文章编号: 1001-0920(2017)11-1949-08 DOI: 10.13195/j.kzyjc.2016.1307

基于边界区域局部模糊增强的πRKM聚类算法

马福民[1†], 逯瑞强[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，南京 210023)

摘 要: 如何对交叉边界区域的数据对象进行度量与处理一直是粗糙k-means(RKM)及其衍生算法的主要出发点. πRKM算法通过引入Laplace无差别原则,较好地解决了传统RKM算法对权重系数的选择比较敏感等相关问题,但没有考虑边界区域多个类簇的交叉程度以及边界区域数据对象的空间位置分布对聚类结果的影响. 鉴于此,设计一种对边界区域的数据对象进行局部模糊度量的方法,并提出基于边界区域局部模糊增强的πRKM聚类改进算法,通过多组实例分析验证了所提出算法的有效性.

关键词: 粗糙聚类；k-means；局部模糊度量；粗糙集

中图分类号: TP18 **文献标志码:** A

Improved πRKM clustering algorithm based on local fuzzy enhancement of boundary region

MA Fu-min[1†], *LU Rui-qiang*[1], *ZHANG Teng-fei*[2]

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

Abstract: The primary starting point of rough k-means(RKM) and its derivatives is how to measure and process the data objects in the boundary regions. The traditional RKM algorithm is more sensitive to the choice of the weight coefficients of the upper and lower approximations, and the partitioning results are easy affected by the non-competitive objects in boundary region. By introducing the Laplace's principle of indifference for measuring the objects in boundary regions, the aforementioned problems of the traditional RKM algorithm are sovled well by using the πRKM algorithm. However, the overlapping degree in boundary regions and spatial distributions of different boundary objects are not considered. In order to better describe data objects in boundary regions, the local fuzzy measurement is introduced, and an improved πRKM clustering algorithm based on local fuzzy enhancement of boundary region is developed. The effectiveness of the proposed algorithm is demonstrated by experimental comparison and analysis.

Keywords: rough clustering；k-means；local fuzzy metric；rough sets

0 引 言

聚类分析将相似的数据对象聚集在相同的类簇,而将相异的数据对象划分到不同的类簇. k-means聚类算法是最典型的划分聚类分析方法之一,在众多研究领域均已得到了广泛的应用[1]. 为了解决类簇交叉边界数据对象的不确定性归属问题,Lingras[2]将粗糙集上、下近似的概念融入到k-means聚类算法中,经过几次完善,形成了较为经典的粗糙k-means(RKM)聚类方法[2-4],该算法将具有明确归属关系的数据对象划分到某一个类簇的下近似集,而将具有不确定性归属关系的数据对象划分到多个类簇的边界区域,这种对数据对象相对客观的描述方法,在很大程度上提高了k-means算法的精度,自提出以来受到越来越多的关注[5-20].

如何对边界区域的数据对象进行度量和处理是粗糙k-means及其衍生算法的主要出发点之一. Peters[11]讨论了边界区域离群数据点的处理,使用相对距离来代替粗糙k-means方法中的绝对距离作为相似性度量标准,有效地减小了离群点的影响. Mitra等[13-14]认为,粗糙k-means方法仅对下近似和边界

收稿日期: 2016-10-07；**修回日期:** 2016-12-31.

基金项目: 国家自然科学基金项目(61403184, 61105082)；江苏省高校自然科学研究重大项目(17KJA120001)；南京邮电大学1311人才计划基金项目(NY2013)；江苏高校优势学科建设工程项目；国家电子商务信息处理国际联合研究中心项目(2013B01035).

作者简介: 马福民(1979−), 女, 副教授, 博士, 从事智能信息处理、智能生产系统等研究；逯瑞强(1995−), 男, 硕士生, 从事信息处理与数据挖掘的研究.

†通讯作者. E-mail: fmmatj@126.com

区域分别赋予一个相对重要性权重系数,而没有对同一区域内不同数据对象归属于类簇的程度进行区分度量,鉴于此,Mitra等[13]提出了粗糙模糊k-means(RFKM)聚类方法,以隶属函数的形式对近似区域中的不同数据对象进行加权度量. 不同于文献[13]的粗糙模糊k-means聚类方法,Maji等[15]认为在下近似集中的所有数据对象因为明确属于该类簇,在计算聚类中心时应具有相同的权重并且不受其他类簇的影响,由此提出了一种仅将边界区域进行模糊化处理的粗糙模糊k-means聚类方法.

Peters[16]进一步分析了现有RKM算法存在的一些问题,认为传统算法对下近似及边界区域相对权重系数的选择是非常敏感的,鲁棒性不够好,在中心均值的迭代计算公式中,边界区域的数据对象对其所属类簇的贡献是非竞争性的,导致这些对象对聚类结果的影响会随着其所属类簇数量的增加而增加. 周杨等[17]指出,固定的权重系数忽略了数据分布的差异性. 张腾飞等[18]分析了类簇内数据不平衡分布对聚类结果的影响. 针对上述问题,Peters[16]从交叉边界区域数据对象隶属于不同类簇个数的角度,结合Laplace的无差别原理,设计了一种基于无差别原理的π粗糙k-means(πRKM)聚类方法.

πRKM算法最突出的特点是将交叉边界的数据对象在没有其他先验知识的情况下,无差别地平均参与到可能归属的类簇中心均值的迭代计算. 另外,该算法无需人为设置边界区域的相对权重系数,具有坚实的理论基础,鲁棒性较好. 然而,πRKM算法忽略了边界交叉区域数据对象的不同空间位置分布,因为同一个边界区域中的数据对象归属于不同类簇的程度在很多情况下是不同的[14-19],尤其是当边界区域随着阈值参数的设置增大时,边界区域中的数据对象之间的差异性也会随之增大. 所以,在类簇中心均值的迭代计算中,同一边界区域中数据对象的加权系数不应该简单平均,为了更好地体现这些数据对象的空间位置分布,加权系数应该有所区分.

本文在πRKM算法的基础上,对边界交叉区域的数据对象进行局部模糊度量,给出一种基于边界区域局部模糊增强的πRKM聚类改进算法(为方便,记为BF-πRKM). BF-πRKM算法在保留了πRKM算法优点的同时,还具有以下特点:

1) 聚类结果具有较强的描述性,下近似集中的数据对象具有明确的归属关系,边界集中的数据对象可以根据其空间位置的分布给出可能归属于哪些类簇及其隶属程度;

2) 在类簇中心均值的迭代计算中,边界区域中所有数据对象的全局权重之和均为1,而且进一步考虑了每一个数据对象的空间位置差异化分布;

3) 边界区域数据对象参与可能归属类簇的中心均值迭代计算的权重系数是由其自身到各个类簇的相对位置决定的,无需增加其他需人为经验调节的参数,算法具有较强的自适应性.

1 粗糙k-means聚类相关算法

1.1 Lingras粗糙k-means算法

在RKM算法[4]设计中,聚类对象的处理具有以下特征:

1) 聚类对象最多只能确定地属于某一个类簇的下近似集;

2) 聚类对象若不能确定地属于任一个类簇的下近似集,则同时属于两个或两个以上类簇的边界集;

3) 每个类簇由下近似集和边界集两部分组成,下近似集和边界集的并集构成类簇的上近似集.

RKM算法的实施步骤如下.

Step 1: 初始化. 需要划分的类簇个数k;每个簇的初始中心$v_i,\ i=1,2,\cdots,k$;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(在后续的很多算法中大多采用相对距离阈值ξ);最大迭代次数$I_{\max}$.

Step 2: 数据对象到各类簇的划分. 根据每个数据对象$X_j(j=1,2,\cdots,N)$到各类簇中心的距离,将其划分到对应类簇的下近似集$\underline{C_i}$或边界集$\widehat{C_i}$中.

Step 3: 类簇中心点的迭代计算. 根据式(1)计算各个类簇新的中心点

$$v_i=\begin{cases} w_l\times\displaystyle\sum_{X_j\in\underline{C_i}}\frac{X_j}{|\underline{C_i}|}+w_b\times\sum_{X_j\in\widehat{C_i}}\frac{X_j}{|\widehat{C_i}|}, \\ \quad \underline{C_i}\neq\varnothing\bigwedge\widehat{C_i}\neq\varnothing; \\ \displaystyle\sum_{X_j\in\underline{C_i}}\frac{X_j}{|\underline{C_i}|},\ \underline{C_i}\neq\varnothing\bigwedge\widehat{C_i}=\varnothing; \\ \displaystyle\sum_{X_j\in\widehat{C_i}}\frac{X_j}{|\widehat{C_i}|},\ \underline{C_i}=\varnothing\bigwedge\widehat{C_i}\neq\varnothing. \end{cases}\tag{1}$$

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 引入模糊度量的RFKM算法

在引入模糊度量的粗糙模糊k-means(RFKM)算法中,基于不同的数据对象到各个类簇的距离计算其模糊隶属程度,是区别对象间差异性的有效工具. 隶属度越大,表明数据对象与该类簇的关系越紧密,在类簇中心的迭代计算中应该赋予更大的权重系数.

模糊隶属度量的计算公式为

$$\mu_{ij}=\frac{1}{\sum\limits_{z=1}^{k}\left(\frac{d_{ij}}{d_{zj}}\right)^{\frac{2}{m-1}}}. \tag{2}$$

其中: $m\in(1,\infty)$ 为隶属度量的模糊化指数, μ_{ij} 为数据对象 X_j 到类簇 C_i 的隶属度权值, d_{ij} 为 X_j 到类簇 C_i 中心点 v_i 的距离, d_{zj} 为 X_j 到类簇 C_z 中心点 v_z 的距离, k 为需要划分的类簇个数.

传统的粗糙模糊 k-means 算法是在 RKM 的基础上, 引入模糊隶属度权值作为数据对象差异度的判断标准, 其中心迭代公式[13] 为

$$v_i=\begin{cases} w_l\times\dfrac{\sum\limits_{X_j\in\underline{C_i}}(\mu_{ij})^m X_j}{\sum\limits_{X_j\in\underline{C_i}}(\mu_{ij})^m}+w_b\times\dfrac{\sum\limits_{X_j\in\widehat{C_i}}(\mu_{ij})^m X_j}{\sum\limits_{X_j\in\widehat{C_i}}(\mu_{ij})^m}, \\ \quad \underline{C_i}\neq\varnothing\bigwedge\widehat{C_i}\neq\varnothing; \\ \dfrac{\sum\limits_{X_j\in\underline{C_i}}(\mu_{ij})^m X_j}{\sum\limits_{X_j\in\underline{C_i}}(\mu_{ij})^m},\ \underline{C_i}\neq\varnothing\bigwedge\widehat{C_i}=\varnothing; \\ \dfrac{\sum\limits_{X_j\in\widehat{C_i}}(\mu_{ij})^m X_j}{\sum\limits_{X_j\in\widehat{C_i}}(\mu_{ij})^m},\ \underline{C_i}=\varnothing\bigwedge\widehat{C_i}\neq\varnothing. \end{cases} \tag{3}$$

在粗糙模糊 k-means 算法中, 对模糊隶属度权值进行了修改, 认为下近似集中的数据对象是明确归属于该类簇的, 将所有下近似集的数据对象的隶属度均赋值为 1, 其模糊隶属度公式[15] 为

$$\mu_{ij}=\begin{cases} \dfrac{1}{\sum\limits_{z=1}^{k}\left(\frac{d_{ij}}{d_{zj}}\right)^{\frac{2}{m-1}}},\ X_j\in\widehat{C_i}; \\ 1,\ X_j\in\underline{C_i}. \end{cases} \tag{4}$$

1.3　Peters 引入无差别原则的 πRKM 算法

Peters[16] 进一步分析了边界区域数据对象在类簇中心的计算中所产生的量化影响. 如图 1 所示, 包含 3 个类簇的数据集被划分为 7 个区域, R_1、R_2、R_3 分别代表类簇 C_1、C_2、C_3 的下近似集, 其他区域为各个类簇的交叉边界集, 比如 R_{12} 代表其中的数据对象既可能属于 C_1 也可能属于 C_2. 在传统的 RKM 算法中, 该区域的数据对象将同时重复参与到 C_1 和 C_2 的中心迭代计算, 同样地, R_{123} 中的数据对象虽然具有更加不确定的归属关系, 但仍将以同样的权重系数同时重复参与到 3 个类簇的中心迭代计算, 这便导致了争议性越大的数据对象所造成的整体影响往往越大, 在算法的迭代计算中, 类簇的中心点也越容易向边界交叉重叠更严重的区域偏移.

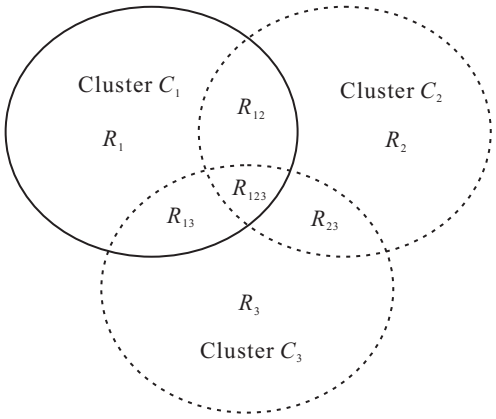


图 1　3 个类簇的区域集合划分

为解决上述问题, Peters 将无差别原则运用到 RKM 算法对边界区域数据对象的处理中, 提出了 πRKM 算法. 该算法认为, 对于每个位于边界集中的数据对象 X_j, 如图 1 中位于 R_{123} 的数据对象, 应该等概率地属于 C_1、C_2、C_3, 从而也应该无差别地平均参与到这 3 个类簇中心均值的迭代计算.

πRKM 算法引入集合 $\{B_{X_j}\}$ 表示其可能归属的类簇, $|B_{X_j}|$ 表示其所处的边界集个数, 在边界集中的数据对象参与中心均值的迭代计算时, 引入一个平均加权系数 $1/|B_{X_j}|$. 如在图 1 中, 假定 $X_1\in R_{12}$, 则有 $\{B_{X_1}\}=\{R_1,R_2\}$, $|B_{X_1}|=2$. πRKM 算法的类簇中心迭代公式为

$$v_i=\frac{\sum\limits_{X_j\in\underline{C_i}}X_j+\sum\limits_{X_j\in\widehat{C_i}}\frac{X_j}{|B_{X_j}|}}{|\underline{C_i}|+\sum\limits_{X_j\in\widehat{C_i}}\frac{1}{|B_{X_j}|}}. \tag{5}$$

这种中心均值计算方法确保每一个数据样本参与运算的权重之和恒定为 1, 减弱了争议性较大的数据对象所带来的不均衡整体影响.

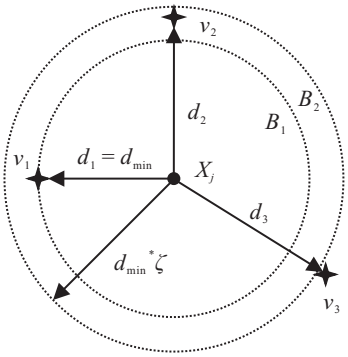
2　基于边界区域局部模糊增强的 πRKM 算法

2.1　边界区域局部模糊度量

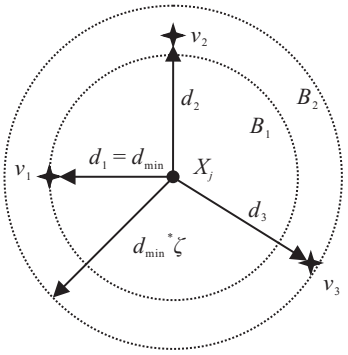
Peters 的 πRKM 算法虽然减弱了争议性的边界数据对全局的影响, 然而对同一边界近似区域内的数据对象仍赋予了同样的权重, 忽视了同一边界区域内部不同数据对象客观存在的不同空间分布, 尤其当边界区域较大时, 这种差异性越发明显.

如图 2 所示, v_1、v_2 和 v_3 是距离数据对象 X_j 较近的 3 个类簇 C_1、C_2 和 C_3 的中心点, 其距离分别为 d_1、d_2 和 d_3, 其中 $d_1<d_2<d_3$, 记最小距离 $d_{\min}=d_1$. 为

了更直观地区分不同的距离差异,引入相对距离阈值参数ξ,并以$d_{\min}$和$d_{\min}\xi$为半径分别作两个虚线圆B_1和B_2. 对比图2(a)和图2(b)可以看出,交叉边界的覆盖区域将随着相对距离阈值ξ的增大而变大.



(a) 相对距离阈值ζ=1.2



(b) 相对距离阈值ζ=1.4

图 2 边界区域随相对距离阈值的变化

由图2(a)可见,当$\xi=1.2$时,$d_2<d_1\xi$,数据对象X_j归属于C_1和C_2两个类簇的边界集,到两个类簇C_1和C_2间的距离差为$d_2-d_1<d_1(\xi-1)=0.2d_1$. 由于数据对象$X_j$距离两个类簇中心的距离相差不是特别大,按照无差别原理,将数据对象X_j平均参与到这两个类簇的中心均值迭代计算还算比较合理.

然而,当相对距离阈值参数ξ进一步增大时,例如取$\xi=1.4$,如图2(b)所示,由于$d_1\xi$变大,$d_3<d_1\xi$,此时数据对象X_j同时归属于类簇C_1、C_2和C_3的边界集,但X_j相对于类簇C_1和C_3中心点的距离差是比较大的,X_j归属于类簇C_1的概率明显应该大于其归属于C_3的概率. 因此,在参与各个交叉类簇的中心均值迭代计算时,数据对象X_j相对类簇C_1的贡献应该大于对C_3的贡献,相应地,其权重系数应该有所区分,而不是简单地进行平均.

为了更好地描述数据对象对交叉类簇的不同归属程度,本文在边界区域引入局部模糊度量方法,仅考虑边界数据对象相对于所在交叉类簇的隶属度,局部模糊度量为

$$\mu_{ij}=\begin{cases}\dfrac{1}{\displaystyle\sum_{C_z\in\{B_{X_j}\}}\left(\dfrac{d_{ij}}{d_{zj}}\right)^{\frac{2}{m-1}}},\ X_j\in\widehat{C_i};\\ 1,\ X_j\in\underline{C_i};\\ 0,\ X_j\notin\overline{C_i}.\end{cases}\tag{6}$$

其中:μ_{ij}为数据对象X_j到类簇C_i的隶属度权值;d_{zj}为X_j到交叉类簇C_z中心点v_z的距离;$\overline{C_i}$为类簇C_i的上近似集,是其边界集与下近似集的合集.

式(6)与(4)的最大区别在于,只考虑数据对象X_j相对于所在交叉类簇的局部隶属度量,对于明确没有归属关系的类簇,相应的隶属度为0,在算法的迭代计算过程中也无需再对其进行计算;而与πRKM算法中的平均加权系数相比,进一步考虑了交叉边界区域中数据对象的不同位置分布.

2.2 BF-πRKM算法设计

在保留πRKM算法优点的基础上,进一步将边界区域的局部模糊度量引入类簇中心均值的迭代计算,有

$$v_i=\frac{\displaystyle\sum_{X_j\in\underline{C_i}}X_j+\sum_{X_j\in\widehat{C_i}}\mu_{ij}X_j}{|\underline{C_i}|+\displaystyle\sum_{X_j\in\widehat{C_i}}\mu_{ij}},\tag{7}$$

其中隶属度权值μ_{ij}由式(6)计算得到. 可以看出,式(7)根据边界集中的数据对象所可能归属的类簇个数以及相对这些类簇的归属程度,有差别地参与到这些类簇的中心均值迭代计算. 当相对距离阈值ξ取值较小时,边界区域的数据对象到所在交叉类簇的距离相差不大,其模糊隶属度也接近于πRKM算法的平均系数. 但当参数ξ取值较大时,边界区域的数据对象距离所在交叉类簇的隶属程度有可能相差也较大. 由于更进一步考虑了这些数据对象的空间分布,在πRKM算法的基础上,可以对归属程度更大的类簇进行局部模糊增强,更利于类簇中心向合理的位置快速移动. 另一方面,BF-πRKM与πRKM算法相同,每一个数据对象在每一步参与到所在交叉类簇中心点迭代运算的权重之和均为1,但边界区域中的数据对象参与到每一个所在交叉类簇的权重体现出了不同的位置分布,算法的描述更为合理.

基于边界区域局部模糊增强的BF-πRKM算法的执行步骤如下.

输入:数据集U:$U=\{X_j|j=1,2,\cdots,N\}$;

输出:将数据对象集合U划分为k个类簇.

聚类算法步骤.

Step 1:设置与初始化参数. k为类簇的数量,即聚

类划分的类簇个数; v_i 为每个类簇的初始中心点的选取; $I_{\max}$ 为算法的最大迭代次数; ξ 为相对距离判断阈值; m 为局部模糊度量的模糊化指数.

Step 2: 对于每个数据对象 X_j, 计算其到各类簇中心的欧氏距离, 并记最小距离为 $d_{\min}$. 假设对应的类簇为 C_i, 若不存在其他类簇 C_z 使得 $d_i < d_{\min}\xi$, 则将 X_j 归到类簇 C_i 的下近似集, 否则, 将 X_j 归到所有使得 $d_i < d_{\min}\xi$ 的各类簇的边界集.

Step 3: 由式(6), 对每个边界区域的数据对象, 计算其到所在交叉类簇的隶属度权值.

Step 4: 根据中心点迭代式(7)重新更新计算各个类簇的中心点.

Step 5: 若各类簇中心不再发生变化或已达到设定的最大迭代次数, 则算法终止, 否则返回 Step 2, 重新进行迭代计算.

3　实例仿真分析

为了验证算法的有效性, 采用人工和UCI标准数据集对算法进行测试, 并与RKM算法、RFKM算法、πRKM算法的实验结果在聚类精度和DBI指标方面进行对比分析.

引用文献[16]的评判标准, 从以下4个方面对聚类结果进行评判:

1) OK: 代表那些位于类簇下近似集中并且聚类正确的对象数(越大越好).

2) ¬OK: 代表那些位于类簇下近似集中但是聚类错误的对象数(越小越好).

3) πOK: 代表那些位于类簇下近似集中并且聚类正确的对象数加上类簇边界集中聚类正确的数据对象数乘以它们的重叠系数(越大越好).

4) Bd: 代表位于边界集中的数据对象数(属于不确定的对象, 即边界集中的对象, 在设置相同的阈值 ξ 下, 希望它越小越好).

3.1　人工数据集测试分析

随机生成按照正态分布的5个部分有严重交叉重叠的类簇, 每一类包含40个数据对象, 以不同符号加以区分, 如图3所示.

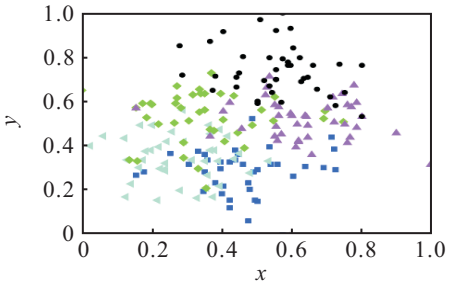


图 3　人工数据集类簇分布

为便于不同算法的对比分析, 实验过程中所有算法均使用相同的初始类簇中心点. 对于RKM和RFKM算法, 根据经验测试的最好结果, 将下近似和边界区域的相对权重系数设定为0.7和0.3; RFKM算法与本文BF-πRKM算法的模糊化指数 m 取值为1.5; 为了测试相对距离判断阈值 ξ 对不同算法的影响, 参数 ξ 分别取值为1.2,1.4,1.6. 按照聚类结果的4个评判指标, 记录各算法在参数 ξ 取值不同时的实验结果, 如表1所示.

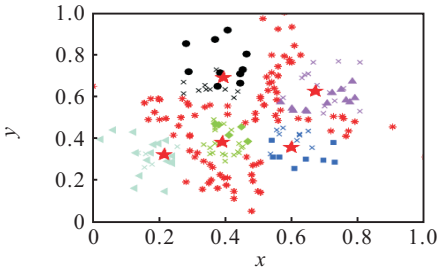
表 1　人工数据集分类精度对比

ξ	RKM				RFKM				πRKM				BF-πRKM			
	OK	πOK	¬OK	Bd	OK	πOK	¬OK	Bd	OK	πOK	¬OK	Bd	OK	πOK	¬OK	Bd
1.2	96	103.50	89	15	94	103.33	87	19	92	96	89	19	94	103	88	18
1.4	69	90.08	87	44	67	85.33	94	39	116	133.17	48	36	117	134.17	47	36
1.6	51	87	67	82	55	84.17	83	62	58	96	56	86	103	131.45	39	58

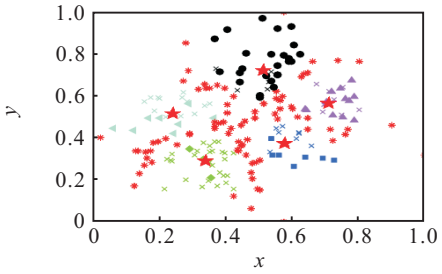
由表1的实验结果可见, 在相对距离判断阈值参数 ξ 取值较小时, 4种算法的实验结果类似, 这是因为边界集的数量较少、差异度不大. 当参数 ξ 取值稍大时, πRKM算法与BF-πRKM算法的优势得以体现, 获得了相对较好的聚类效果. 当参数 ξ 继续增大时, 边界区域的数据对象进一步增多、差异度增大, 此时, 采取简单平均加权系数的πRKM算法的劣势体现出来, 聚类效果变差, 而BF-πRKM算法仍然可以保持较高的聚类精度, 具有更强的适应性.

为了更直观地进行对比分析, 图4给出了这4种算法在阈值参数 ξ 取1.6时的聚类结果. 图4中, 五角星代表各个类簇的中心点, 圆圈代表落入交叉边界集的数据对象, 叉号代表被错误划分到其他类簇下近似集的数据对象.

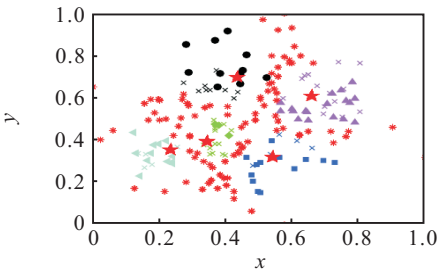
比较图4中4种不同算法的聚类结果可以看出, BF-πRKM算法最终得到的类簇中心点位置更为合理, 被错误划分的数据对象最少, 类簇更加紧凑. 图4(d)用虚线圆标注了与πRKM算法聚类结果相比有



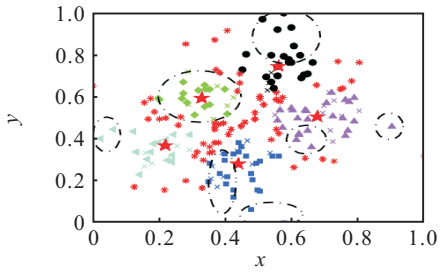
(a) RKM算法的聚类结果



(b) RFKM算法的聚类结果



(c) πRKM算法的聚类结果



(d) BF-πRKM算法的聚类结果

图 4 4种不同算法对类簇的聚类结果对比

区别的部分数据对象,可以看出,在阈值参数ξ取值较大时,BF-πRKM算法具有明显的优势.

3.2 UCI数据集测试分析

选取5个类簇交叉的程度依次递增的标准UCI数据集Iris、Wine、Breast Cancer(Prognostic)、Ionosphere和Fertility进行测试分析,仍然采用上述4种算法对这些数据集进行测试,实验过程中所有数据集在参与计算时都进行归一化处理. 所有算法针对同一个数据集使用相同的初始类簇中心点. 对于RKM算法和RFKM算法所需要的下近似和边界区域的相对权重系数、RFKM算法和BF-πRKM算法所需要的模糊化指数,根据经验测试的最好结果,针对不同数据集的设置如表2所示.

表 2 聚类算法的参数设置

数据集	相关参数			
	k	w_l	w_b	m
Iris	3	0.9	0.1	2
Wine	3	0.9	0.1	1.5
Breast Cancer	6	0.8	0.2	2.5
Ionosphere	2	0.9	0.1	2
Fertility	2	0.7	0.3	2

相对距离阈值参数ξ仍然分别取值为1.2, 1.4, 1.6. 按照聚类结果的4个评判指标,记录各算法针对不同数据集在参数ξ取值不同时的实验结果,如表3所示.

表3中,针对Ionosphere和Fertility存在部分实验结果缺失是由于这两个数据集的类簇交叉较为严重,当ξ取值较大时,边界区域的不确定数据对象过多,而下近似区域中的数据对象过少,造成部分类簇中心点重叠在一起,无法给出有意义的聚类结果. 这从

表 3 不同数据集的分类精度对比

数据集	ξ	RKM				RFKM				πRKM				BF-πRKM			
		OK	πOK	¬OK	Bd	OK	πOK	¬OK	Bd	OK	πOK	¬OK	Bd	OK	πOK	¬OK	Bd
Iris	1.2	123	130.50	12	15	124	133	8	18	126	132.50	11	13	126	132.50	11	13
	1.4	120	132	6	24	120	132	6	24	124	134	6	20	124	134	6	20
	1.6	118	132.50	3	29	118	132.50	3	29	122	134.50	3	25	121	134	3	26
Wine	1.2	145	159.33	4	29	145	159.33	4	29	151	162.83	3	24	153	164	3	22
	1.4	134	153	1	43	132	152.33	1	45	136	154.83	1	41	136	154.83	1	41
	1.6	116	141.83	1	61	117	143.17	1	60	100	134.33	1	77	113	141.67	1	64
Breast Cancer	1.2	490	515.50	28	51	497	518.50	29	43	502	522	27	40	502	522	27	40
	1.4	451	496	28	90	454	498.50	26	89	468	509.50	18	83	469	510	18	82
	1.6	412	479	134	23	416	480.50	24	129	404	483.50	6	159	427	494.50	7	135
Ionosphere	1.2	191	220.50	101	59	191	220.50	101	59	201	230.50	91	59	202	231	91	58
	1.4	163	209	96	92	164	210	95	92	156	227	53	142	166	228.50	60	125
	1.6	–	–	–	–	–	–	–	–	–	–	–	–	134	234.5	4	201
Fertility	1.2	27	32	63	10	28	33	62	10	28	50.50	27	45	38	52.50	33	29
	1.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	1.6	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

某种程度上也说明了采取多组阈值参数对某些数据集进行对比分析的必要性.

从上述实验结果可以看出,对于类簇交叉重叠度不是特别高的Iris和Wine数据集,4种算法的聚类结果相近,但随着数据集类簇交叉重叠复杂度越来越高,πRKM算法与BF-πRKM算法的优势越来越明显,这是由于πRKM算法与BF-πRKM算法在边界区域中引入了竞争机制,减弱了边界区域的数据对象对聚类结果的全局影响. 更进一步,由于BF-πRKM算法在边界区域引入了局部模糊度量,较好地考虑了边界区域数据对象的差异性,算法具有更强的自适应性. 因此,随着阈值参数ξ的增大,BF-πRKM算法获得了更好的聚类效果,尤其是针对类簇交叉较为严重的Ionosphere和Fertility数据集,BF-πRKM算法的优越性更为明显.

3.3　DBI指标测试分析

由于聚类分析是一种无监督的分类算法,很多情况下,事先并不知道数据对象的决策标签,为了更好地对算法进行评估,进一步采用聚类分析中常用的DBI指数分析方法测试算法的聚类质量.

DBI指数的计算方法为

$$\mathrm{DBI}=\frac{1}{k}\sum_{z=1}^{k}\max_{z\neq i}\left\{\frac{S(C_z)+S(C_i)}{d_{(C_z,C_i)}}\right\},\qquad(8)$$

其中$S(C_i)$为类簇C_i的簇内离散度,其值越低,说明其簇内部越紧凑. 为了便于比较,采用如下计算公式[16]:

$$S(C_i)=\frac{\displaystyle\sum_{X_j\in\overline{C_i}}\frac{\|X_j-v_i\|}{|B_{X_j}|}}{\displaystyle\sum_{X_j\in\overline{C_i}}\frac{1}{|B_{X_j}|}}.\qquad(9)$$

DBI指数的目标是在最小化各个簇内离散度$S(C_i)$的同时,最大化类簇之间的距离$d(C_z,C_i)$,其值越小,代表聚类效果越佳.

由于πRKM算法、BF-πRKM算法与RKM算法和RFKM算法相比具有不同的算法结构,这里仅对πRKM算法和BF-πRKM算法进行DBI指数的对比分析,结果如表4所示.

由表4可见,无论是人为数据集或是UCI标准数据集,相比较于πRKM算法,在不同的比例阈值ξ和不同的测试数据集下,BF-πRKM算法的DBI指数均低于πRKM的DBI指数,类簇内更加紧凑,聚类效果更为出色.

综合上述聚类结果和DBI指数分析可以看出,对于边界交叉不是很严重的数据集,当阈值参数ξ取值不大时,BF-πRKM算法与πRKM算法的结果相当,稍优于传统的RKM算法与RFKM算法. 当阈值参数ξ取值较大时,BF-πRKM算法相比πRKM算法具有更好的自适应性,对于边界交叉比较严重的数据集,BF-πRKM算法相比πRKM算法具有更好的聚类效果.

表 4　不同算法的DB指标测试结果对比

数据集	ξ	πRKM	BF-πRKM
人工数据集	1.2	0.113 6	0.117 1
	1.4	0.129 5	0.122 6
	1.6	0.292 3	0.154 5
Iris	1.2	0.196 1	0.159 4
	1.4	0.205 4	0.202 7
	1.6	0.212 6	0.208 9
Wine	1.2	0.781 0	0.771 4
	1.4	0.866 5	0.833 5
	1.6	1.267 1	0.965 9
Breast Cancer	1.2	0.867 2	0.866 4
	1.4	0.930 7	0.914 8
	1.6	1.267 1	1.038 8
Ionosphere	1.2	2.738 9	2.707 1
	1.4	4.246 1	3.471 3
	1.6	–	4.847 4
Fertility	1.2	2.501 8	2.496 4
	1.4	–	–
	1.6	–	–

4　结　论

聚类分析边界区域的存在是由于信息的缺失或遗漏导致部分数据对象暂时不可分,如何对这些边界区域的数据对象进行度量和处理已成为聚类分析结果的重要影响因素. 本文在保留Peters的πRKM算法优点的基础上,通过引入局部模糊度量的方法,对边界区域内数据对象相对于所在交叉类簇的不同空间位置分布进一步进行描述和度量,提高了算法的自适应性. 通过实验对比分析,验证了算法的可行性,尤其是在类簇交叉较为严重的数据聚类分析应用领域具有较为明显的优势. 如何针对不同的应用需求,进一步分析相对距离阈值参数的设置对聚类结果的影响,将是下一步的研究工作.

参考文献(References)

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

[2] Lingras P. Rough set clustering for web mining[C]. IEEE Int Conf on Fuzzy Systems. Honolulu: HI, 2002: 1039-1044.

[3] Lingras P, Yan R, West C. Comparison of conventional and rough K-means clustering[J]. Lecture Notes in Artificial Intelligence, 2003, 2639(1): 130-137.

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

[5] Lingras P, Peters G. Rough clustering[J]. Data Mining and Knowledge Discovery, 2011, 1(1): 64-72.

[6] Lingras P, Peters G. Rough sets: Selected methods and applications in management and engineering[C]. Applying Rough Set Concepts to Clustering. London: Springer, 2012: 23-37.

[7] Peters G, Crespo F, Lingras P. Soft clustering-fuzzy and rough approaches and their extensions and derivatives[J]. Int J of Approximate Reasoning, 2013, 54(2): 307-322.

[8] Saha I, Sarkar J P, Maulik U. Ensemble based rough fuzzy clustering for categorical data[J]. Knowledge-Based Systems, 2015, 77(3): 114-127.

[9] Lai J Z C, Juan E Y T, Lai F J C. Rough clustering using generalized fuzzy clustering algorithm[J]. Pattern Recognition, 2013, 46(9): 2538-2547.

[10] Shi J, Lei Y, Zhou Y, et al. Enhanced rough-fuzzy c-means algorithm with strict rough sets properties[J]. Applied Soft Computing, 2016, 46(9): 827-850.

[11] Peters G. Outliers in rough k-means clustering[J]. Lecture Notes in Computer Science, 2005, 3776(11): 702-707.

[12] Hu Q, Yu D. An improved clustering algorithm for information granulation[J]. Lecture Notes in Computer Science, 2005, 3613(8): 494-504.

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

[14] Mitra S, Banka H. Application of rough sets in pattern recognition[J]. Trans on Rough Sets, 2007, 7(1): 151-169.

[15] Maji P, Pal S K. RFCM: A hybrid clustering algorithm using rough and fuzzy sets[J]. Fundamenta Informaticae, 2007, 80(4): 475-496.

[16] Peters G. Rough clustering utilizing the principle of indifference[J]. Information Science, 2014, 277(2): 358-374.

[17] 周杨, 苗夺谦, 岳晓冬. 基于自适应权重的粗糙K-均值聚类算法[J]. 计算机科学, 2011, 38(6): 237-241.
(Zhou Y, Miao D Q, Yue X D. Rough K-means clustering based on self-adaptive weights[J]. Computer Science, 2011, 38(6): 237-241.)

[18] 张腾飞, 陈龙, 李云. 基于簇内不平衡度量的粗糙K-means聚类算法[J]. 控制与决策, 2013, 28(10): 1479-1484.
(Zhang T F, Chen L, Li Y. Rough K-means clustering based on unbalanced degree of cluster[J]. Control and Decision, 2013, 28(10): 1479-1484.)

[19] Zhang T F, Chen L, Ma F M. A modified rough c-means clustering algorithm based on hybrid imbalanced measure of distance and density[J]. Int J of Approximate Reasoning, 2014, 55(8): 1805-1818.

[20] Peters G. Is there any need for rough clustering?[J]. Pattern Recognition Letters, 2014, 53(2): 31-37.

（责任编辑：郑晓蕾）

下 期 要 目