

# 一种新的粗糙 Leader 聚类算法

殷 钢 苗夺谦 段其国

(同济大学计算机科学与技术系 上海 201804)

**摘要** 聚类是数据挖掘领域重要的研究方向。在众多的聚类算法中,Leader 算法运用很广泛,但 Leader 算法没有考虑到聚类分析中内在的不确定性。对 Leader 算法做了相应改进,加入了粗糙集和粒计算的思想,使其能够处理聚类中固有的不确定性,得到更合理的聚类结果。最后,通过实验证明了该算法的优越性。

**关键词** 聚类,Leader 算法,粗糙集,粒计算

中图分类号 TP301.6 文献标识码 A

## New Rough Leader Clustering Algorithm

YIN Gang MIAO Duo-qian DUAN Qi-guo

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

**Abstract** Clustering is a major research orientation in data mining. Among all the clustering algorithms, leader algorithm is widely used, but it fails to take into consideration the inherent uncertainty involved in clustering analysis. This paper proposed an improved leader algorithm based on rough set and granular computing. The novel leader algorithm can deal with the intrinsic uncertainty in clustering analysis and make the clustering results more reasonable. Finally, the superiority of the new rough Leader algorithm is proved by experimentation.

**Keywords** Clustering, Leader algorithm, Rough set, Granular computing

近 10 年来,数据挖掘<sup>[1]</sup>逐渐成为数据库和人工智能等研究领域的热点。聚类<sup>[2]</sup>是数据挖掘中重要的研究课题之一。在数据挖掘中,聚类分析能作为一个独立的工具来获得数据分布的情况,观察每个簇的特点,集中对特定的某些簇做进一步的分析。此外,聚类分析还可以作为其它算法的预处理步骤,这些算法再在生成的簇上进行处理。

Leader 聚类算法<sup>[3-5]</sup>在数据挖掘和 Web 挖掘中使用广泛。其特点是只需要扫描一遍数据集,因而可以节省大量的运行时间和内存空间,尤其适合处理大数据集。此外,Leader 算法不需要指定簇的个数且对孤立点具有鲁棒性。然而 Leader 算法也存在着一些缺点,例如没有考虑到聚类分析中固有的不确定性,以及样本在划分时的含糊性,且聚类结果受样本排列顺序的影响较大。

本文在现有研究基础之上提出了一种改进的粗糙 Leader 聚类算法,通过引入软计算的思想避免聚类时的硬划分,用粗糙集<sup>[6,7]</sup>中的边界区域来刻画聚类中内在的不确定性。同时结合粒计算<sup>[8,9]</sup>的相关思想,通过阈值的调整不断细化边界区域中样本点聚类的粒度,从而得到更好的聚类结果,并发现数据集中存在的孤立点。实验验证了该算法相比经典 Leader 算法的优越性。

## 1 背景知识

### 1.1 聚类算法

所谓聚类,就是将物理或抽象对象的集合聚成由类似的对象构成的多个类或簇的过程。由聚类生成的簇是一组数据对象的集合,同一簇中的对象尽可能相似,而不同簇中的对象尽可能相异。通过聚类,人们能够识别密集和稀疏的区域,发现全局的分布模式和数据属性之间有趣的相互关系。

大体上,主要的聚类算法可以划分为如下几类。

(1) 基于划分的方法:给定要构建的划分的数目  $k$ ,将数据划分为  $k$  个组,每个组至少包含一个对象;每个对象必须属于且只属于一个组。

(2) 基于层次的方法:对给定数据对象集合进行层次的分解。根据层次的分解如何形成,层次的方法又可以分为自底向上的方法和自顶向下的方法。

(3) 基于密度的方法:只要临近区域的密度(对象或数据点的数目)超过某个阈值,就继续聚类。这样的方法可以用来过滤“噪声”孤立点数据,发现任意形状的簇。

(4) 基于网格的方法:把对象空间量化为有限数目的单元,形成一个网格结构。所有的聚类操作都在这个网格结构(即量化的空间)上进行。

(5) 基于模型的方法:为每个簇假定一个模型,寻找数据对给定模型的最佳拟合。

### 1.2 粗糙集理论

波兰数学家 Pawlak 于 1982 年首次提出了粗糙集理论。粗糙集理论是一种刻画不完整性和不确定性的数学工具,能

到稿日期:2008-06-03 本文受国家自然科学基金资助项目(60475019),国家自然科学基金资助项目(60775036),2006 年博士学科点专项科研基金(20060247039)资助。

殷 钢 男,硕士,主要研究方向为数据挖掘、Web 挖掘,E-mail:yg020840@hotmail.com;苗夺谦 男,教授,博士生导师,主要研究方向为粗糙集、数据挖掘;段其国 男,博士,主要研究方向为 Web 挖掘、Web 智能。

有效地分析和处理不精确、不一致和不完整的信息,并从中发现隐含的知识,揭示潜在的规律。

令知识库  $K = (U, R)$ , 对每个概念  $X(X \subseteq U)$ , 当  $X$  能用属性子集  $R$  确切地描述时, 称  $X$  是  $R$  可定义的, 否则称  $X$  是  $R$  不可定义的。  $R$  可定义集也称作  $R$  精确集,  $R$  不可定义集也称作  $R$  粗糙集。这时, 集合  $X$  只能通过逼近的方式来刻画, 称包含于  $X$  中的最大可定义集为  $X$  的下近似(记为  $\underline{R}(X)$ ), 包含  $X$  的最小可定义集为  $X$  的上近似(记为  $\overline{R}(X)$ )。

集合  $X$  关于  $R$  的下近似和上近似分别定义为:

$$\underline{R}(X) = \{x | x \in U \text{ 且 } [x]_R \subseteq X\}$$

$$\overline{R}(X) = \{x | x \in U \text{ 且 } [x]_R \cap X \neq \emptyset\}$$

其中  $[x]_R$  表示由  $x$  决定的等价类。

集合  $X$  的  $R$  边界域定义为:

$$BN_R(X) = \overline{R}(X) - \underline{R}(X)$$

由以上定义可知,  $\underline{R}(X)$  实际上是由那些根据已有知识判断肯定属于  $X$  的对象所组成的最大的集合,  $\overline{R}(X)$  是那些可能属于概念  $X$  的对象组成的最小集合,  $BN_R(X)$  由不能肯定分类到这个概念  $X$  或其补集  $\bar{X}$  中的所有元素组成。

## 2 Leader 聚类算法及其改进

### 2.1 Leader 聚类算法

Leader 聚类算法属于层次的聚类方法, 在该算法中每一个簇由一个 Leader 表示。Leader 是一个区间, 该区间的端点是两个向量, 分别表示了簇的下界与上界。簇  $C$  的下界向量与上界向量定义如下:

$$\min(C) = \{f_1 f_2 \cdots f_m\}, \text{ where } f_i = \min_{x \in C} (F_i(x))$$

$$\max(C) = \{f_1 f_2 \cdots f_m\}, \text{ where } f_i = \max_{x \in C} (F_i(x))$$

其中  $m$  是向量的维度,  $F_i(x)$  表示向量  $x$  第  $i$  维的值。

Leader 聚类算法的基本步骤如下:

INPUT: 数据集 DATASET, 阈值 Thre

OUTPUT: 若干个簇及其 Leader:  $C_1(L_1), C_2(L_2), \dots$

Step1 用第一个数据对象  $v_1$  构造初始的簇  $C_1$ , 并令  $\min(C_1) = \max(C_1) = v_1$ ;

Step2 如果 DATASET 为空则算法结束, 否则转 Step3;

Step3 对 DATASET 中任一数据对象  $v_i$ , 计算  $v_i$  到所有现有 Leader 的距离  $D(v_i, L_j), j \in 1, 2, \dots$ , 找出与  $v_i$  距离最近的  $L_k$ ;

Step4 IF  $D(v_i, L_k) \leq \text{Thre}$ , 则将  $v_i$  分配到  $L_k$  所代表的簇  $C_k$  中, 并更新  $\min(C_k)$  和  $\max(C_k)$ ;

Step5 IF  $D(v_i, L_k) > \text{Thre}$ , 则用  $v_i$  构造新的簇  $C_m$ , 并令  $\min(C_m) = \max(C_m) = v_i$ ;

Step6 转 Step2。

其中  $D(v_i, L_k)$  表示数据对象  $v_i$  与簇  $k$  的 Leader 之间的距离, 距离的计算公式如下:

$$D(x, y) = \sum_{i=1}^m [(F_i(a) - F_i(c))^2 + (F_i(b) - F_i(d))^2]$$

这里  $x = [a, b]$  与  $y = [c, d]$  表示两个区间,  $a, b, c, d$  均为  $m$  维向量,  $F_i(x)$  表示向量  $x$  第  $i$  维的值。

### 2.2 改进的粗糙 Leader 聚类算法

经典 Leader 算法的优点是时间复杂度较低、聚类时无需指定簇的个数。但该算法也存在着一些缺点, 例如在聚类时

使用单阈值来划分样本, 这样的硬划分消除了含糊性。但样本划分时的含糊性是聚类分析所固有的不确定性, 忽视了它, 聚类的结果是不合理的。此外, 经典 Leader 算法聚类的结果受样本排列顺序的影响很大。

由于粗糙集在处理不确定性问题上的强大能力, 本文借鉴了粗糙集中上、下近似的思想, 在经典 Leader 算法的基础上增加一个阈值, 使用双阈值来控制样本的划分。这两个阈值起到了类似于粗糙集中划分上、下近似的作用, 因而分别称为上近似阈值和下近似阈值。如果样本到某个簇的距离小于下近似阈值, 那么样本肯定属于该簇; 而如果这个距离大于下近似阈值但小于上近似阈值, 则样本该如何划分是含糊的、不确定的。介于上、下近似阈值之间的这段中间区域便等同于粗糙集中上、下近似之差, 即边界区域。

边界区域中的样本由于包含的划分信息较含糊, 如果硬划分, 不但自身较易错分, 还会影响到 Leader 的区间表示, 进而影响到后续样本的聚类, 这也是经典 Leader 算法受样本聚类顺序影响较大的原因。本文提出的改进算法对边界区域中的样本进行缓存, 等到数据集中其它的包含了较充分的划分信息的样本聚类结束之后, Leader 的区间表示较准确时再对缓存的样本进行迭代聚类, 并通过调整上、下近似阈值不断缩小边界区域, 以控制算法结束。

另一方面, 对上、下近似阈值的调整也包含了粒计算的相关思想, 通过减小上近似阈值可以细化边界区域中样本聚类的粒度, 构造出更多、更小的粒。由于边界区域是类别间产生重叠的部分, 为了得到更合理的聚类结果, 应该将其中的样本构造成新的、细粒度的簇而不是硬划分到已有的簇中。而粒度不断细化的结果就是构造出一些包含了很少样本甚至是单个样本的簇, 包含于其中的就是数据集中的孤立点, 因此该改进算法也有助于发现数据集中的孤立点。

改进的粗糙 Leader 聚类算法的基本步骤如下:

INPUT: 数据集 DATASET, 两个阈值  $L\_Thre, U\_Thre$ , 迭代次数  $n$

OUTPUT: 若干个簇及其 Leader:  $C_1(L_1), C_2(L_2), \dots$

Step1 用第一个数据对象  $v_1$  构造初始的簇  $C_1$ , 并令  $\min(C_1) = \max(C_1) = v_1$ ;

Step2 如果 DATASET 为空则转 Step8, 否则转 Step3;

Step3 对 DATASET 中任一数据对象  $v_i$ , 计算  $v_i$  到所有现有 Leader 的距离  $D(v_i, L_j), j \in 1, 2, \dots$ , 找出与  $v_i$  距离最近的  $L_k$ ;

Step4 IF  $D(v_i, L_k) \leq L\_Thre$ , 则将  $v_i$  分配到  $L_k$  所代表的簇  $C_k$  中, 并更新  $\min(C_k)$  和  $\max(C_k)$ ;

Step5 IF  $D(v_i, L_k) > U\_Thre$ , 则用  $v_i$  构造新的簇  $C_m$ , 并令  $\min(C_m) = \max(C_m) = v_i$ ;

Step6 IF  $L\_Thre < D(v_i, L_k) \leq U\_Thre$ , 则将  $v_i$  缓存到临时数据集 TEMP 中;

Step7 转 Step2;

Step8 如果 TEMP 为空则算法结束, 否则转 Step9;

Step9 将 TEMP 中的数据对象放回 DATASET 中, 清空 TEMP;

Step10  $U\_Thre = U\_Thre - \Delta$ ;

Step11 转 Step2。

其中  $\Delta = (U\_Thre - L\_Thre) / n$

### 3 实验分析

#### 3.1 实验数据

实验采用的数据来自于 UCI Machine Learning Repository<sup>[10]</sup> 的两个数据集 Iris 和 WDBC (Wisconsin Diagnostic Breast Cancer)。其中 Iris 数据集包含 150 个样本,分为 setosa, versicolor 和 virginica 3 类,每一类各包含 50 个样本,每个样本由 4 个属性描述;WDBC 数据集包含 569 个样本,分为 benign 和 malignant 两类, malignant 包含 212 个样本, benign 包含 357 个样本,每个样本由 30 个属性描述。

#### 3.2 评价标准

Entropy(熵)和 Precision(精度)是两种常用的聚类评价标准<sup>[11,12]</sup>,通过比较聚类结果和标准分类结果来衡量聚类的好坏。

Entropy 衡量的是聚类的纯度,值越小聚类越合理。假设数据集的样本总数为  $N$ ,聚类得到的簇为  $\{A_1, \dots, A_m\}$ ,而标准分类得到的类别为  $\{B_1, \dots, B_n\}$ ,则 Entropy 的定义如下:

$$Entropy = \sum_{i=1}^m \frac{|A_i|}{N} \left( - \sum_{j=1}^n p_{ij} \log p_{ij} \right)$$

其中  $p_{ij}$  表示簇  $A_i$  的样本中其标准分类结果属于类别  $B_j$  的比例:

$$p_{ij} = \frac{|\{x | x \in A_i \text{ 且 } x \in B_j\}|}{|A_i|}$$

Precision 是假设一个簇的类别标识就是占最多数的那个标准类别的标识,其衡量的是聚类的精度,值越大聚类越合理。同样假设数据集的样本总数为  $N$ ,聚类得到的簇为  $\{A_1, \dots, A_m\}$ ,而标准分类得到的类别为  $\{B_1, \dots, B_n\}$ ,则 Precision 的定义如下:

$$Precision = \sum_{i=1}^m \frac{|A_i|}{N} Precision(A_i)$$

其中  $Precision(A_i)$  表示簇  $A_i$  的样本中占最多数的那个标准类别所占的比例:

$$Precision(A_i) = \frac{\max(|\{x | x \in A_i \text{ 且 } x \in B_j\}|)}{|A_i|}$$

#### 3.3 Iris 实验结果

表 1 所列的是 Iris 数据集上经典 Leader 算法取不同阈值所得到的实验结果。实验目的是为了比较阈值的合理性。

表 1 Iris 实验的阈值选取

阈值	1	3	5	10	30
簇	26	11	6	3	2
熵	0.07	0.20	0.24	0.46	0.67
精度	0.97	0.92	0.94	0.87	0.67

综合考虑簇的个数及聚类合理性的两项评价指标,应选取 5 作为经典 Leader 算法在 Iris 数据集上的阈值。相应地,改进的 Leader 算法的上、下近似阈值选取为 2.5/7.5,即在经典算法阈值的基础上加减 50%,从而形成一段中间区域,以捕获聚类中内在的不确定性,避免对样本的硬划分。

图 1 和图 2 分别为 Iris 数据集上经典 Leader 算法和改进的粗糙 Leader 算法的 5 次对比实验结果。

图 1 的实验结果显示,5 次试验中,改进的粗糙 Leader 算法都具有更小的 Entropy 值,这说明改进算法聚类的纯度更高,聚类结果更合理。

图 2 的实验结果显示,5 次试验中,改进的粗糙 Leader 算法都具有更高的 Precision 值,这说明改进算法聚类的精度更

高,从另一个侧面说明了聚类结果更合理。

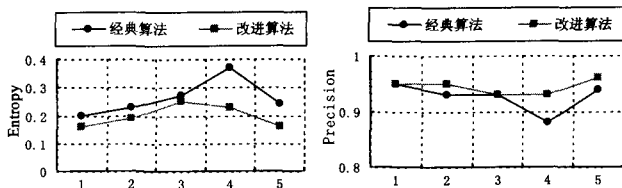


图 1 Iris 实验的 Entropy 对比 (值越小越好)

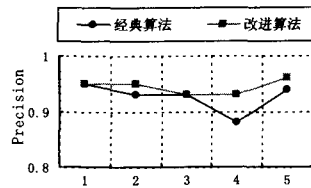


图 2 Iris 实验的 Precision 对比 (值越大越好)

由 Entropy 和 Precision 两项评价指标的对比可以看出改进的粗糙 Leader 算法的优越性。由于它将边界区域中的样本进行缓存,与数据集中包含了较确定的划分信息的样本进行分离,等到该次聚类结束之后,Leader 的区间表示较准确时,再对缓存的样本进行迭代聚类,从而使聚类结果更加合理。

另一方面,由图 1 和图 2 中曲线的平滑程度可以看出,经典 Leader 算法的聚类结果受样本排列顺序的影响较大,曲线上波动较明显,而改进的粗糙 Leader 算法的两条曲线都较平滑,上下波动较小,这说明其实验结果受样本排列顺序的影响相对较小。这也得益于改进算法对边界区域中样本点的滞后处理。

#### 3.4 WDBC 实验结果

表 2 所列的是 WDBC 数据集上经典 Leader 算法取不同阈值所得到的实验结果。实验目的是为了比较阈值的合理性(注:下表中阈值的单位为 100,000)。

表 2 WDBC 实验的阈值选取

阈值	1	10	20	30	50	100
簇	33	9	6	5	4	2
熵	0.28	0.51	0.47	0.63	0.63	0.86
精度	0.88	0.83	0.88	0.81	0.81	0.69

综合考虑簇的个数及聚类合理性的两项评价指标,应选取 2,000,000 作为经典 Leader 算法在 WDBC 数据集上的阈值。相应地,改进的 Leader 算法的上、下近似阈值选取为 1,000,000, / 3,000,000,即在经典算法阈值的基础上加减 50%,从而形成一段中间区域,上下近似阈值之间的这段中间区域就对应了粗糙集理论中的边界区域。

图 3 和图 4 分别为 WDBC 数据集上经典 Leader 算法和改进的粗糙 Leader 算法的 5 次对比实验结果。

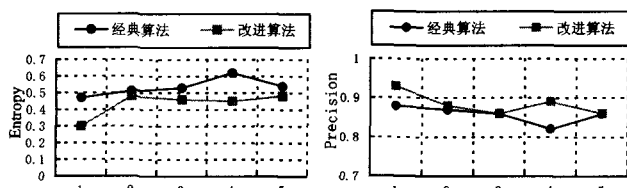


图 3 WDBC 实验的 Entropy 对比 (值越小越好)

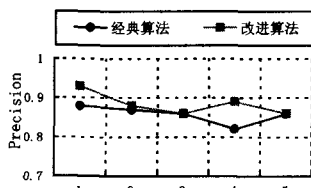


图 4 WDBC 实验的 Precision 对比 (值越大越好)

图 3 的实验结果显示,5 次试验中,改进的粗糙 Leader 算法都具有更小的 Entropy 值,这说明改进算法聚类的纯度更高,聚类结果更合理。

图 4 的实验结果显示,5 次试验中,改进的粗糙 Leader 算法都具有更高的 Precision 值,这说明改进算法聚类的精度更

(下转第 219 页)

了 CTL \* K 的模型检查算法,其核心思想就是将 CTL \* K 公式的检查问题转化为 CTL \* 公式的模型检查问题。这种算法不需要显式地构建状态空间,可以使检查的系统规模得以大幅度提高。并且将算法编码后,容易集成到 NuSMV 模型检查器,从而使能够对 CTL 进行检查的 NuSMV 也可以对 CTL \* K 进行检查。

### 参考文献

[1] Memillan K L. Symbolic model checking :  $10^{20}$  states and beyond[J]. Information and Computation, 1992, 98(2): 142-170

[2] Holzmann G. Design and Validation of Computer Protocols [M]. Prentice Hall, 1990

[3] Holzmann G. The model checker spin[J]. IEEE Trans. on Software Engineering, 1997, 23(5): 279-295

[4] van der Hoek W, Wooldridge M. Model checking knowledge and time[C] // Stefan Leue C C, ed. Proc. of the 9th Int'l Spin. Workshop on Model Checking of Software. Berlin: Springer-Verlag, 2002: 1-16

[5] 吴立军, 苏开乐. 多智能体系统时态认知规范的模型检测算法[J]. 软件学报, 2004, 15 (7): 1012-1020

[6] 吴立军, 苏开乐, 等. 多主体系统时态认知规范的“On the Fly”模型检测算法研究[J]. 计算机研究与发展, 2006, 43 (8): 1417-1424

[7] Emerson E A, Lei Chin - laung. Modalities for model checking (extended abstract): Branching time strikes back[C] // Proceedings of the 12th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. New York, 1985: 84-96

[8] Clarke E M, Emerson E A. Design and synthesis of synchroniza-

tion skeletons using branching-time temporal logic[C] // Proceedings of Logic of Programs Workshop. Yorktown Heights, New York, 1981: 52-71

[9] Grumberg O, Clarke E M, Peled D A. Model Checking. Cambridge[M] MA: The MIT Press, 2000

[10] Schnoebelen P h. The Complexity of Temporal Logic Model Checking[J]. Advances in Modal Logic, 2002, 4: 1-44

[11] Su Kaile, LüGuanfeng, Chen Qingliang. Knowledge structure approach to verification of authentication protocols[J]. Science in China (Series E), 2005, 35 (4): 337-351

[12] 苏开乐, 骆翔宇, 吕关锋. 符号化模型检测 CTL \* [J]. 计算机学报, 2005, 28(11): 1798-1805

[13] Chaum D. The dining cryptographers problem : Unconditional sender and recipient untraceability[J]. Journal of Cryptology, 1988, 1(1): 65-75

[14] van der Meyden R, Su Kaile. Symbolic model checking the knowledge of the dining cryptographers[C] // Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04). Washington, DC, USA, IEEE Computer Society, 2004: 280-291

[15] Kacprzak M, Lomuscio A, Niewiadomski A, et al. Comparing BDD and SAT based techniques for model checking Chaum's dining cryptographers protocol[J]. Fundamenta Informaticae, 2006

[16] Climatti A, Clarke E M, Giunchiglia E. NuSMV Version 2: An OpenSource tool for symbolic model checking[C] // Proceedings of the International Conference on Computer-Aided Verification (CAV 2002). Copenhagen, Denmark, 2002: 359-364

(上接第 205 页)

高,从另一个侧面说明了聚类结果更合理。

WDBC 数据集上的实验结果同样显示出了改进的粗糙 Leader 算法的优越性,聚类的纯度和精度都更高,而且聚类结果受样本排列顺序的影响相对较小。

**结束语** 聚类分析是数据挖掘领域一个非常活跃的研究课题。经典的 Leader 聚类算法有很多优点,例如只需扫描一遍数据集且无需指定簇的个数。然而 Leader 算法也存在一些缺点,例如没有考虑到聚类分析中固有的不确定性,聚类的结果受样本排列顺序的影响较大。

本文在现有研究基础之上提出了一种改进的基于粗糙集和粒计算思想的 Leader 聚类算法。该算法使用粗糙集中上、下近似之差,即边界区域来捕捉聚类分析中内在的不确定性,对边界区域中的样本进行迭代聚类,并通过调整上、下近似的阈值对边界区域进行细分,细化其聚类的粒度。实验验证了改进的粗糙 Leader 算法相对于经典的 Leader 算法的优越性,不但聚类结果更加合理,且不易受样本排列顺序的影响。

### 参考文献

[1] Han Jiawei, Kamber M. Data Mining Concepts and Techniques [M]. 范明, 孟小峰, 译. 北京: 机械工业出版社, 2001

[2] Theodoridis S, Koutroumbas K. Pattern Recognition[M]. 李晶皎, 译. 北京: 电子工业出版社, 2004

[3] Spath H. Cluster Analysis Algorithms for Data Reduction and Classification of Objects[M]. Chichester, Ellis Horwood Limited, 1980

[4] Gowda K C, Diday E. Symbolic Clustering Using a New Dissimilarity Measure[J]. Pattern Recognition, 1991, 24(6): 567-578

[5] Asharaf S, Murty M N, Shevade S K. Rough Set Based Incremental Clustering of Interval Data[J]. Pattern Recognition Letters, 2006, 27(6): 515-519

[6] Pawlak Z. Rough Sets [J]. International Journal of Computer and Information Sciences, 1982, 11(2): 341-356

[7] 王珏, 苗夺谦, 等. 关于 Rough Set 理论与应用的综述[J]. 模式识别与人工智能, 1996, 9(4): 337-344

[8] Lin T Y, Zadeh L A. Special Issue on Granular Computing and Data Mining[J]. International Journal of Intelligent Systems, 2004, 19(7): 565-566

[9] 苗夺谦, 王国胤, 姚一豫, 等. 粒计算: 过去、现在和展望[M]. 北京: 科学出版社, 2007

[10] Blake C L, Merz C J. UCI Repository of Machine Learning Database[EB/OL]. 2007. <http://archive.ics.uci.edu/ml>

[11] Tao L, Zheng C. An Evaluation on Feature Selection for Text Clustering[C] // Proceedings of the 20th International Conference on Machine Learning. Washington D. C., 2003

[12] 刘涛, 陈正. 一种高效的用于文本聚类的无监督特征选择算法[J]. 计算机研究与发展, 2005, 42(3): 381-386