{C}_i\neq\phi;\\ w_l\dfrac{\sum\limits_{x_n\in\underline{C}_i}x_n}{|\underline{C}_i|}, & \text{otherwise.}\end{cases}\tag{1}$$

其中:参数w_l和w_b为下近似和边界区域的相对重要性权重系数,$w_l+w_b=1$,通常情况下$w_l>w_b$;$|*|$为集合$*$中数据对象个数.

考虑到下近似或边界为空集的情况,Peters[11]修正了类簇中心的迭代计算公式如下:

$$v_i=\begin{cases} w_l\dfrac{\sum\limits_{x_n\in\underline{C}_i}x_n}{|\underline{C}_i|}+w_b\dfrac{\sum\limits_{x_n\in\hat{C}_i}x_n}{|\hat{C}_i|}, & \underline{C}_i\neq\phi\bigwedge\hat{C}_i\neq\phi;\\ \dfrac{\sum\limits_{x_n\in\underline{C}_i}x_n}{|\underline{C}_i|}, & \underline{C}_i\neq\phi\bigwedge\hat{C}_i=\phi;\\ \dfrac{\sum\limits_{x_n\in\hat{C}_i}x_n}{|\hat{C}_i|}, & \underline{C}_i=\phi\bigwedge\hat{C}_i=\phi.\end{cases}\tag{2}$$

step 1: 初始化. 包括类簇个数K,每个簇的初始中心$v_i(i=1,2,\ldots,K)$,下近似和边界集的相对权重系数w_l、w_b,距离判断阈值δ,最大迭代次数$I_{\max}$.

step 2: 数据对象到各类簇的划分. 根据每一个数据对象$x_n(n=1,2,\ldots,N)$到各类簇中心的距离,将其划分到对应类簇的下近似集$\underline{C}_i$或边界区域$\hat{C}_i$.

step 3: 类簇中心点的迭代计算. 由式(2)计算各

个类簇的中心点.

step 4: 算法终止判断. 若各类簇中心不再发生变化或已达到设定迭代次数,则算法终止,否则返回step 2重新进行迭代计算.

1.2 增量操作

对于新增数据$x_{\rm new}$的插入,可能会出现以下几种情况[18]:

1) 新增数据$x_{\rm new}$被划分至已有的类簇,且不改变原有的划分结果,如图1所示,圆点为原数据点,三角形为新增数据点.

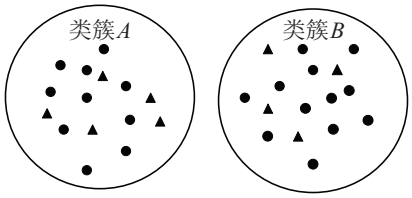

图 1 新增数据划分至已有类簇

2) 新增数据$x_{\rm new}$造成了原划分结果的变化,导致类簇的合并或者分裂,如图2和图3所示.

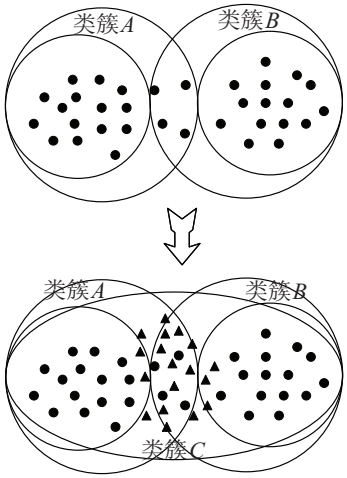

图 2 新增数据导致类簇合并

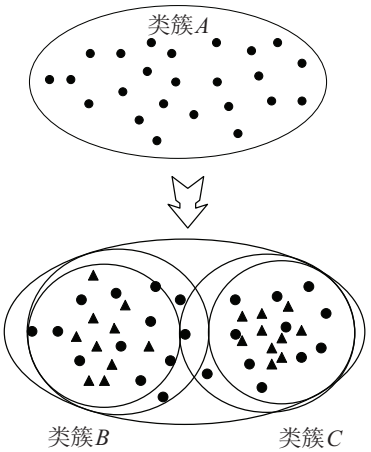

图 3 新增数据导致类簇分裂

3) 新增数据$x_{\rm new}$被划分为孤立点或与原孤立点合并形成新的类簇,如图4所示.

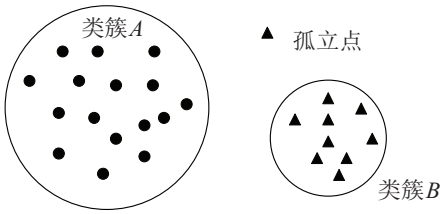

图 4 孤立点和新增簇

2 考虑边界样本邻域归属信息的增量聚类算法

2.1 考虑邻域归属信息的新增数据点划分方法

在传统的粗糙K-means聚类算法中,对于数据的划分采用数据对象与类簇中心点之间的距离进行衡量. 对于具有明确归属关系的新增数据对象,可以将其划分到与其距离最近的类簇中去,但对位于几个可能归属类簇边界区域的新增数据对象,仅仅依据距离度量很难判断该数据点更大概率应该归属于哪个类簇. 实际上,边界区域新增数据点的类簇归属不仅与其所处的位置分布有关,其邻域范围内原有数据点的归属信息,即邻域归属信息,也是非常重要的参考度量.

为了进一步对边界区域新增数据点进行描述,考虑其邻域归属信息,定义相对于类簇的综合归属度量为

$$h_{ik}=\frac{|\overline{f_{\xi_{ik}}(x_k)}|}{|L_\xi(x_k)|}\sum_{x_z\in\overline{f_{\xi_{ik}}(x_k)}}\exp\left(-\frac{d_{zk}}{2\xi^2}\right).\tag{3}$$

其中: $|L_\xi(x_k)|$为数据点x_k半径为ξ的邻域内的数据个数, $|\overline{f_{\xi_{ik}}(x_k)}|$为数据点$x_k$的$\xi$邻域内属于类簇$C_i$的数据个数, $\sum\limits_{x_z\in\overline{f_{\xi_{ik}}(x_k)}}\exp\left(-\frac{d_{zk}}{2\xi^2}\right)$计算数据点与所属类簇的距离度量, d_{zk}为数据点x_z与x_k之间的欧氏距离.

基于上述综合归属度量,在对某一新增数据点x_k进行划分时,若x_k邻域内的数据点均为某一类簇的数据,则将x_k直接划分至该类簇的下近似中,若x_k邻域内包含多个类簇的数据,则通过设置阈值判断数据点的归属,此时x_k很有可能被划分至多个类簇的边界区域中,或者形成新的类簇. 若数据点x_k邻域内没有数据,或者邻域内数据点均为少数孤立点时,则将x_k判断为孤立点.

相应地,类簇中心的迭代计算公式更新如下:

$$v_i'=\begin{cases}v_{i\rm old}+w_l\dfrac{x_k-v_{i\rm old}}{|\underline{C}_i|+1},\ x_k\in\underline{C}_i;\\ v_{i\rm old}+w_b\dfrac{h_{ik}}{\sum\limits_{z=1}^{K}h_{zk}}\dfrac{x_k-v_{i\rm old}}{|\hat{C}_i|+1},\ x_k\in\hat{C}_i.\end{cases}\tag{4}$$

其中: $v_{i\text{old}}$ 为类簇 C_i 之前的类簇中心, v_i' 为插入新增数据点之后类簇 C_i 新的类簇中心.

2.2　类簇结构信息度量

在增量聚类中, 由于类簇的结构可能会随着新增数据的加入而发生变化, 需要对类簇的结构做进一步描述. 结合粗糙 K-means 算法, 定义类簇 C_i 的类簇信息为

$$\text{CF}_i = (v_i, \{\text{den}_k | k = 1, 2, \ldots, |\overline{C}_i|\}, \text{dis}_i, \text{Den}_i).$$

其中: $\text{den}_k = |L_\xi(x_k)|/\xi$ 为每个数据 x_k 的 ξ 邻域密度, ξ 为数据点 x_k 的邻域半径, $|L_\xi(x_k)|$ 为数据点 x_k 的 ξ 邻域内数据点的个数; $\text{dis}_i = \dfrac{\sum\limits_{x_k, x_j \in \overline{C}_i} d(x_k, x_j)}{|\overline{C}_i|(|\overline{C}_i| - 1)}$ 为类簇中数据点的平均距离, $|\overline{C}_i|$ 为类簇上近似的数据个数; $\text{Den}_i = \sum\limits_{x_k \in \overline{C}_i} \text{den}_k \Big/ |\overline{C}_i|$ 为类簇密度, 即该类簇中所有数据点的平均密度.

2.3　类簇的合并或分裂操作

数据的插入有可能会导致类簇的合并或者分裂, 在分类质量较好的类簇结构中, 类簇的密度通常情况下由类簇中心至类簇边缘逐渐减少, 类簇中心的数据点相对较密、类簇边界区域的数据点相对较为稀疏.

1) 当新增数据点的加入使得两个类簇边界区域的邻域密度比当前两个类簇的密度大, 且边界区域的数据点与两个类簇中心点之间的距离均小于这些类簇数据点的平均距离时, 则合并这两个类簇. 即新增数据点 $x_{\text{new}} \in \hat{C}_A$ 且 $x_{\text{new}} \in \hat{C}_B$, 若满足以下条件, 则合并类簇 A 和 B:

$$\text{den}_{\text{new}} > \text{Den}_A \bigwedge \text{den}_{\text{new}} > \text{Den}_B, \tag{5}$$

$$d(x_{\text{new}}, v_A) < \text{dis}_A \bigwedge d(x_{\text{new}}, v_B) < \text{dis}_B. \tag{6}$$

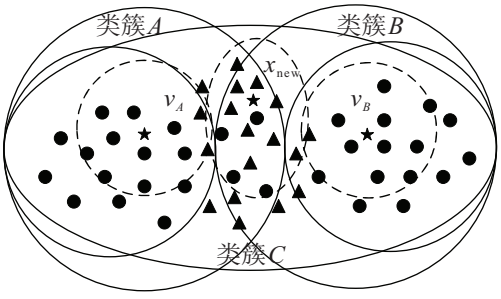

图 5　类簇合并示意图

如图 5 所示. 可以看出, x_{new} 的插入导致类簇 A 和 B 边界区域的密度比其所属类簇 A 和 B 的类簇密度都大, 且边界区域的数据点到类簇 A、B 中心 v_A、v_B 的距离都较近, 导致类簇 A 与 B 之间出现了较强的密度联通性, 显然, 将这两个类簇合并为一个类簇更为合理.

2) 当某个类簇需要分裂时, 表明类簇中出现了两个密度较高且相互距离较远的区域, 即在类簇 A 中, 新增数据点 $x_{\text{new}} \in \underline{C}_A$, 且在其所属区域 B 和类簇中心区域 C 形成两个高密度区域, 若满足以下条件, 则需对类簇 A 进行分裂:

$$\text{den}_{\text{new}} > \text{Den}_C, \tag{7}$$

$$d(x_{\text{new}}, v_C) > \text{dis}_C. \tag{8}$$

类簇分裂条件示意图如图 6 所示. 可以看出, 因为数据点 x_{new} 的插入, 导致类簇中出现了两块密度较高的区域, 且这两块区域之间的距离较远, 显然, 将该类簇分裂为两个类簇更为合理.

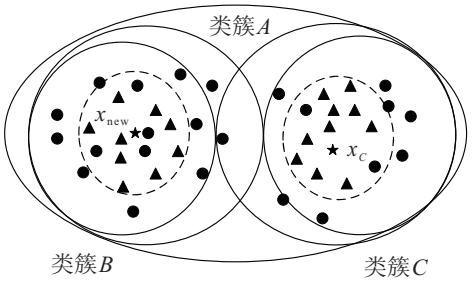

图 6　类簇分裂条件示意图

2.4　基于邻域归属信息的粗糙 K-means 增量算法

根据新增数据点划分以及类簇的分裂与合并操作, 设计基于邻域归属信息的粗糙 K-means 增量聚类算法 (rough K-means incremental clustering algorithm based on neighborhood belonging information, IRKM), 描述如下.

算法 1　IRKM 算法.

输入: 数据集 $D = \{x_1, x_2, \ldots, x_N\}$, 类簇个数 K, 上下近似权重 w_l 和 w_b, 邻域半径 ξ, 距离判决阈值 δ, 粗糙 K-means 聚类结果, 包括类簇 C_i, 中心 v_i, 新增数据 x_k, 最大迭代次数 T;

输出: 类簇结构信息和类簇划分结果.

```
while 存在新增数据点 x_k
  for j = 1; j ⩽ K; j++
    计算 d(x_k, v_j) 并划分 x_k
    if x_k ∈ C_j
      通过式(4)更新类簇中心 v_j
    else
      通过式(3)计算 h_jk (x_k ∈ Ĉ_j, j ∈ {1, 2, ..., K})
      通过式(4)更新类簇中心 v_j
    end
    计算 den_k
    if x_k ∈ C_j
      if  den_k > Den_j and d(x_k, v_j) > dis_j
```

通过RKM($K=2, v=\{x_k, v_j\}, w_l, w_b, \delta, T$)分裂$C_j$

更新CF_j

else if $x_k \in \hat{C}_j$ ($j \in \{1,2,\ldots,K\}$)

if $\mathrm{den}_k > \mathrm{Den}_j$ and $d(x_k, v_j) < \mathrm{dis}_j$

合并类簇$C_j(x_k \in \hat{C}_j, j \in \{1,2,\ldots,K\})$

更新CF_j

else if $\mathrm{den}_k = 0$

将x_k划分为噪声点

if $\mathrm{den}_k > \min\limits_{j=1,2,\ldots,K} \mathrm{Den}_j$

创建类簇C_{K+1}

更新CF_{K+1}

end

if 出现分裂、合并、创建类簇的行为

运行RKM

end

return 类簇结构信息和类簇划分结果

算法在一次增量操作中所耗费时间$O(N)$,即为在计算x_k的邻域密度时所需的时间,当出现创建、合并、分裂类簇时,需要$O(tN)$的时间,其中t为粗糙K-means算法的迭代次数,N为当前数据集的数据个数,但一般情况下创建、合并和分裂操作较少,所以一次增量操作的平均耗费时间为$O(N)$.

3 实验分析

采用Silhouette指标和DBI指标评价聚类结果的质量. Silhouette指标用来描述类簇的簇内紧密性和簇间分离性,其在评价球形类簇聚类质量和估计最佳聚类个数上均有良好的效果[23]. DBI指数的目的是在最小化各类簇簇内离散度的同时最大化簇间距离[12]. Silhoutte值越大表示聚类效果越好,而DBI值越小表示聚类效果越好. 为验证算法有效性,将所提出算法在人工模拟数据集和UCI数据集上进行实验,并将所提出增量粗糙K-means (IRKM)与静态粗糙K-means (RKM)[14]、增量模糊K-means (IFKM)[15]和批处理增量模糊K-means(AIFC)算法[24]进行对比分析.

3.1 人工数据集实验分析

采用3个不同分布的人工数据集Dataset 1、Dataset 2和Dataset 3,每个数据集均由原始数据和增量数据两部分组成. Dataset1的原始数据集由2类类簇组成,每类包含50个数据,新增数据集由一类簇组成,也包含50个数据;Dataset 2的原始数据集由3类类簇组成,第1类包含50个数据,第2、 3类各包含20个数据,新增数据集由1类簇组成,包含60个数据;Dataset 3的原始数据集由2类类簇组成,第1类包含50个数据,第2类包含30个数据,新增数据集由2类簇组成,各包含30个数据. 数据集分布如图7所示. 图7中,Dataset 1和Dataset 2星形数据点为新增数据,其他形状数据点为原始数据;Dataset 3星形和圆形数据点为新增数据,其他形状数据点为原始数据. Dataset 1测试增量算法新增1个类簇的功能,Dataset 2测试增量算法合并类簇的功能,Dataset 3测试增量算法分裂类簇的功能.

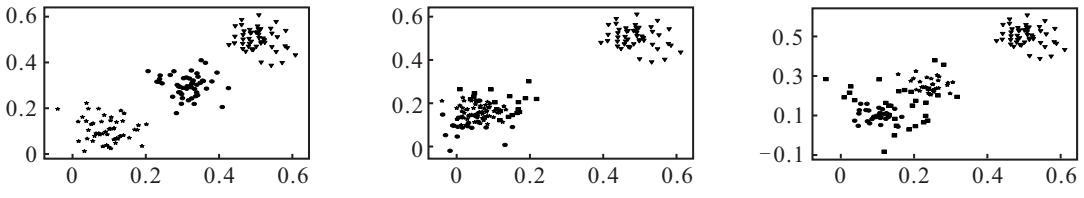

图7 人工数据集分布

聚类算法相关参数设置如表1所示. 表1中:K为初始聚类个数,w_l和w_b为RKM下近似集和边域集的权重系数,m为FKM模糊指数,δ为RKM算法的距离判决阈值,ξ为IRKM算法的邻域半径,T为最大迭代次数. 由于AIFC算法为批处理式的增量聚类算法,其新增数据以数据块形式到达,本次实验中每个新增数据块包含10个新增数据点,其他3种算法的新增数据点逐个添加.

3种算法的初始类簇中心均相同,对3个数据集的聚类结果如图8～图10和表2～表5所示.

表1 初始参数设置

数据集	相关参数设置						
	K	w_l	w_b	m	δ	ξ	T
Dataset 1	2	0.9	0.1	2	0.01	0.05	100
Dataset 2	3	0.9	0.1	2	0.01	0.05	100
Dataset 3	2	0.9	0.1	2	0.01	0.08	100

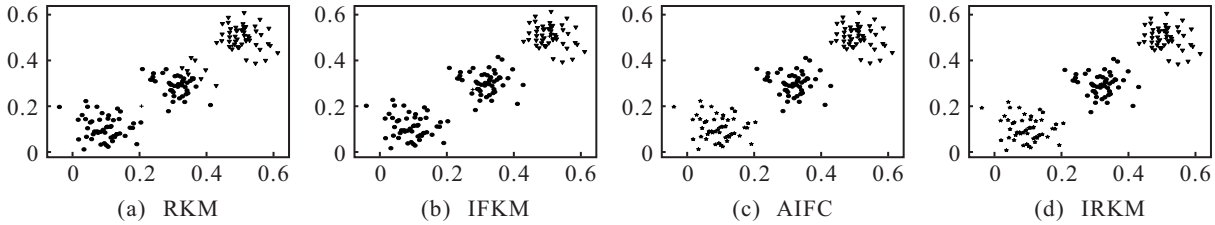

图 8　Dataset 1 聚类结果

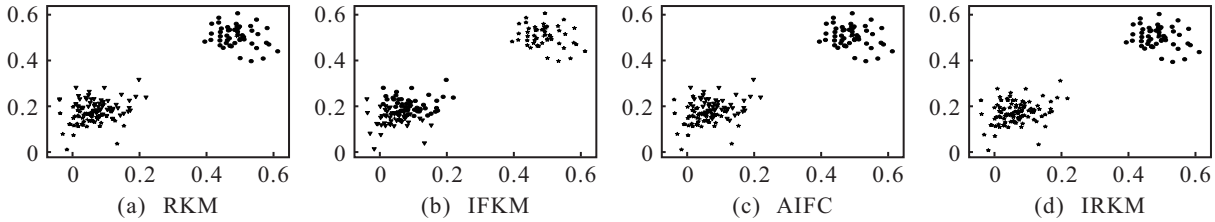

图 9　Dataset 2 聚类结果

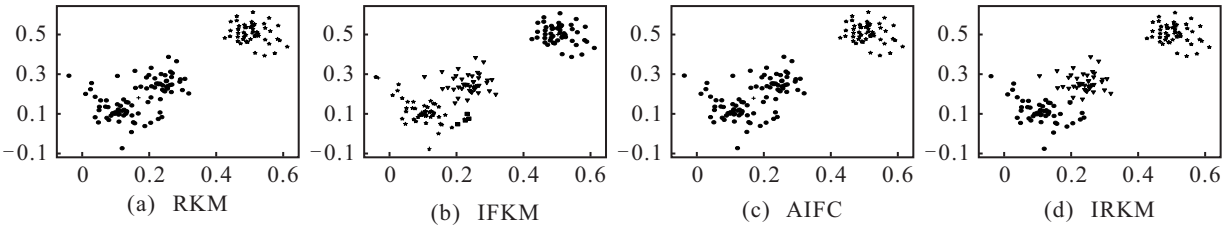

图 10　Dataset 3 聚类结果

表 2　最终类簇个数

数据集	最终类簇个数		
	RKM	IFKM	IRKM
Dataset 1	2	2	3
Dataset 2	3	3	2
Dataset 3	2	6	3

表 3　聚类时间对比

数据集	聚类时间/s		
	RKM	IFKM	IRKM
Dataset 1	0.567	**0.291**	0.465
Dataset 2	**1.076**	2.032	2.257
Dataset 3	**0.546**	0.569	0.945

表 4　Silhouette 指标对比

数据集	Silhouette 指标		
	RKM	IFKM	IRKM
Dataset 1	0.419	0.302	**0.703**
Dataset 2	0.489	0.514	**0.864**
Dataset 3	0.559	−0.557	**0.602**

表 5　DBI 指标对比

数据集	DBI 指标			
	RKM	IFKM	AIFC	IRKM
Dataset 1	0.806	0.841	**0.578**	**0.578**
Dataset 2	1.159	1.270	1.404	**0.318**
Dataset 3	0.490	0.655	0.490	**0.355**

由聚类结果可以看出, 在 3 类人工数据集上, IRKM 算法在 Silhouette 指标和 DBI 指标上均取得了最好的效果. 在 Dataset 1 上, IRKM 算法和 AIFC 算法成功创建了新的类簇, 而 IFKM 算法并未创建新的类簇, RKM 算法不会改变类簇的个数, 其最终的聚类个数始终是输入的值. 在 Dataset 2 上, IRKM 算法合并了类簇 2 和类簇 3, 由图 9 可以看出, 类簇 2 与类簇 3 之间具有一个密度较大的区域, 且与类簇中心之间的距离较小, 因此将类簇 2 和类簇 3 合并为一类是合理的. 在 Dataset 3 中, 类簇 2 中出现了 2 个密度较高的区域, 且两个区域之间的距离较大, 此时应该对类簇 2 进行分裂. 由图 10 可见, IRKM 算法将类簇 2 分为 2 类, IFKM 将类簇 2 分为 6 类, 而 AIFC 算法和 RKM 算法未能对类簇 2 进行分裂.

在聚类时间上, 在没有改变类簇结构的数据集 Dataset 1 上, 其他 3 种增量聚类算法的聚类时间开销比 RKM 算法低. 而在 Dataset 2 和 Dataset 3 上, AIFC 算法的聚类时间开销最小, 其次是 RKM 算法, 但这两种算法均未对 Dataset 2 和 Dataset 3 的类簇结构做出改变, 只是保持了原始输入的类簇个数. IRKM 在 Dataset 2 和 Dataset 3 上时间开销较多, 这是因为 IRKM 算法分别对 Dataset 2 和 Dataset 3 的类簇结构进行了合并和分裂的操作, 从而导致时间消耗较多. 综合来看, IRKM 算法效果相对较好.

3.2 UCI数据集实验分析

在UCI数据库中选取Iris、Wine、Wheat_seeds、Credit Approval和Pima Indians Diabetes五种不同规模的数据集进行聚类分析. 聚类前对所有数据集进行归一化处理,为了实现增量式聚类算法的验证分析,将所有数据集重新拆分为不同的静态数据和增量数据组合.

将Iris拆分成3个不同的数据集: iris 1从每个类簇随机抽取10个数据形成包含30个数据的增量数据,剩余120个数据作为静态数据; iris 2将第1类簇50个数据作为增量数据,剩余100个数据作为静态数据; iris 3从第2、3类簇中随机抽取15个数据形成包含30个数据的增量数据,剩余120个数据作为静态数据.

将Wine拆分成3个不同的数据集: wine 1从每个类簇随机抽取20个数据形成包含60个数据的增量数据,剩余118个数据作为静态数据; wine 2将第3类簇的48个数据作为增量数据,剩余130个数据作为静态数据; wine 3从第2、3类簇中随机抽取20个数据形成包含40个数据的增量数据,剩余138个数据作为静态数据.

将Wheat_seeds拆分成3个不同的数据集: seed 1从每个类簇随机抽取20个数据形成包含60个数据的增量数据,剩余150个数据作为静态数据; seeds 2将第3类簇的70个数据作为增量数据,剩余140个数据作为静态数据; seeds 3从第2、3类簇中随机抽取20个数据形成包含40个数据的增量数据,剩余170个数据作为静态数据.

表6 算法参数设置

数据集	相关参数设置						
	K	w_l	w_b	m	δ	ξ	T
iris 1	3	0.9	0.1	2	0.01	1.0	100
iris 2	2	0.9	0.1	2	0.01	1.2	100
iris 3	4	0.9	0.1	2	0.01	0.8	100
wine 1	3	0.8	0.2	2	0.01	0.75	100
wine 2	2	0.8	0.2	2	0.01	0.70	100
wine 3	4	0.8	0.2	1.5	0.01	0.75	100
seeds 1	3	0.8	0.2	1.5	0.01	0.40	100
seeds 2	2	0.8	0.2	1.5	0.01	0.45	100
seeds 3	4	0.8	0.2	1.5	0.01	0.45	100
credit 1	2	0.9	0.1	2	0.01	0.7	100
credit 2	3	0.9	0.1	2	0.01	0.7	100
pima 1	2	0.9	0.1	2	0.01	0.48	100
pima 2	3	0.9	0.1	2	0.01	0.6	100

将Credit Approval拆分成两个不同的数据集, credit 1和credit 2均是从数据集中随机抽取200个数据作为新增数据,每次抽取后剩余的490个数据作为静态数据. 将Pima Indians Diabetes也拆分成两个不同的数据集, pima 1和pima 2均是从数据集中随机抽取200个数据作为新增数据,每次抽取后剩余的568个数据作为静态数据.

在对同一数据集进行测试时使用相同的初始聚类中心与初始参数. 对于AIFC算法,5种数据集下的每个新增数据块中均包含10个新增数据点,只有在wine 2数据集中,最后一个数据块中包含8个数据. 在对算法参数进行设置时选择最优参数组合,参数设置如表6所示,实验结果如表7～表10所示.

表7 聚类时间对比

数据集	聚类时间/s			
	RKM	IFKM	AIFC	IRKM
iris 1	0.844	1.208	**0.334**	0.796
iris 2	0.835	0.320	**0.276**	0.606
iris 3	1.539	0.590	**0.301**	0.711
wine 1	15.880	0.617	**0.273**	1.324
wine 2	1.657	0.501	**0.336**	1.206
wine 3	3.396	0.932	**0.627**	1.240
seeds 1	22.759	3.790	**0.314**	2.674
seeds 2	1.099	**0.105**	0.330	2.367
seeds 3	33.804	**0.278**	0.436	1.937
credit 1	8.353	3.283	**1.406**	3.893
credit 2	32.344	12.638	**2.276**	5.480
pima 1	48.341	2.667	**2.358**	5.223
pima 2	56.343	2.473	**1.516**	6.424

表8 Silhouette指标对比

数据集	Silhouette指标对比			
	RKM	IFKM	AIFC	IRKM
iris 1	0.422	**0.452**	0.359	0.403
iris 2	0.493	0.589	0.589	**0.606**
iris 3	**0.537**	0.452	0.530	0.438
wine 1	0.334	−0.976	0.335	**0.348**
wine 2	0.090	−0.893	−0.252	**0.300**
wine 3	0.066	−0.769	−0.133	**0.319**
seeds 1	0.437	−0.490	0.416	**0.447**
seeds 2	0.283	0.105	0.130	**0.442**
seeds 3	0.339	−0.867	0.331	**0.428**
credit 1	**0.338**	−0.819	0.202	0.337
credit 2	−0.057	−0.913	−0.697	**0.207**
pima 1	0.403	−0.983	0.379	**0.405**
Pima 2	0.210	−0.892	−0.749	**0.400**

表 9　DBI 指标对比

数据集	DBI 指标对比			
	RKM	IFKM	AIFC	IRKM
iris 1	0.935	1.010	1.007	**0.899**
iris 2	1.579	0.952	0.956	**0.939**
iris 3	1.084	1.548	1.170	**1.003**
wine 1	2.164	3.196	**2.063**	2.146
wine 2	1.912	3.624	2.887	**1.883**
wine 3	2.423	5.801	4.107	**2.094**
seeds 1	1.313	5.513	1.374	**1.267**
seeds 2	2.237	2.110	1.929	**1.233**
seeds 3	1.508	2.331	1.548	**1.278**
credit 1	**2.767**	4.918	3.676	2.851
credit 2	2.263	4.503	4.472	**2.187**
pima 1	2.512	7.836	2.602	**2.499**
pima 2	2.547	7.119	5.772	**2.494**

表 10　最终类簇个数

数据集	最终类簇个数			
	RKM	IFKM	AIFC	IRKM
iris 1	3	3	3	3
iris 2	2	3	3	3
iris 3	4	2	4	3
wine 1	3	3	3	3
wine 2	2	5	4	3
wine 3	4	29	4	3
seeds 1	3	16	3	3
seeds 2	2	2	2	3
seeds 3	4	4	4	3
credit 1	2	12	2	2
credit 2	3	10	4	2
pima 1	2	8	2	2
pima 2	3	8	4	2

由上述聚类结果可以看出,在 Silhouette 指标上,IRKM 算法在 iris 2、wine 1、wine 2、wine 3、seed 1、seeds 2、 seeds 3、 credit 2、 pima 1 和 pima 2 这 10 个数据集上均取得了最好的效果,仅在 iris 1 和 iris 3 上低于 RKM 和 IFKM,在 credit 1 上略低于 RKM 算法. RKM 算法在 iris 3 上聚类效果最好,在 wine 1 上聚类效果略低于 IRKM,在其他数据集上聚类效果不佳,而 IFKM 算法在 iris 1 和 credit 1 上聚类效果最好,在其他数据集上均表现不佳. AIFC 算法在所有数据集上均没有取得最好的聚类效果,仅在 iris 2 和 iris 3 上取得了次好的聚类效果. 在 DBI 指标上,除了在数据集 wine 1 上低于 AIFC 算法,在 credit 1 上低于 RKM 算法,IRKM 在其他算法上均得到了最佳效果. 由此可见,相比于其他 3 种算法,IRKM 算法的聚类质量最佳.

在聚类时间上,IRKM 算法优于 RKM 算法但低于其他两类增量聚类算法,IFKM 算法在 seeds 2、seeds 3 上时间开销最少,AIFC 算法在 iris 1、 iris 2、iris 3、wine 1、wine 2、wine 3、seeds 1、credit 1、credit 2、pima 1 和 pima 2 上时间开销最少,RKM 算法则是时间开销最大的算法. 在最终类簇个数上,IRKM 算法在不同数据集上都得到了合理的类簇个数,RKM 算法在聚类时不会改变类簇的个数,所以最终的类簇数便是算法设定的值. AIFC 算法在 iris 1、iris 2、wine 1、seeds 1、 credit 1 和 pima1 上得到了合理的类簇个数,其他数据集上则没有得到合理的类簇个数. IFKM 算法在 iris 1 和 iris 2 上可以得到合理的类簇个数,但在其他数据集上倾向于将数据划分为更多的类簇,从而影响了最终的聚类质量.

由表 8～表 10 可见,当类簇需要分裂或者合并时,IRKM 在聚类质量上具有明显的优势,这是因为 IRKM 算法随时检查聚类中类簇簇内与簇间的结构变化,IRKM 不允许簇内出现两个距离较远的高密度区域,也不允许出现两个类簇间出现距离较近的高密度区域,因此,IRKM 算法可以根据数据分布合理地改变类簇的结构,从而提高聚类质量.

综上,IFKM 算法和 AIFC 算法所需运算时间较少,但多数情况下聚类结果差强人意,并不能得到合理的类簇个数;RKM 算法聚类在处理增量数据时所需时间最多,且聚类过程中不能改变类簇的个数,从而聚类质量较差;IRKM 算法在多数情况下,聚类过程所消耗时间大于 IFKM 和 AIFC 这两种增量聚类算法,但在划分新增类簇时综合考虑了边界数据点的邻域归属信息,使得新增数据的划分更准确. 此外,在增量聚类过程中始终关注类簇结构的变化,通过数据点的分布自适应调整类簇个数,从而得到较为合理的聚类结果.

4　结　论

针对增量聚类算法中类簇交叉边界区域不确定性信息的处理,在考虑边界区域新增数据点传统位置分布的基础上,进一步引入邻域归属信息,并关注类簇结构的变化,提出了一种考虑边界样本邻域归属信息的粗糙 K-means 增量聚类算法,不仅实现了新增数据点的合理划分,还可根据类簇结构的变化自适应分裂或合并类簇. 不同数据集的聚类结果对比分析,验证了算法的有效性. 本文研究更多地考虑了球形类簇的聚类分析,如何将所提出算法适用于不同数据类型的增量式聚类,并进一步降低算法的运算复杂度将是下一步研究工作的重点.

参考文献(References)

[1] Han J, Kamber M. Data mining, concepts and techniques[M]. The 3rd edition. San Francisco: Morgan Kaufmann Publishers, 2011: 15-22.

[2] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1): 48-61.
(Sun J G, Liu J, Zhao L Y. Clustering algorithms research[J]. Journal of Software, 2008, 19(1): 48-61.)

[3] 黄晓辉, 王成, 熊李艳, 等. 一种集成簇内和簇间距离的加权k-means聚类方法[J]. 计算机学报, 2019, 42(12): 2836-2848.
(Huang X H, Wang C, Xiong L Y, et al. A weighting k-means clustering approach by integrating intra-cluster and inter-cluster distances[J]. Chinese Journal of Computers, 2019, 42(12): 2836-2848.)

[4] 梁吉业, 白亮, 曹付元. 基于新的距离度量的K-modes聚类算法[J]. 计算机研究与发展, 2010, 47(10): 1749-1755.
(Liang J Y, Bai L, Cao F Y. K-modes clustering algorithm based on a new distance measure[J]. Journal of Computer Research and Development, 2010, 47(10): 1749-1755.)

[5] Cassisi C, Ferro A, Giugno R, et al. Enhancing density-based clustering: Parameter reduction and outlier detection[J]. Information Systems, 2013, 38(3): 317-330.

[6] Zhao Q P, Shi Y, Liu Q, et al. A grid-growing clustering algorithm for geo-spatial data[J]. Pattern Recognition Letters, 2015, 53: 77-84.

[7] Tzortzis G, Likas A. The minmax k-means clustering algorithm[J]. Pattern Recognition, 2014, 47(7): 2505-2516.

[8] Lei T, Jia X H, Zhang Y N, et al. Superpixel-based fast fuzzy C-means clustering for color image segmentation[J]. IEEE Transactions on Fuzzy Systems, 2019, 27(9): 1753-1766.

[9] Lingras P, West C. Interval set clustering of web users with rough K-means[J]. Journal of Intelligent Information Systems, 2004, 23(1): 5-16.

[10] Mitra S, Banka H, Pedrycz W. Rough-fuzzy collaborative clustering[J]. IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics, 2006, 36(4): 795-805.

[11] Peters G. Rough clustering utilizing the principle of indifference[J]. Information Sciences, 2014, 277: 358-374.

[12] 马福民, 逯瑞强, 张腾飞. 基于边界区域局部模糊增强的π RKM聚类算法[J]. 控制与决策, 2017, 32(11): 1949-1956.
(Ma F M, Lu R Q, Zhang T F. Improved π RKM clustering algorithm based on local fuzzy enhancement of boundary region[J]. Control and Decision, 2017, 32(11): 1949-1956.)

[13] Zhang T F, Ma F M, Yue D, et al. Interval type-2 fuzzy local enhancement based rough K-means clustering considering imbalanced clusters[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(9): 1925-1939.

[14] Khan I, Huang J Z, Ivanov K. Incremental density-based ensemble clustering over evolving data streams[J]. Neurocomputing, 2016, 191: 34-43.

[15] Zheng L W, Huo H, Guo Y Y, et al.Supervised adaptive incremental clustering for data stream of chunks[J]. Neuro Computing, 2017, 219(5): 502-517.

[16] Ester M, Kriegel H P, Sander J, et al. Incremental clustering for mining in a data warehousing environment[C]. International Conference on Very large Data Bases. Piscataway: IEEE, 1998: 323-333.

[17] 黄德才, 李晓畅. 基于相对密度的混合属性数据增量聚类算法[J]. 控制与决策, 2013, 28(6): 815-822.
(Huang D C, Li X C. Incremental relative density-based clustering algorithm for mixture data sets[J]. Control and Decision, 2013, 28(6): 815-822.)

[18] Bakr A M, Ghanem N M, Ismail M A. Efficient incremental density-based algorithm for clustering large datasets[J]. Alexandria Engineering Journal, 2015, 54(4): 1147-1154.

[19] Islam M N, Loo C K, Seera M. Incremental clustering-based facial feature tracking using Bayesian ART[J]. Neural Processing Letters, 2017, 45(3): 887-911.

[20] Tudu B, Ghosh S, Bag A K, et al. Incremental FCM technique for black tea quality evaluation using an electronic nose[J]. Fuzzy Information and Engineering, 2015, 7(3): 275-289.

[21] Wang L, Xu P P, Ma Q. Incremental fuzzy clustering of time series[J]. Fuzzy Sets and Systems, 2021, 421: 62-76.

[22] Laohakiat S, Saing V. An incremental density-based clustering framework using fuzzy local clustering[J]. Information Sciences, 2021, 547: 404-426.

[23] Cheng D D, Zhu Q S, Huang J L, et al. A novel cluster validity index based on local cores[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(4): 985-999.

作者简介

马福民(1979−), 女, 教授, 博士, 从事智能信息处理、智能生产系统等研究, E-mail: fmmatj@126.com;

孙静勇(1996−), 男, 硕士生, 从事信息处理与数据挖掘的研究, E-mail: 874562939@qq.com;

张腾飞(1980−), 男, 教授, 博士, 从事智能信息处理、大数据分析等研究, E-mail: tfzhang@126.com.

(责任编辑：郑晓蕾)