文章编号: 1001-0920 (2013) 10-1479-06

基于簇内不平衡度量的粗糙 K-means 聚类算法

张腾飞[a], 陈 龙[a], 李 云[b]

(南京邮电大学 a. 自动化学院，b. 计算机技术研究所，南京 210023)

摘 要: 粗糙 K-means 聚类算法是一种有效的处理聚类边界模糊问题的算法, 但大多数算法对簇的下近似集和边界中的对象使用统一的权值, 忽略了簇内对象之间的差异性. 针对这一问题提出一种新的改进算法, 通过对簇内的每个对象加入簇内不平衡度量, 以区分不同对象对簇的贡献程度, 使得聚类结果簇内更紧凑、簇间更疏远. 不同数据集的仿真实验结果表明, 所提出算法可以有效提高聚类结果的精度.

关键词: 簇内不平衡度量; 粗糙集; 粗糙 K-means 聚类

中图分类号: TP18 **文献标志码:** A

Rough K-means clustering based on unbalanced degree of cluster

ZHANG Teng-fei[a], *CHEN Long*[a], *LI Yun*[b]

(a. College of Automation，b. Institute of Computer Technology，Nanjing University of Posts and Telecomunications, Nanjing 210023，China．Correspondent：ZHANG Teng-fei，E-mail：tfzhang@126.com)

Abstract: Rough K-means clustering is a valid algorithm to process the inseparability of border of clusters. But to most algorithms, weights of objects in the lower approximate set or the upper approximate set are all the same without paying attention to the diversity in clusters. Therefore, a new algorithm is proposed. The algorithm can make the cluster has a more compact center, and the borders are separated each other with the unbalanced degree of cluster which means the contribution of an object to the cluster. The simulation analysis shows that this algorithm can improve the precision of the clustering results effectively.

Key words: unbalanced degree of cluster; rough set; rough K-means clustering

0 引 言

聚类是一种将数据对象分成类或簇的过程, 它要求簇内数据对象高度相似, 而簇间数据对象高度相异. 聚类过程中通过某种差异度量将数据对象区分开, 常见的差异度量有距离度量等. 聚类方法大致可以分为划分方法、层次方法、密度方法、栅格方法和模型方法. 目前, 聚类算法在包括数据挖掘、统计学、机器学习、空间数据库技术等众多研究领域得到了广泛应用[1].

Lingras 等[2-3]在对 web 数据进行挖掘时, 发现 K 均值聚类的结果往往有一个模糊的、粗糙的、分辨不清的边界, K 均值的“硬划分”聚类方法不能满足对含有模糊性数据聚类的需求. Lingras 将粗糙集理论融入到 K 均值聚类算法中, 每个簇看作是一个粗糙集, 每一个对象或者确定属于某个簇的下近似集, 或者同时属于多个簇的上近似集. 该算法虽然提高了聚类边界的聚类精度, 但其中心均值迭代公式只考虑了簇的上下近似集非空、上近似集为空这两种情况; 算法使用欧氏距离作为相异度评判标准, 聚类结果在很大程度上受到孤立点的影响; 且传统的粗糙 K-means 算法只考虑了簇内边界对象的不可分辨性, 对多个对象使用统一的权值, 忽略了簇内对象的差异性.

Peters[4-5]对粗糙 K-means 算法的中心均值迭代公式作了改进, 对仅有下近似集或上近似集为空的粗糙权值赋 1, 表示上近似集或下近似集中对象确定归属于当前簇. 使用相对距离代替粗糙 K-means 中的绝对距离作为相异度判断标准, 有效地减小了离群点

收稿日期: 2012-07-05; **修回日期:** 2012-09-02.

基金项目: 国家自然科学基金项目(61105082, 61073114); 南京邮电大学“攀登计划”项目(NY212093); 江苏省教育厅高校自然科学基金基础研究项目(11KJB120001).

作者简介: 张腾飞(1980–), 男, 副教授, 博士, 从事智能信息处理、智能控制等研究; 陈龙(1988–), 男, 硕士生, 从事模式识别与智能系统的研究.

的影响. 文献 [6-7] 分别介绍了粗糙模糊 K 均值和模糊粗糙 K 均值方法, 综合了两者的特点, 利用模糊隶属度衡量聚内对象对簇间的贡献程度, 在粗糙 K 均值的基础上提高了聚类边界精度. 有些学者[6,8]还提出了 Collaborative RKM 的方法, 合并聚类结果中意义不明的簇, 从而提高了聚类精度.

上述改进算法虽然能够提高边界的聚类精度, 但依然存在问题. 粗糙 K 均值算法将一个簇分割成两部分, 忽略了内部不同对象间的区别. 如在下近似集中, 每个对象与中心均值的距离不同, 表明对象与中心的紧密程度不同[9-10]. 统一赋予相同的权值, 必定造成中心均值点有较大程度的移动, 从而影响聚类精度. 本文针对这一问题, 提出一种基于簇内不平衡度量的粗糙 K 均值聚类算法, 根据每个对象与中心点距离不同, 赋予不同的簇内不平衡度. 越是靠近中心的对象, 簇内不平衡度越高, 在中心迭代时的贡献程度便越高, 最终使得靠近中心的点更加聚集. 通过对 UCI 数据集进行聚类仿真实验分析验证了所提出算法的有效性.

1 粗糙 K-means 聚类算法

1.1 粗糙 K 均值算法的基本原理

粗糙集理论由 Pawlak[11]提出, 是继 Zadeh 模糊集理论后处理不精确、模糊问题的又一重要工具. 粗糙集理论的主要思想是使用上近似集和下近似集描述一个集合. 对于任意子集 $X \in U$, 有

$$\underline{\mathrm{Apr}}_A(X) = \{x | x \in U \bigwedge [x]_A \subseteq X\}, \tag{1}$$

$$\overline{\mathrm{Apr}}_A(X) = \{x | x \in U \bigwedge [x]_A \bigcap X \neq \varnothing\}. \tag{2}$$

其中: A 为等价空间, $[x]_A$ 为等价类, $\underline{\mathrm{Apr}}_A(X)$ 为 x 的下近似集, $\overline{\mathrm{Apr}}_A(X)$ 为 x 的上近似集.

Lingras 提出的粗糙 K 均值聚类算法将聚类的每一个簇看作一个粗糙集, 即每个簇都有一个下近似集, 一个上近似集. 在处理聚类结果边界模糊的问题上, 对边界点应使用以下几种情况:

1) 聚类对象确定的属于某一个簇的下近似集;

2) 聚类对象属于多个簇的上近似集;

3) 聚类对象不可能同时属于某个下近似集, 又属于其他簇的多个上近似集[2,9].

粗糙 K 均值算法的具体步骤如下.

Step 1: 初始化总数为 N 的数据集、目标聚类个数 k、初始聚类中心 $C_i(i = 1, 2, \cdots, k)$、下近似权值 W_{low}、上近似权值 W_{up} 和距离判断阈值 Δ.

Step 2: 对于聚类对象 $X_j(j = 1, 2, \cdots, N)$, 计算其到各中心的欧氏距离, 并将 X_j 归到最近的中心 C_i 对应簇 U_i 的上近似集 $\overline{B}U_i$.

Step 3: 若存在中心 C_i', 使得 X_j 到 C_i' 的距离和 X_j 到 C_i 的距离之差小于 Δ, 则将 X_j 归到簇 U_i' 的上近似集 $\overline{B}U_i'$, 否则, 将 X_j 归到簇 U_i' 的下近似集 $\underline{B}U_i'$.

Step 4: 迭代中心或均值, 迭代公式如下:

$$C_i = \begin{cases} W_{\mathrm{low}} \times \dfrac{\sum\limits_{X_j \in \underline{B}U_i} X_j}{|\underline{B}U_i|} + W_{\mathrm{up}} \times \dfrac{\sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} X_j}{|(\overline{B}U_i - \underline{B}U_i)|}, \\ \qquad \overline{B}U_i - \underline{B}U_i \neq \varnothing; \\ W_{\mathrm{low}} \times \dfrac{\sum\limits_{X_j \in \underline{B}U_i} X_j}{|\underline{B}U_i|}, \ \text{otherwise}. \end{cases} \tag{3}$$

Step 5: 重复 Step 2 ~ Step 4, 直到没有新的数据对象.

1.2 粗糙 K 均值的一些改进算法

Peters 对粗糙 K 均值的中心迭代算法进行改进, 考虑了下近似集为空集而边界非空的, 去除了下近似集非空而边界为空和下近似集为空而边界非空这两种情况下中心迭代的粗糙权值, 修改后的公式为

$$C_i = \begin{cases} W_{\mathrm{low}} \times \dfrac{\sum\limits_{X_j \in \underline{B}U_i} X_j}{|\underline{B}U_i|} + W_{\mathrm{up}} \times \dfrac{\sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} X_j}{|(\overline{B}U_i - \underline{B}U_i)|}, \\ \qquad \underline{B}U_i \neq \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) \neq \varnothing; \\ \dfrac{\sum\limits_{X_j \in \underline{B}U_i} X_j}{|\underline{B}U_i|}, \ \underline{B}U_i \neq \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) = \varnothing; \\ \dfrac{\sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} X_j}{|(\overline{B}U_i - \underline{B}U_i)|}, \ \underline{B}U_i = \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) \neq \varnothing. \end{cases} \tag{4}$$

粗糙 K 均值算法赋予下近似集和边界中的对象以相同的权值, 忽略了对象之间的差异性. 隶属度是区别对象间差异的有力手段, 隶属度基于每个对象到各个簇的距离进行计算, 表明各个对象与簇的关联程度, 隶属度的值越大, 表明对象与该簇的关系越紧密. 隶属度权值的计算方法[12]为

$$\mu_{ik} = 1 \Big/ \sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}. \tag{5}$$

粗糙模糊 K 均值[5]在粗糙 K 均值的基础上加入了模糊隶属度权值, 并以隶属度作为相异度判断准则, 其中心迭代公式为

$$C_i =$$

$$\begin{cases} W_{\text{low}} \times \dfrac{\sum\limits_{X_j \in \underline{B}U_i} \mu_{ij}^m X_j}{\sum\limits_{X_j \in \underline{B}U_i} \mu_{ij}^m} + W_{\text{up}} \times \dfrac{\sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} \mu_{ij}^m X_j}{\sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} \mu_{ij}^m}, \\ \qquad \underline{B}U_i \neq \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) \neq \varnothing; \\ \dfrac{\sum\limits_{X_j \in \underline{B}U_i} \mu_{ij}^m X_j}{\sum\limits_{X_j \in \underline{B}U_i} \mu_{ij}^m}, \ \underline{B}U_i \neq \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) = \varnothing; \\ \dfrac{\sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} \mu_{ij}^m X_j}{\sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} \mu_{ij}^m}, \ \underline{B}U_i = \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) \neq \varnothing. \end{cases} \tag{6}$$

模糊粗糙 K 均值算法[7]对粗糙模糊 K 均值的隶属度计算公式作了进一步改进, 所有下近似集中的对象隶属度均赋值为 1, 认为确定属于当前簇. 模糊粗糙 K 均值算法的中心迭代公式为

$$C_i = \sum_{k=1}^{N} \mu_{ik}^m X_k \Big/ \sum_{k=1}^{N} \mu_{ik}^m. \tag{7}$$

2　基于簇内不平衡度量的粗糙 K 均值算法

2.1　簇内不平衡度量

本文提出簇内不平衡度作为中心迭代时每个对象的权值计算如下:

$$M_{ij} = \frac{2/\pi \times \arctan(-\|X_j - C_i\|^2) + 1}{\sum\limits_{X_k \in U_i} (2/\pi \times \arctan(-\|X_k - C_i\|^2) + 1)}, \tag{8}$$

其中 $\|X_j - C_i\|$ 为对象 X_j 到所在簇中心 C_i 的欧氏距离. 距离是对象在簇内不均衡分布最直接的度量方式, 但归一化后的欧氏距离值差异很小, 且呈线性分布, 直接使用距离作为簇内对象的不平衡度量并不理想. 式 (8) 采用函数 $y = 2/\pi \times \arctan(-x) + 1$ 重新分配距离二范数值的分布, 函数曲线如图 1 所示.

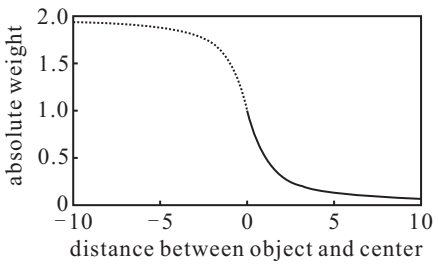

图 1　改进的反正切曲线

由图 1 可见, 式 (8) 采用的是正半轴实线部分, 根据距离范围 x 轴取值范围为 $[0, +\infty)$, y 轴取值范围为 $(0, 1]$. 越靠近原点的部分权值越高, 表明越靠近中心的对象贡献程度越高. 而曲线越向右走, 权值下降越快, 到边界附近趋于平稳, 表明距离中心过远的点对该簇的影响较小.

2.2　基于簇内不平衡度量的粗糙 K 均值聚类算法

将簇内不平衡度融入粗糙 K 均值算法, 改进粗糙 K 均值算法的中心均值迭代公式. 算法的具体步骤如下.

Step 1: 初始化总数为 N 的数据集、目标聚类个数 k、聚类中心 $C_i (i = 1, 2, \cdots, k)$、下近似权值 W_{low}、上近似权值 W_{up} 和距离判断阈值 Δ.

Step 2: 对于聚类对象 $X_j (j = 1, 2, \cdots, N)$, 计算其到各中心的欧氏距离, 并将 X_j 归到最近的中心 C_i 对应簇 U_i 的上近似集 $\underline{B}U_i$.

Step 3: 若存在中心 C_i', 使得 X_j 到 C_i' 的距离和 X_j 到 C_i 的距离之差小于 Δ, 则将 X_j 归到簇 U_i' 的上近似集 $\overline{B}U_i'$, 否则, 将 X_j 归到 U_i 的下近似集 $\underline{B}U_i$.

Step 4: 计算 M_{ij} 并迭代中心, 如下所示:

$$C_i = \begin{cases} W_{\text{low}} \times \sum\limits_{X_j \in \underline{B}U_i} M_{ij} \times X_j + \\ W_{\text{up}} \times \sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} M_{ij}' \times X_j', \\ \qquad \underline{B}U_i \neq \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) \neq \varnothing; \\ \sum\limits_{X_j \in \underline{B}U_i} M_{ij} \times X_j, \\ \qquad \underline{B}U_i \neq \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) = \varnothing; \\ \sum\limits_{X_j \in (\overline{B}U_i - \underline{B}U_i)} M_{ij}' \times X_j', \\ \qquad \underline{B}U_i = \varnothing \bigwedge (\overline{B}U_i - \underline{B}U_i) \neq \varnothing. \end{cases}$$

Step 5: 重复 Step 2 ∼ Step 4, 直到没有新的数据对象.

算法分别对粗糙 K 均值聚类产生的 3 种情况都插入了簇内不平衡度 M_{ij}, 这实际上是对每次迭代中心移动的步长的改进, 簇内重要度使得每次迭代的中心移动变小, 趋于稳定, 更易收敛.

3　实验仿真分析

为了验证算法的处理效果, 本文选取 UCI 数据库中 Iris 和 Wine 两个数据集进行分析, 两个数据均有较为明确的分类决策, 有利于最终聚类结果的精度分析. 两数据集特征如表 1 所示.

表 1　Iris 和 Wine 数据集特征

数据名称	分类个数	条件属性个数	决策属性个数	数据对象个数
Iris	3	4	1	150
Wine	3	13	1	178

3.1　Iris 数据集仿真分析

对 Iris 数据集进行聚类采用统一的初始聚类中心, 表 2 给出了初始中心具体数值. 本文分别采用粗

糙 K-means 算法、模糊 K-means 算法、粗糙模糊 K-means 算法[13]、模糊粗糙 K-means 算法[7]和本文设计的基于簇内不平衡度量的粗糙 K-means 算法对 Iris 数据集进行聚类, 表 3 是几种算法聚类结果的精度对比. 表 3 中, 几种算法使用了相同的粗糙权值, 即下近似权值 $W_{\text{low}} = 0.9$, 边界权值 $W_{\text{up}} = 0.1$. 其中精度指的是对比原数据集的决策属性值, 被正确聚类的对象所占对象总数的百分比. 由表 3 可见, 在同一初始情况下, 几种算法的精度逐步提升. 当改进算法加入模糊隶属度后, 算法的精度有了大幅度的提升, 这表明经典粗糙 K 均值算法单纯使用粗糙权值并不能使聚类结果簇间产生分离, 加入模糊隶属度能够产生更好的簇间聚类效果. 而在粗糙 K 均值算法中加入簇内不平衡度量, 相比模糊粗糙 K 均值算法和粗糙模糊 K 均值算法而言, 精度提高 1.4%, 表明簇内不平衡度量不仅能使靠近簇中心的对象更加紧凑, 而且使得簇间区分更明显.

表 2 Iris 数据集初始中心

聚类中心号	条件属性号			
	1	2	3	4
1	6.74	3.04	5.61	2.04
2	5.86	2.75	4.33	1.38
3	5	3.42	1.48	0.25

为了更直观地比较聚类算法的结果, 实验通过主成分分析 (PCA) 方法将 4 维数据映射到 2 维平面上. PCA 方法可以有效地找出数据中最“主要”的元素和结构, 去除噪音和冗余, 将原有的复杂数据降维, 揭示隐藏在复杂数据背后的简单结构.

图 2 ~ 图 6 是 Iris 数据集仿真结果降维后的分布图, 分别使用点、圈、星来表示簇 1、簇 2、簇 3, 并以黑色十字标注中心, 其中虚线部分标注的是聚类后决策属性号与簇号不符的对象.

由图 2 可见, 3 个簇都是线性分布的, 其中簇 1 与簇 2 和簇 3 距离较远, 簇 2 与簇 3 距离较近. Iris 数据集包含的数据对象 (簇 1) 被明显地分隔出来, 而簇 2 和簇 3 的簇边界是交织的、模糊不清的、较难区分的. 对比表 3 的精度可以发现, 聚类算法的效果是逐渐提高

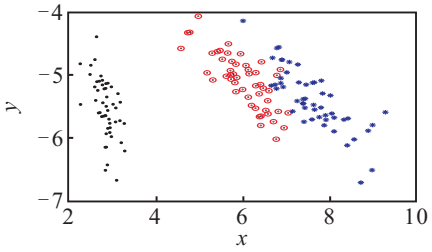

图 2 Iris 数据点的决策分布

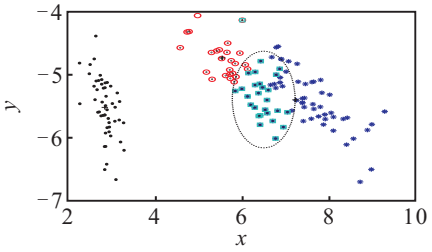

图 3 粗糙 K-means 算法 Iris 数据聚类分布

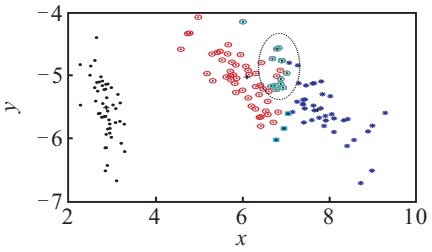

图 4 模糊 K-means 算法 Iris 数据聚类分布

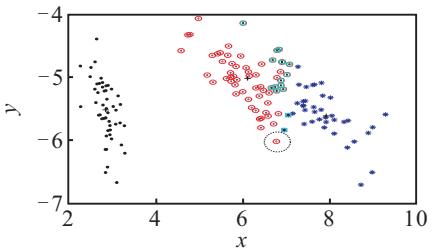

图 5 粗糙模糊 K-means 算法 Iris 数据聚类分布

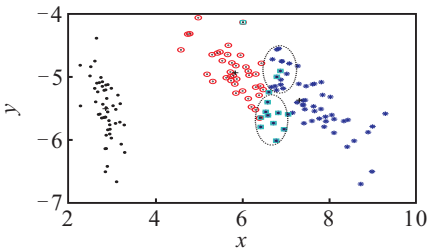

图 6 本文改进的粗糙 K-means 算法 Iris 数据聚类分布

表 3 Iris 数据集的不同聚类算法精度对比

算 法	聚类个数	模糊指数	隶属度判断差值	决策距离判断差值	下近似权值	边界权值	健康函数判断差值	精度/%
粗糙 K 均值算法	3			0.01	0.9	0.1	0.1	87.11
模糊 K 均值算法	3	2				0.1		90.28
粗糙模糊 K 均值算法	3	2	0.02		0.9	0.1	0.1	90.72
模糊粗糙 K 均值算法	3	2		0.01			0.1	89.79
基于簇内不平衡度量的粗糙 K 均值算法	3			0.01	0.9	0.1	0.1	92.13

表 4　Wine 数据集初始中心

聚类中心号	条件属性号												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	12.96	2.51	2.4	19.78	104	2.14	1.62	0.39	1.52	5.66	0.89	2.4	738
2	13.81	1.89	2.44	16.92	105	2.86	3.02	0.29	1.91	5.76	1.08	3.1	1 210
3	12.52	2.47	2.29	20.8	92.44	2.07	1.77	0.39	1.45	4.11	0.94	2.49	459

表 5　Wine 数据集的不同聚类算法精度对比

算　法	聚类个数	模糊指数	隶属度判断差值	决策距离判断差值	下近似权值	边界权值	健康函数判断差值	精度/%
粗糙 K 均值算法	3			0.01	0.9	0.1	0.1	64.62
模糊 K 均值算法	3	1.5				0.1		72.11
粗糙模糊 K 均值算法	3	1.5	0.01		0.9	0.1	0.1	72.11
模糊粗糙 K 均值算法	3	1.5		0.01			0.1	72.37
基于簇内不平衡度量的粗糙 K 均值算法	3			0.01	0.9	0.1	0.1	72.37

的, 从分布图上反映出来的变化是簇 2 和簇 3 的边界点归属情况. 粗糙 K 均值算法的聚类结果将簇 1 和簇 2 分得很清楚, 簇中几乎没有被错误聚类的点. 但是在簇 3 中, 相当一部分本该属于簇 2 的点划入了簇 3, 如图 3 虚线框出的部分, 这部分对象大幅度地降低了簇 3 的聚类精度, 从而影响了整体效果. 从图 4 和图 5 反映出, 加入了隶属度的 K 均值算法得到了更好的聚类效果. 图 4 中, 簇 1 和簇 3 被清楚地分隔出来, 簇 3 中只有 3 个错误聚类点, 而簇 2 中有很多本属于簇 3 的点, 如虚线圈出的部分. 但是, 图 4 错误聚类点数量明显少于图 3, 这表明 Iris 数据集对簇间差异度较为敏感. 图 5 在模糊集方法的基础上加入了粗糙权值, 但是收效甚微, 只得到下方一个点的改善. 由图 6 可见, 簇 2 和簇 3 的边界不再如前述的几种方法一样, 在某一个簇中含有大量的错误聚类点, 基于簇内不平衡度的粗糙 K 均值算法使得簇 2 和簇 3 的边界各有牺牲, 两个簇中各含有一些错误聚类点, 但错误点数量少于前几种方法, 聚类点在分布上更加均衡, 精度上提高了约 1.4%. 这表明聚类对象使用不同权值, 可以使得聚类结果簇内更紧凑、簇间区分更明显.

3.2　Wine 数据集仿真分析

本文对 Wine 数据同样作了一组对比, 使用的算法与 Iris 数据集相同, 初始中心点如表 4 所示. 表 5 为 Wine 数据集聚类不同聚类算法的精度对比. 与 Iris 数据相同, 表 5 中 Wine 数据集聚类取统一粗糙权值, 且精度指的是对比原数据集的决策属性值被正确聚类的对象所占对象总数的百分比. 可以发现, 在加入隶属度后, 粗糙 K 均值聚类的效果有了大幅度的提高, 这点与 Iris 数据集相同. 虽然加入簇内不平衡度后算法没有较粗糙模糊 K 均值和模糊粗糙 K 均值算法有明显的改善, 但也获得了最高的聚类精度. 对 Wine 数据集的仿真结果也能够反映出, 基于簇内不平衡度的粗糙 K 均值算法能够使得聚类边界更加清晰, 簇间区分更加明显.

仿真实验对聚类算法的运行时间进行了对比, 结果如表 6 所示. 由表 6 可见, 基于簇内不平衡度量的粗糙 K 均值算法的运行速度并不慢, 虽然算法精度有了提高, 但是运行时间没有太大牺牲, 这表明基于簇内不平衡度量的粗糙 K 均值算法的性能是较为高效的.

表 6　Iris、Wine数据集不同聚类算法运行时间对比

算　法	运行时间/s	
	Iris	Wine
粗糙 K 均值算法	0.088	0.084
模糊 K 均值算法	0.069	0.166
粗糙模糊 K 均值算法	0.216	0.311
模糊粗糙 K 均值算法	0.109	0.098
基于簇内不平衡度量的粗糙 K 均值算法	0.161	0.198

由以上对比结果可见, 每次聚类的结果总是在加入隶属度的参数后有了大幅度的精度提高, 但这并不能表明聚类对象的簇间关系比簇内关系更重要, 或者说某些使用了簇内关系因素的算法不好. 而簇内不平衡度兼顾了簇内和簇间的因素, 使簇内紧致, 簇间分离, 令改进后的粗糙 K 均值算法有了更好的聚类效果. 本文使用的仿真初始参数均是统一的, 如下近似集权值为 0.9, 边界权值为 0.1, 这些初始参数的影响也是不容忽视的. 从整体上观察, 无论是 Iris 数据集还是 Wine 数据集, 本文设计的方法均得到了更好的聚类效果.

4　结　论

聚类对象在簇内分布不平衡, 对簇聚类贡献度有差异. 针对这一问题, 本文提出一种基于簇内不平衡度的粗糙 K 均值算法, 通过距离衡量分布不均的对象在簇内的权重, 从而使得聚类结果簇内紧致、簇间分离. 通过对两个 UCI 数据进行仿真实验, 对比以往的

粗糙 K 均值和一些改进聚类算法, 表明了本文设计的基于簇内不平衡度的粗糙 K 均值算法有较好的聚类效果.

仿真实验的过程中也反映出一些问题: 并不是簇内最紧凑时, 聚类达到了最佳效果, 当簇内对象聚集时, 可能会在簇间分隔上产生一些牺牲. 另外, 本文讨论的方法也受到一些主观参数的影响, 较为明显的如聚类个数、差异度判断阈值等, 这些参数也与数据集的敏感度相关, 如何改善是下一步的研究方向.

参考文献(References)

[1] Han Jia-wei, Kamber Miche-line. Data mining, concepts and techniques[M]. San Mateo: Morgan Kaufmann Publishers, 2001: 15-22.

[2] 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.

[3] Lingras Pawan, Yan Rui, West Chad. Comparison of conventional and rough K-means clustering[C]. The 9th Int Conf on RSFDGrC. Chongqing, 2003: 130-137.

[4] Peters Georg. Outliers in rough k-means clustering[C]. The 1st Int Conf on PReMI. Kolkata, 2005: 702-707.

[5] Peters Georg. Some refinements of rough k-means clustering[J]. Pattern Recognition, 2006, 39(8): 1481-1491.

[6] Sushmita Mitra, Haider Banka, Witold Pedrycz. Rough fuzzy collaborative clustering[J]. IEEE Trans on Systems, Man and Cybernetics, Part B: Cybernetics, 2006, 36(4): 795-805.

[7] Hu Qing-hua, Yu Da-ren. An improved clustering algorithm for information granulation[C]. The 2nd Int Conf on FSKD. Changsha, 2005: 494-504.

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

[9] 谢娟英, 张琰, 谢维信, 等. 一种新的密度加权粗糙 K-均值聚类算法[J]. 山东大学学报: 理学版, 2010, 45(7): 1-6.
(Xie J Y, Zhang Y, Xie W X, et al. A novel rough K-means clustering algorithm based on the weight of density[J]. J of Shandong University: Natural Science, 2010, 45(7): 1-6.)

[10] 刘兵, 夏士雄, 周勇, 等. 基于样本加权的可能性模糊聚类算法[J]. 电子学报, 2012, 40(2): 371-375.
(Liu B, Xia S X, Zhou Y, et al. A sample-weighted possibilistic fuzzy clustering algorithm[J]. Acta Electronica Sinica, 2012, 40(2): 371-375.)

[11] Lingras Pawan, Peters Georg. Rough clustering[J]. Data Mining and Knowledge Discovery, 2011, 1(1): 64-72.

[12] Bezdek J C. Pattern recognition with fuzzy objective function algorithms[M]. New York: Spriger, 1981: 43-85.

[13] 王丹, 吴孟达. 粗糙模糊 C 均值融合聚类[J]. 国防科技大学学报, 2011, 33(3): 145-150.
(Wang D, Wu M D. Rough fuzzy C-means combination clustering[J]. J of National of Defense University Technology, 2011, 33(3): 145-150.)

(上接第1478页)

[10] MacGregor J, Jackle C. Process monitoring and diagnosis by multiblock PLS methods[J]. AIChE Journal, 1994, 40(5): 826-838.

[11] Zhang Y, Zhou H, Qin S. Decentralized fault diagnosis of large-scale processes using multiblock kernel principal component analysis[J]. Acta Automatic Sinica, 2010, 36(4): 593-597.

[12] Iri M, Aoki K, Shima E, et al. An algorithm for diagnosis of system failures in the chemical process[J]. Computers and Chemical Engineering, 1979, 3(1/4): 489-493.

[13] Vedam H, Venkatasubramanian V. PCA-SDG based process monitoring and fault diagnosis[J]. Control Engineering Practice, 1999, 7: 903-917.

[14] 孙运莲. 基于分块和核参数选择的KPCA研究[D]. 哈尔滨: 哈尔滨工业大学计算机科学与技术学院, 2010.
(Sun Y L. Research on KPCA based on block and kernel parameter selection[D]. Harbin: School of Computer Science and Technology，Harbin Institute of Technology, 2010.)

[15] Tarjan R E. Depth-first search and linear graph algorithm[J]. SIAM J on Computing, 1972, 1(2): 146-160.

[16] Downs D, Vogel E. A plant-wide industrial process control problem[J]. Computers & Chemical Engineering, 1993, 17: 245-255.

[17] 别立波. 基于数据驱动的连续过程故障发现与诊断研究[D]. 沈阳: 沈阳工业大学信息科学与工程学院, 2009.
(Bie L B. Fault detection and diagnosis of continuous process based on data-driven method[D]. Shenyang: School of Information Science and Engineering, Shenyang University of Technology, 2009.)