基于空间距离自适应权重度量的粗糙 K-means 算法

王慧研[1] 张腾飞[1] 马福民[2]
(南京邮电大学自动化学院 南京 210023)[1] (南京财经大学信息工程学院 南京 210023)[2]

摘 要 粗糙 K-means 算法中下近似和边界区域权重系数的设置对算法的聚类效果有着重要的影响。传统的粗糙 K-means 算法及很多改进的粗糙 K-means 算法对所有类簇的下近似和边界区域设置固定的权重,忽视了簇内数据对象分布差异性的影响。针对这个问题,根据下近似和边界区域的数据对象相对于类簇中心的空间分布情况,提出一种新的基于空间距离自适应权重度量的粗糙 K-means 算法。该算法在每次迭代过程中,根据每个类簇的下近似和边界区域的数据对象相对于类簇中心的平均距离,综合度量下近似和边界区域对于类簇中心迭代计算的不同重要程度,动态地计算下近似和边界区域的相对权重系数。通过实例验证及实验仿真证明了所提算法的有效性。

关键词 粗糙 K-means,粗糙集,聚类算法,自适应权重
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.07.033

Rough K-means Algorithm with Self-adaptive Weights Measurement Based on Space Distance

WANG Hui-yan[1] ZHANG Teng-fei[1] MA Fu-min[2]
(College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)[1]
(College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China)[2]

Abstract The setting of weights coefficient of lower approximation and boundary area in rough K-means algorithm has an important influence on final clustering results of algorithm. However, traditional rough K-means and many refined rough K-means algorithms set up fixed weights of lower approximations and boundary area for all clusters, ignoring the effect of distribution difference of data objects within clusters. To cope with this problem, a new rough K-means algorithm with self-adaptive weights measurement based on space distance was proposed according to the spatial distribution of objects in lower approximation and boundary area relative to the cluster centers. During each iteration process, different importance of lower approximation and boundary area on iterative computation of cluster centers was measured based on average distance of objects in lower approximation and boundary area relative to cluster centers and the relative weights coefficient of lower approximation and boundary area were dynamically calculated. The validity of the algorithm was verified by experimental analysis.

Keywords Rough k-means, Rough set, Clustering algorithm, Self-adaptive weight

1 引言

聚类分析是数据挖掘[1]领域十分重要的研究内容。聚类可以理解为根据相似度的标准将对象的集合划分为若干个簇的过程,其中将相似度较高的对象放在同一簇中,而将差异度较大的对象划分到不同的簇中。目前,常见的聚类算法有:基于密度的聚类、基于划分的聚类、基于层次的聚类、基于模型的聚类以及基于网格的聚类。K-均值(K-means)是其中应用最为广泛的划分聚类方法之一[2-4]。

由于经典 K-means 算法将每个对象确定地划分到某个簇,而在实际应用中,簇与簇之间往往存在交叉重叠的现象,针对不确定性数据对象的归属问题,学者们相继提出了许多软聚类算法。Bezdek 将模糊集理论引入 K-means,提出了模糊 K-均值(Fuzzy K-means,FKM)算法[5-6],其使用隶属度函数来衡量对象到类簇的隶属程度。FKM 算法是对经典 K-means 算法的改进,获得了广泛的关注[7-8]。然而,模糊隶属度函数的值并不总是与对象的隶属程度相符,而且容易受到噪声点的影响[9-10]。针对 FKM 存在的问题,Krishnapuram 和 Keller 提出了可能性 K-均值(Possibilistic K-means,PKM)算法[9-10],但 PKM 算法的缺陷是容易产生一致性聚类问题[11]。

Lingras 等[12]将粗糙集[13]与 K-means 算法相结合,提出

到稿日期:2017-05-18 返修日期:2017-07-11 本文受国家自然科学基金项目(61403184),江苏省高校自然科学研究重大项目(17KJA120001),江苏省"青蓝工程"基金(QL2016),南京邮电大学"1311人才计划"基金(NY2013),南京邮电大学科研项目基金(NY215149)资助。
王慧研(1990-),女,硕士生,主要研究方向为粗糙聚类算法;张腾飞(1980-),男,博士,教授,硕士生导师,主要研究方向为智能信息处理、智能控制等,E-mail:tfzhang@126.com(通信作者);马福民(1979-),女,博士,副教授,硕士生导师,主要研究方向为智能信息处理、智能生产系统等。

了粗糙 K-均值(Rough K-means，RKM)算法。该算法根据对象的位置分布将对象划分到各个类簇的下近似或交叉边界区域，下近似中的对象确定属于该簇，边界区域中的对象可能属于该簇，下近似和边界区域构成了簇的上近似。RKM 算法由于较为客观地描述数据对象，获得了广泛的关注，而经典的 RKM 算法无法处理模糊性信息，因此近些年很多学者相继提出了系列的改进算法。文献[14-17]将粗糙聚类和模糊度量相结合，给出了粗糙模糊 K-均值(Rough-Fuzzy K-means，RFKM)算法。文献[18]考虑到簇内对象空间分布的差异性对聚类结果的影响，提出了一种基于距离和密度混合度量的粗糙 K-means 聚类算法。

粗糙 K-means 算法中，权重系数的设置影响着最终的聚类效果。针对如何选取合适的权重系数，学者们提出了许多方法。Lingras 等[12]的粗糙 K-means 及其很多变体都采用固定的下近似和边界权重，其主要依赖于经验或者多次实验来选取使结果最优的权重。文献[19]提出了基于精确度的自适应权重，根据下近似和上近似的对象个数计算权重。文献[20]引进粗糙熵[21]，在聚类过程中根据粗糙熵评估选取的权重参数。文献[22]提出了基于无差异原则的 π 粗糙 K-均值(π Rough K-means，πRKM)算法，给出了不含下近似和边界区域权重参数的均值公式。文献[23]针对粗糙模糊 K-means 参数获取存在的问题，提出了基于不确定度量的聚类参数获取方法。

在现有的粗糙 K-means 及其改进算法或衍生算法中，设置下近似和边界区域的相对重要性权重系数的方法仍然存在问题。首先，在计算下近似和边界区域权重系数时，忽视下近似和边界区域中数据对象分布的差异性；其次，固定的权重系数忽视了不同簇间的差异性以及每次迭代后簇的变化。针对上述问题，提出一个新的基于空间距离自适应权重度量的粗糙 K-means 算法(Rough K-means Algorithm with Self-adaptive Weights Measurement Based on Space Distance，SDSW-RKM)。SDSW-RKM 算法中，下近似和边界的权重系数取决于下近似和边界区域的对象相对于簇心的分布情况，即若下近似或者边界区域的对象离簇心的平均距离越近，则下近似或者边界区域中对象与簇心的相似度越大，那么，下近似或者边界区域的对象对簇心计算的重要性也就越大。此外，每次迭代都根据新的数据集划分情况动态地计算每个簇的下近似和边界区域权重。通过在 UCI 标准数据集上的仿真分析，验证了本文算法的有效性。

2　粗糙聚类的基本知识

2.1　Lingras 粗糙 K-means 算法

Lingras 等[12]将粗糙集引入 K-means 算法，提出了粗糙 K-means。确切地讲，粗糙 K-means 引用了粗糙集理论中的区间解释，用上近似和下近似来描述每个簇。上近似和下近似的成员要满足以下粗糙集性质：

1)数据对象至多只能属于一个簇的下近似。

2)如果一个对象不属于任何簇的下近似，则一定至少归属于两个簇的上近似。

3)一个簇的上近似中的对象也一定是该簇的上近似的成员。

由于上近似和下近似对象对于簇心的重要性不同，Lingras 等改进了均值计算公式，加入了下近似和边界区域权重。本文中，$\overline{C}_i$ 表示簇 C_i 的上近似，$\underline{C}_i$ 表示下近似，$\hat{C}_i$ 表示边界区域，w_l 表示下近似权重，w_b 表示边界区域权重，m_i 表示簇 C_i 的中心。类簇中心的计算公式如下：

$$m_i=\begin{cases}\omega_l\sum\limits_{x_n\in\underline{C}_i}\dfrac{x_n}{|\underline{C}_i|}+\omega_b\sum\limits_{x_n\in\hat{C}_i}\dfrac{x_n}{|\hat{C}_i|}, & \underline{C}_i\neq\varnothing\wedge\hat{C}_i\neq\varnothing\\ \sum\limits_{x_n\in\underline{C}_i}\dfrac{x_n}{|\underline{C}_i|}, & \underline{C}_i\neq\varnothing\wedge\hat{C}_i=\varnothing\\ \sum\limits_{x_n\in\hat{C}_i}\dfrac{x_n}{|\hat{C}_i|}, & \underline{C}_i=\varnothing\wedge\hat{C}_i\neq\varnothing\end{cases}\tag{1}$$

其中，$w_l+w_b=1$。文献[12]并没有给出上近似和边界区域权重的计算方法，而是通过多次实验来选择使聚类效果最优的权重值。

对于任意对象 x_n，先找到离对象 x_n 最近的簇心 m_p，将 x_n 划分至$\overline{C}_p$。如果存在其他簇心 m_t，使得 $d(x_n,m_t)-d(x_n,m_p)\leqslant\varepsilon$ $(t\neq p)$，则将 x_n 划分到$\overline{C}_t$；否则 $x_n\in\underline{C}_p$。其中，ε 是阈值参数，之后的算法大多采用相对阈值。最后，判断算法是否满足终止条件。

2.2　Peters 粗糙 K-means 算法

Peters[24]详细分析了 Lingras 等的粗糙 K-means 算法，提出了改进的粗糙聚类算法。不同于 Lingras 等的均值函数，Peters 提出的均值函数更加符合粗糙集理论中下近似和上近似的概念。均值计算公式如下：

$$m_k=\omega_l\sum_{x_n\in\underline{C}_k}\frac{x_n}{|\underline{C}_k|}+\omega_u\sum_{x_n\in\overline{C}_k}\frac{x_n}{|\overline{C}_k|}\tag{2}$$

其中，$\omega_l+\omega_u=1$。

Peters 在分配对象的阶段将距离类簇均值最近的对象分配到该簇的下近似，避免了下近似为空的情况，同样也使得上近似不为空。因此，Peters 的均值函数不需要像 Lingras RKM 一样考虑多种情况。此外，Peters 采用相对距离阈值代替了绝对距离阈值：

$$T=\{t:\frac{d(x_n,m_t)}{d(x_n,m_p)}\leqslant\zeta\wedge t\neq p\}\tag{3}$$

相对距离阈值在近几年的算法中使用得比较多，因此本文也采用相对距离阈值。

2.3　πRKM

传统的粗糙聚类算法首先计算下近似和边界区域的子均值，再对两部分的均值进行加权求和，忽略了下近似和边界对象个数的差异。Peters[22]提出了不含权重参数的均值函数公式，直接基于加权的对象进行计算。此外，πRKM 还利用无差异原则来处理边界对象，认为边界对象属于所有该边界对象所在簇的可能性是相同的，其在均值函数中的权重为可能属于的簇的个数的倒数。均值公式如下：

$$m_i=\frac{\sum\limits_{x_n\in\overline{C}_i}\dfrac{x_n}{|B_{x_n}|}}{\sum\limits_{x_n\in\overline{C}_i}\dfrac{1}{|B_{x_n}|}}\tag{4}$$

其中，B_{x_n} 表示边界对象 x_n 可能属于的簇的集合，$|B_{x_n}|$ 是集合 B_{x_n} 中簇的个数。

式(4)的均值函数不含任何权重参数，但是有些情况下又需要设置权重参数，因此 Peters 又给出了以下均值函数：

$$m_i=\frac{w_l\sum\limits_{x_n\in\underline{C_i}}x_n+w_b\sum\limits_{x_n\in\overline{C_i}}\frac{x_n}{|B_{x_n}|}}{w_l|\underline{C_i}|+w_b\sum\limits_{x_n\in\overline{C_i}}\frac{1}{|B_{x_n}|}} \tag{5}$$

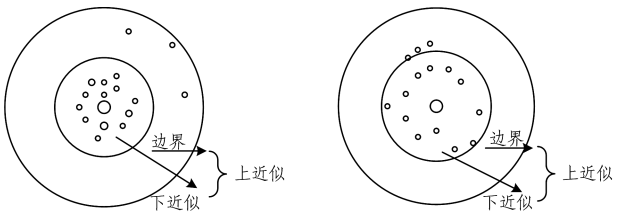
Peters 将式(4)称为标准情况下的均值公式($w_l=w_b=0.5$)，其中不含下近似和边界区域的权重参数。Peters 也指出，在有些情况下，需要设置下近似和边界区域权重，因此给出了式(5)，但没有给出下近似和边界区域权重的计算方法。

3 基于空间距离自适应权重度量的粗糙 K-means 算法

3.1 簇内对象分布情况对簇心的影响分析

在 Lingras 等[12]提出的粗糙 K-means 算法中，下近似中的对象确定属于该簇，边界中的对象可能属于该簇。由于下近似和边界区域的重要性不同，在均值函数中通过权重系数来体现。Lingras 等的算法以及很多变体均是根据经验或者通过多次实验来选取固定的权重系数。固定的权重系数忽视了不同簇间的差异性以及每次迭代后簇的变化。同时，为了选取合适的权重进行多次实验降低了算法的效率，也缺乏科学的依据。此外，现有的粗糙 K-means 算法在计算权重系数时忽略了簇内对象分布的差异性，事实上，类簇中心迭代计算中下近似和边界区域的权重应该同下近似和边界对象与簇心的相对距离有关。

不同的簇中下近似和边界区域对象的分布情况也不同，如图 1 所示。在图 1(a)中，与边界对象相比，下近似对象更紧凑，离簇心的距离更近，边界区域距离簇心相对较远。因此，在计算中心均值时，下近似对象的重要性更大，应该赋予下近似较大的权重，边界区域较小的权重。图 1(b)中下近似对象相对分散，与簇心距离相对较远，而边界区域对象与簇心的距离相对较近。因此，与图 1(a)相比，图 1(b)在计算中心均值时，应该减小下近似权重，增大边界区域权重。



(a)下近似区域数据对象分布较为紧凑　(b)下近似区域数据对象分布较为分散

图 1 下近似和边界区域对象的分布情况

Fig. 1 Distribution of lower approximation and boundary area objects

使用固定的权重会造成给簇内对象分布情况不同的簇设置相同的下近似和边界区域权重，忽略了不同簇间对象分布的差异性。

3.2 基于下近似和边界对象分布差异的自适应权重

一个对象与簇心的距离越近，与簇心的相似度就越大，在计算中心均值时应该给予较大的权重。下近似和边界区域是对象组成的集合，可以根据下近似和边界区域对象与簇心的平均距离来衡量下近似和边界区域对于簇心计算的重要性。

图 2 为下近似和边界区域到簇心 M 的平均距离示意图。L_1 为下近似到簇心的平均距离，L_2 为边界区域到簇心的平均距离。L_1 和 L_2 的相对大小反映了下近似和边界区域对簇心计算的相对重要性，L_1 或 L_2 越小，表明下近似或边界区域对簇心的计算越重要。

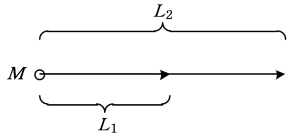


图 2 下近似和边界区域到簇心的平均距离

Fig. 2 Average distance from lower approximation and boundary region to cluster center

从模糊集中隶属度概念的角度来理解，将下近似和边界区域抽象为两个点。用 L_1 与 L_2 的相对大小来衡量簇心到下近似和边界区域的“隶属度”，“隶属度”之和为 1。这里的“隶属度”表示下近似和边界区域对于簇心的重要性。“隶属度”越小，说明下近似或边界区域对于簇心计算的重要性越小；“隶属度”越大，说明下近似或边界区域对于簇心计算的重要性越大。

对于任意的簇 C_i，簇心为 m_i。下近似到簇心的平均距离为：

$$D(lower,i)=\frac{\sum\limits_{x_n\in\underline{C_i}}d(x_n,m_i)}{|\underline{C_i}|} \tag{6}$$

边界区域到簇心的平均距离为：

$$D(border,i)=\frac{\sum\limits_{x_n\in\overline{C_i}}d(x_n,m_i)}{|\overline{C_i}|} \tag{7}$$

参考隶属度函数公式，计算下近似的权重：

$$\begin{aligned}w_l^i&=\frac{1}{\sum\limits_{j=lower,border}\left(\frac{D(lower,i)}{D(j,i)}\right)^m}\\&=\frac{1}{\left(\frac{D(lower,i)}{D(lower,i)}\right)^m+\left(\frac{D(lower,i)}{D(border,i)}\right)^m}\\&=\frac{1}{1+\left(\frac{D(lower,i)}{D(border,i)}\right)^m}\end{aligned} \tag{8}$$

考虑到下近似或者边界为空的情况，可以得到簇 C_i 的下近似权重为：

$$w_l^i=\begin{cases}\dfrac{1}{1+\left(\left(\dfrac{\sum\limits_{x_n\in\underline{C_i}}d(x_n,m_i)}{|\underline{C_i}|}\right)/\left(\dfrac{\sum\limits_{x_n\in\overline{C_i}}d(x_n,m_i)}{|\overline{C_i}|}\right)\right)^m}, & \underline{C_i}\neq\varnothing\wedge\overline{C_i}\neq\varnothing\\1, & \underline{C_i}\neq\varnothing\wedge\overline{C_i}=\varnothing\\0, & \underline{C_i}=\varnothing\wedge\overline{C_i}\neq\varnothing\end{cases} \tag{9}$$

其中，m 是指数参数且 $m>1$。$w_b^i+w_l^i=1$，即 $w_b^i=1-w_l^i$。

当类簇的下近似为空时，只存在边界区域的数据对象，类簇中心仅由边界区域的数据对象决定，因此边界区域权重设置为 1。为了避免下近似为空的情况，在算法迭代过程中每次都将距离簇心最近的对象分配到该簇的下近似。当边界区域为空时，则只有下近似的对象，下近似对簇心的重要性最大，因此下近似权重设置为 1。

一般，下近似到簇心的平均距离小于边界区域到簇心的平均距离，即 $\frac{D(lower,i)}{D(border,i)}\in(0,1)$，因此下近似权重 $w_l^i\in(0.5,1)$，边界区域权重 $w_b^i\in(0,0.5)$。当下近似到簇心的平均距离远小于边界区域到簇心的平均距离时，$\frac{D(lower,i)}{D(border,i)}$ 趋近于 0；下近似集权重 w_l^i 趋近于 1，边界区域权重 w_b^i 趋近于 0。反之，当下近似到簇心的平均距离约等于上近似到簇心的平均距离时，$\frac{D(lower,i)}{D(border,i)}$ 趋近于 1，下近似权重 w_l^i 趋近于 0.5，边界区域权重 w_b^i 也趋近于 0.5。

3.3 SDSW-RKM 算法

本文参考了文献[22]的均值公式(式(5))，直接对个体对象加权，而非先计算下近似和边界区域的子均值，再对子均值加权求和。结合本文提出的权重计算方法(见式(9))，新的均值公式如下：

$$m_i=\begin{cases}\dfrac{w_l^i\sum\limits_{x_n\in\underline{C_i}}x_n+w_b^i\sum\limits_{x_n\in\hat{C}_i}\frac{x_n}{|B_{x_n}|}}{w_l^i|\underline{C_i}|+w_b^i\sum\limits_{x_n\in\hat{C}_i}\frac{1}{|B_{x_n}|}}, & \underline{C_i}\neq\varnothing\wedge\hat{C}_i\neq\varnothing\\ \dfrac{\sum\limits_{x_n\in\underline{C_i}}x_n}{|\underline{C_i}|}, & \underline{C_i}\neq\varnothing\wedge\hat{C}_i=\varnothing\\ \dfrac{\sum\limits_{x_n\in\hat{C}_i}\frac{x_n}{|B_{x_n}|}}{\sum\limits_{x_n\in\hat{C}_i}\frac{1}{|B_{x_n}|}}, & \underline{C_i}=\varnothing\wedge\hat{C}_i\neq\varnothing\end{cases}\tag{10}$$

其中，$w_b^i=1-w_l^i$。

当下近似为空时，边界区域权重为 1；当边界区域为空时，下近似权重为 1，如式(10)所示。

SDSW-RKM 算法处理 N 个数据对象$\{x_n\}$的过程如下。

算法 1　SDSW-RKM 算法

输入：数据集，类簇个数 K，相对距离阈值 ζ，指数参数 m

输出：K 个粗糙类簇

Step1　初始化。产生初始簇心，初始分配每个数据对象到某一个簇的下近似中(比如到最近的簇的下近似)。

Step2　根据式(9)计算每个簇下近似和边界区域的权重：w_l^i 和 w_b^i，$w_b^i=1-w_l^i$。

Step3　根据均值式(10)计算每个簇的均值。

Step4　1)将离每个簇中离簇心最近的对象放到该簇的下近似中，保证每个簇的下近似不为空；

2)对于剩余对象 x_r($r=1,\cdots,N-K$)，计算离其最近的中心 m_p，若 x_r 距离其他中心 m_t 的距离与其到中心 m_p 的距离之比不超过阈值 ζ，则将 x_r 分配到簇 $\overline{C_t}$ 和 $\overline{C_p}$，否则分配到 $\underline{C_p}$。

Step5　如果算法收敛或达到最大迭代次数，则停止；否则转 Step2。

4　仿真实验

4.1　实验环境和实验对象

为了验证本文算法的有效性，将 SDSW-RKM 算法与相关算法进行仿真实验分析。实验环境如下：操作系统为 Windows 7，CPU 主频为 2.20 GHz，4 核，内存为 2 GB，仿真软件为 MATLAB 7.10.0。对来源于 UCI 数据库的 3 个数据集(Iris，Wine 和 Ionosphere)进行仿真分析。数据集的具体特征如表 1 所列。

表 1　数据集信息

Table 1　Information of data sets

数据集	样本个数	分类属性个数	决策属性个数	类簇数
Iris	150	4	1	3
Wine	178	13	1	3
Ionosphere	351	34	1	2

4.2　评价指标

借鉴文献[22]的 4 个基本标准与文献[25]的 Davies-Bouldin (DB)指标和 Dunn 指标来评估聚类结果的质量。

文献[22]采用以下 4 个标准来评估聚类质量：1)Ok，表示类簇的下近似中分类正确的对象个数，该值越大越好；2)πOk，表示下近似和边界区域正确分类的对象个数，包括正确分配到下近似的对象(Ok)以及正确分配到边界的对象，边界正确的对象占 $\frac{1}{|B_{x_n}|}$，该值越大越好；3)$\neg Ok$，表示下近似中被错误分类的对象个数，该值越小越好；4)Bd，表示边界对象的个数，该值越小越好。

常用的评价指标还有 Davies-Bouldin (DB)[25] 和 Dunn[25]。DB 指标即最小化簇内距离 $S(v_i)$，最大化簇间距离 $d(v_i,v_k)$。簇内越相似，簇间分隔越大，DB 指标越小，聚类效果也越好。而 Dunn 指标则越大越好。

DB 指标的公式为：

$$DB=\frac{1}{c}\sum_{i=1}^{c}\max_{k\neq i}\left\{\frac{S(v_i)+S(v_k)}{d(v_i,v_k)}\right\}\tag{11}$$

其中，簇内距离 $S(v_i)$ 参考了文献[22]的公式：

$$\begin{aligned}S(v_i)&=\frac{\underline{w}\sum\limits_{x_n\in\underline{C_i}}\|x_n-m_i\|^2+\hat{w}\sum\limits_{x_n\in\hat{C}_i}\frac{\|x_n-m_i\|^2}{|B_{x_n}|}}{\underline{w}|\underline{C_i}|+\hat{w}\sum\limits_{x_n\in\hat{C}_i}\frac{1}{|B_{x_n}|}}\\&=\frac{\sum\limits_{x_n\in\overline{C_i}}\frac{\|x_n-m_i\|^2}{|B_{x_n}|}}{\sum\limits_{x_n\in\overline{C_i}}\frac{1}{|B_{x_n}|}}\end{aligned}\tag{12}$$

Dunn 指标的公式为：

$$Dunn=\min_i\left\{\min_{k\neq i}\left\{\frac{d(v_i,v_k)}{\max\limits_l S(v_l)}\right\}\right\}\tag{13}$$

4.3　聚类效果比较

使用 4.1 节中的部分数据集进行实验分析，将数据集归一化到区间[0,1]，同一个数据集选取相同的初始中心。阈值参数 $\zeta=1.2,1.4,1.6,1.8$。Lingras RKM，Peters RKM 和 πRKM 算法选取下近似权重 $w_l=0.5,0.7,0.9$。SDSW-RKM 算法不需要设置权重参数，指数参数 $m=4$。不同算法在权重变化时的聚类结果如表 2 所列。

表 2 不同算法聚类效果的比较

Table 2 Comparison of clustering results with different algorithms

w_l			0.5				0.7				0.9			
数据集	算法	ζ	*Ok*	πOk	$\rightarrow Ok$	*Bd*	*Ok*	πOk	$\rightarrow Ok$	*Bd*	*Ok*	πOk	$\rightarrow Ok$	*Bd*
Iris	Lingras RKM	1.2	127	135.50	6	17	131	133.00	15	4	130	132.50	15	5
		1.4	117	131.50	4	29	121	133.00	5	24	125	135.00	5	20
		1.6	111	129.50	2	37	118	131.50	5	27	118	132.00	4	28
		1.8	88	117.50	1	61	110	129.33	1	39	114	131.00	2	34
	Peters RKM	1.2	131	133.50	14	5	130	132.50	15	5	130	132.50	15	5
		1.4	125	135.00	5	20	125	135.00	5	20	125	135.00	5	20
		1.6	121	134.00	3	26	121	134.00	3	26	119	132.50	4	27
		1.8	114	131.00	2	34	114	131.00	2	34	114	130.00	4	32
	πRKM	1.2	131	133.50	14	5	130	132.50	15	5	130	132.50	15	5
		1.4	124	134.50	5	21	125	135.00	5	20	125	135.00	5	20
		1.6	119	132.00	5	26	121	134.00	3	26	119	132.50	4	27
		1.8	113	130.83	1	36	114	131.00	2	34	114	130.00	4	32
	SDSW-RKM	1.2	130	132.50	15	5	—	—	—	—	—	—	—	—
		1.4	125	135.00	5	20	—	—	—	—	—	—	—	—
		1.6	119	133.00	3	28	—	—	—	—	—	—	—	—
		1.8	115	130.50	4	31	—	—	—	—	—	—	—	—
Wine	Lingras RKM	1.2	137	152.33	9	32	146	159.83	4	28	152	163.50	3	23
		1.4	125	147.33	2	51	129	151.00	1	48	138	155.67	1	39
		1.6	86	124.33	1	91	117	142.33	1	60	121	145.83	1	56
		1.8	33	89.83	0	145	92	126.17	1	85	101	133.17	1	76
	Peters RKM	1.2	153	164.00	3	22	155	165.00	3	20	156	165.50	3	19
		1.4	136	154.67	1	41	137	155.17	1	40	136	155.17	1	41
		1.6	114	141.83	1	63	121	145.83	1	56	125	147.67	1	52
		1.8	77	118.83	0	101	98	130.83	1	79	103	134.33	1	74
	πRKM	1.2	150	162.33	3	25	155	165.00	3	20	157	167.00	1	20
		1.4	135	154.17	1	42	137	155.50	1	40	136	155.17	1	41
		1.6	105	137.17	1	72	119	144.83	1	58	125	147.67	1	52
		1.8	—	—	—	—	91	127.50	0	87	102	133.83	1	75
	SDSW-RKM	1.2	157	167.00	1	20	—	—	—	—	—	—	—	—
		1.4	136	155.17	1	41	—	—	—	—	—	—	—	—
		1.6	125	147.67	1	52	—	—	—	—	—	—	—	—
		1.8	101	133.67	1	76	—	—	—	—	—	—	—	—
Ionosphere	Lingras RKM	1.2	182	226.00	81	88	191	225.00	92	68	202	231.00	91	58
		1.4	152	228.00	47	152	163	225.00	64	124	169	222.00	76	106
		1.6	124	233.50	8	219	142	221.00	51	158	151	223.00	56	144
		1.8	98	222.00	5	248	137	225.00	38	176	141	222.50	47	163
	Peters RKM	1.2	202	231.00	91	58	203	231.50	91	57	204	232.00	91	56
		1.4	168	223.50	72	111	168	222.50	74	109	169	220.50	79	103
		1.6	148	226.50	46	157	147	223.50	51	153	148	220.00	59	144
		1.8	117	229.50	9	225	138	221.50	46	167	142	223.00	47	162
	πRKM	1.2	202	231.00	91	58	201	230.00	92	58	203	231.50	91	57
		1.4	156	227.50	52	143	172	225.00	73	106	168	219.50	80	103
		1.6	—	—	—	—	147	225.50	47	157	147	219.00	60	144
		1.8	—	—	—	—	98	221.50	6	247	135	218.50	49	167
	SDSW-RKM	1.2	206	232.50	92	53	—	—	—	—	—	—	—	—
		1.4	168	219.50	80	103	—	—	—	—	—	—	—	—
		1.6	148	220.00	59	144	—	—	—	—	—	—	—	—
		1.8	144	224.00	47	160	—	—	—	—	—	—	—	—

由表 2 可知，对于不同的数据集和不同的相对阈值，各个算法对于权重的敏感性不同。在 Iris 数据集上，阈值相同时，Lingras RKM 的聚类结果对于权重的变化敏感，而 Peters RKM 和 πRKM 在权重变化时结果稳定。Peters RKM，πRKM 和 SDSW-RKM 算法的聚类结果非常相似。

在 Wine 数据集上，Lingras RKM 的聚类结果对于权重的变化非常敏感，聚类结果波动很大。Lingras RKM 算法在 ζ=1.2，1.6 时，聚类结果不如 Peters RKM，πRKM 和 SDSW-RKM 算法。在 ζ=1.4，1.8 时，Lingras RKM 最佳权重对应的聚类结果与 Peters RKM，πRKM 的最佳聚类结果相当，与 SDSW-RKM 算法的聚类结果接近。在 ζ=1.4 时，Peters RKM 和 πRKM 的聚类结果在不同的权重选择下变化不大，与 SDSW-RKM 算法相似。在 ζ=1.2，1.6，1.8 时，Peters RKM 和 πRKM 算法对权重的选择较为敏感，聚类结果随着权重取值的变化出现较明显的差异，并且在最佳权重取值下对应的聚类结果与 SDSW-RKM 算法的聚类结果相当。

针对 Ionosphere 数据集进行聚类分析也可得到类似的结果。

由上述分析可知，Lingras RKM 关于权重变化最敏感。在某些情况下，Peters RKM 和 πRKM 算法的聚类结果对于权重的选择也较为敏感，需要多次实验选取最佳权重，最佳权重对应的聚类结果与 SDSW-RKM 算法的结果相似。而 SD-

SW-RKM 算法中下近似和边界区域权重不需要多次实验就能得到较好的聚类结果，提高了算法的效率。当算法的聚类结果对权重的选择敏感时，即体现出 SDSW-RKM 算法的优越性。

4.4 DB 和 Dunn 指标的比较

在 Iris，Wine 和 Ionosphere 3 个数据集上进行仿真实验。聚类结果采用 DB 和 Dunn 指标进行对比，使用式(11)、式(13)来计算 DB 和 Dunn 指标的值。由于聚类指标与聚类算法均值公式的形式有关，因此只对 SDSW-RKM 算法和 πRKM 算法进行比较。

实验之前，将所有的数据集先归一化到区间[0，1]，同一数据集采用相同的初始中心。πRKM 运行在标准模式下，即 $w_l=w_b=0.5$。SDSW-RKM 算法不需要设置权重参数，指数参数 $m=4$。阈值参数 $\zeta=1.2,1.4,1.6,1.8$。

两个算法的 DB 和 Dunn 指标的对比结果如表 3 所列。由表 3 可知，在所有数据集上，对于所有不同取值的阈值，SDSW-RKM 算法的 DB 指标值均比 πRKM 的 DB 指标值小，SDSW-RKM 算法的 Dunn 指标也比所有的 πRKM 算法的 Dunn 指标值大。这说明 SDSW-RKM 算法的 DB 指标和 Dunn 指标均优于 πRKM 算法，SDSW-RKM 算法的聚类效果更好。SDSW-RKM 算法在使簇内对象更加紧凑、簇间分隔尽可能远方面具有更好的效果。

表 3　DB 和 Dunn 指标对比结果

Table 3　Comparison results of DB and Dunn

算法		πRKM				SDSW-RKM			
ζ		1.2	1.4	1.6	1.8	1.2	1.4	1.6	1.8
Iris	DB	0.1885	0.2071	0.2144	0.2953	0.1818	0.1839	0.1951	0.2004
	Dunn	8.5895	7.1981	6.9604	5.2913	9.2500	8.4514	8.3323	7.8991
Wine	DB	0.7828	0.8666	1.1990	—	0.7159	0.7297	0.7677	0.8826
	Dunn	2.2378	2.0856	1.4950	—	2.3317	2.5254	2.4007	2.0698
Ionosphere	DB	2.7383	4.2560	—	—	2.3369	2.4315	2.5550	2.6923
	Dunn	0.5862	0.3880	—	—	0.6731	0.6705	0.6023	0.5513

4.5 运行时间

将 3 个数据集的数据进行归一化。Lingras RKM，Peters RKM 算法的权重设为 $w_l=0.7$，$w_b=0.3$。πRKM 算法运行在标准模式下，即 $w_l=w_b=0.5$。SDSW-RKM 算法不需要设置权重参数，指数参数 $m=4$。阈值参数 $\zeta=1.2,1.4,1.6,1.8$。由于 SDSW-RKM 算法需要通过计算得到下近似和边界区域的权重，因此实验对几种聚类算法的仿真时间和迭代次数进行对比，结果如表 4 所列。由表 4 可以看出，SDSW-RKM 算法在运行时间上并不慢且迭代次数也不算多。

表 4　算法的运行时间和迭代次数的对比

Table 4　Comparison of running time and iteration numbers of algorithms

数据集	ζ	Lingras RKM		Peters RKM		πRKM		SDSW-RKM	
		运行时间/s	迭代次数	运行时间/s	迭代次数	运行时间/s	迭代次数	运行时间/s	迭代次数
Iris	1.2	0.040081	3	0.075796	4	0.087946	6	0.093037	4
	1.4	0.070063	6	0.095718	7	0.089107	6	0.121408	9
	1.6	0.061329	4	0.110058	9	0.083272	5	0.123597	9
	1.8	0.090043	10	0.085400	5	0.097533	7	0.084361	3
Wine	1.2	0.085368	7	0.125770	8	0.104682	7	0.147261	9
	1.4	0.087525	7	0.100829	5	0.108254	7	0.116469	5
	1.6	0.083379	6	0.104036	5	0.093829	5	0.128283	6
	1.8	0.100569	8	0.142500	8	—	—	0.127421	6
Ionosphere	1.2	0.153972	6	0.198630	5	0.105835	4	0.189058	4
	1.4	0.159951	6	0.210227	5	0.212299	11	0.287566	7
	1.6	0.227246	9	0.358144	9	—	—	0.402196	10
	1.8	0.272009	11	0.365537	9	—	—	0.483219	12

总体来说，SDSW-RKM 算法改进了聚类结果，且并未在运行时间和迭代次数上做出太多牺牲。

结束语　下近似和边界区域对于簇心计算的重要性不同，如何选择下近似和边界区域的权重将影响算法的聚类结果。本文根据下近似和边界对象相对于簇心的分布情况，设计了新的基于空间距离自适应权重度量的粗糙 K-means 算法。通过与其他算法在 UCI 数据集上的对比分析，验证了本文算法的有效性，尤其是在聚类结果对于权重的选择较为敏感的情况下，本文算法具有优越性。

参 考 文 献

[1] HAN J，KAMBER M. Data mining，concepts and techniques (third edition) [M]. California：Morgan Kaufmann Publishers，2011.

[2] HARTIGAN J A，WONG M A. A K-Means Clustering Algorithm [J]. Journal of the Royal Statistical Society，1979，28(1)：100-108.

[3] AMORIM R C D. A Survey on Feature Weighting Based K-Means Algorithms[J]. Journal of Classification，2016，33(2)：210-242.

[4] KHANDARE A，ALVI A S. Survey of Improved k-means Clustering Algorithms：Improvements，Shortcomings and Scope for Further Enhancement and Scalability[C]// Information Systems Design and Intelligent Applications. New Delhi：Springer，2016：495-503.

[5] BEZDEK J C. Fuzzy Mathematics in Pattern Classification[D]. Ithaca:Cornell University,1973.

[6] BEZDEK J C. Pattern Recognition with Fuzzy Objective function Algorithms [M]. New York:Plenum Press,1981.

[7] JIANG Z H,LI T T,MIN W F,et al. Fuzzy c-means clustering based on weights and gene expression programming [J]. Pattern Recognition Letters,2017,90:1-7.

[8] CHEN H P,SHEN X J,LV Y D,et al. A novel automatic fuzzy clustering algorithm based on softpartition and membership information [J]. Neurocomputing,2017,236(SI):104-112.

[9] KRISHNAPURAM R,KELLER J M. A Possibilistic Approach to Clustering [J]. IEEE Transactions on Fuzzy Systems,1993,1(2):98-110.

[10] KRISHNAPURAM R,KELLER J M. The Possibilistic C-Means Algorithm: Insights and Recommendations [J]. IEEE Transactions on Fuzzy Systems,1996,4(3):385-393.

[11] BARNIM,CAPPELLINI V,MECOCCI A. Comments on A Possibilistic Approach to Clustering [J]. IEEE Transactions on Fuzzy Systems,1996,4(3):393-396.

[12] 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.

[13] PAWLAK Z. Rough Sets[J]. International Journal of Computer & Information Sciences,1982,11(5):341-356.

[14] MAJI P,PAL S K. RFCM:A Hybrid Clustering Algorithm Using Rough and Fuzzy Sets[J]. Fundamenta Informaticae,2007,80(4):475-496.

[15] MITRA S,BANKA H,PEDRYCZ W. Rough-Fuzzy Collaborative Clustering [J]. IEEE Transactions on Systems Manand Cybernetics Part B Cybernetics,2006,36(4):795-805.

[16] PAUL S,MAJI P. A New Rough-Fuzzy Clustering Algorithm and its Applications [C]// Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012). New Delhi:Springer,2014:1245-1251.

[17] SHI J,LEI Y,ZHOU Y,et al. Enhanced rough-fuzzy c-means algorithm with strict rough setsproperties [J]. Applied Soft Computing,2016,46:827-850.

[18] ZHANG T F,CHEN L,MA F M. A modified rough c-means-clustering algorithm based on hybrid imbalanced measureof distance and density[J]. International Journal of Approximate Reasoning,2014,55(8):1805-1818.

[19] WANG H,ZHOU M. A refined rough k-means clustering with hybrid threshold [C]// International Conference on Rough Sets and Current Trends in Computing. Berlin Heidelberg:Springer,2012:26-35.

[20] MALYSZKO D,STEPANIUK J. Rough Entropy Based k-Means Clustering [C]// Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. Berlin Heidelberg: Springer, 2009: 406-413.

[21] PAL S K,SHANKAR B U,MITRA P. Granular computing, rough entropy and object extraction [J]. Pattern Recognition Letters,2005,26(16):2509-2517.

[22] PETERS G. Rough Clustering Utilizing the Principle of Indifference [J]. Information Sciences,2014,277(2):358-374.

[23] WANG X E,HAN D Q,HAN C Z. Selection method for Parameters of Rough Fuzzy C-Means Clustering Based on Uncertainty Measurement[J]. Journal of Xi'an Jiaotong University,2013,47(6):55-60. (in Chinese)
王学恩,韩德强,韩崇昭. 采用不确定性度量的粗糙模糊 C 均值聚类参数获取方法[J]. 西安交通大学学报,2013,47(6):55-60.

[24] PETERS G. Some refinements of rough k-means clustering [J]. Pattern Recognition,2006,39(8):1481-1491.

[25] BEZDEK J C,PAL N R. Some New Indexes of Cluster Validity [J]. IEEE Transactions on System Manand Cybernetics Part B Cybernetics,1988,28(3):301-315.

(上接第 166 页)

[10] KAMARA S,PAPAMANTHOU C,ROEDER T. Dynamic searchable symmetric encryption[C]// Acm Conference on Computer & Communications Security. ACM,2012:965-976.

[11] KAMARA S,PAPAMANTHOU C. Parallel and Dynamic Searchable Symmetric Encryption[M]// Berlin:Springer,2013:258-274.

[12] CASH D,JARECKI S,JUTLA C,et al. Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries [M]. Berlin:Springer,2013:353-373.

[13] AU M H,TSANG P P,SUSILO W,et al. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems [M]. Berlin: Springer, 2009:295-308.

[14] SHAO J,CAO Z,LIANG X,et al. Proxy re-encryption with keyword search[J]. Information Sciences,2010,180(13):2576-2587.

[15] LEE S H,LEE I Y. A Study of Practical Proxy Reencryption with a Keyword Search Scheme considering Cloud Storage Structure[J]. The Scientific World Journal, 2014, 2014(2): 1661-1667.

[16] FANG L,SUSILO W,GE C,et al. Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search [J]. Theoretical Computer Science,2012,4629(1):39-58.

[17] DODIS Y,OSTROVSKY R,REYZIN L,et al. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data[J]. SIAM Journal on Computing,2008,38(1):97-139.