

基于差别矩阵的不一致决策表规则获取算法

钱文彬¹ 杨炳儒¹ 徐章艳² 谢永红¹

(北京科技大学计算机与通信工程学院 北京 100083)¹

(广西师范大学计算机科学与信息工程学院 桂林 541004)²

摘要 针对传统基于差别矩阵的不一致决策表规则获取算法效率不理想的情况,提出了一种快速的基于差别矩阵的规则获取算法。算法首先引入简化决策表思想,删除决策表中可能存在的许多重复对象;然后基于简化决策表构造不同决策类之间的子差别矩阵,以有效地解决对象分布的非平衡性问题和缩小算法的求解空间;且采用启发式向后贪心搜索策略求解相对最小属性约简;并根据规则可信度获取有效的决策规则,可信度可动态设置,使算法具有较好的适应性。最后通过算例分析和实验比较验证了算法能获取有效的决策规则。

关键词 粗糙集理论,不一致决策表,属性约简,规则获取

中图法分类号 TP18 文献标识码 A

Rule Extraction Algorithm Based on Discernibility Matrix in Inconsistent Decision Table

QIAN Wen-bin¹ YANG Bing-ru¹ XU Zhang-yan² XIE Yong-hong¹

(School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China)¹

(School of Computer Science and Information Engineering, Guangxi Normal University, Guilin 541004, China)²

Abstract Since the efficiency of traditional rule extraction algorithms based on discernibility matrix in inconsistent decision table is often poor, a quick rule extraction algorithm based on discernibility matrix was proposed to deal with the problem. The definite of simplified decision table is first introduced, and many duplicate objects are deleted in decision table. Then the subsets of discernibility matrix is constructed with respect to different decision classes, which effectively avoids the imbalance of objects and compresses the storage space of algorithm, and adopting the heuristic search strategy with backward greedy to calculate the relative minimal attribute reduction. Some useful decision rules based on reliability are extracted, what's more, the reliability is dynamically given, and the algorithm has good adaptability. Finally, example analysis and experiential results show that the proposed algorithm can exact effective decision rules from inconsistent decision table.

Keywords Rough set theory, Inconsistent decision table, Attribute reduction, Rule extraction

1 引言

粗糙集理论作为一种处理不确定、不精确和不完备信息的有效数学理论工具^[1-4],已广泛应用于知识发现、数据挖掘、机器学习和案例诊断等诸多领域^[5-7]。粗糙集理论的重要目的是从决策表中获取有用的决策规则。如何从决策表中快速地获取有效的规则已是许多研究人员关注的一个热点课题。

现实领域的复杂系统中有些对象往往呈现出不一致性,把含有不一致对象的决策表称为不一致决策表,否则称为一致决策表。近年来针对一致决策表的规则获取的研究取得了显著进步,已有很多学者提出了许多有效的规则获取算法^[8-12]。文献[8]利用分而治之的思想设计了一种高效的规则提取算法;文献[9]利用差别向量的方法对规则进行约简,并在此基础上构造了一种基于模糊粗糙集的分类器,以有效

处理噪音数据;文献[10]采用属性归纳方法提出了一种基于变精度粗糙集模型的多层规则获取算法;文献[11]研究了当决策表中对象动态增加时规则的变化情况,并设计了相应的规则获取算法;文献[12]通过对决策规则的提取质量及计算效率上的改进,提出了一种改进型LEM2规则提取算法。但对于不一致决策表的规则获取研究进展则较为缓慢,且算法的时空复杂度较高^[13-15];文献[13]利用决策表的布尔矩阵设计了规则获取算法;文献[14]基于决策分布函数和差别矩阵给出了规则获取算法;文献[15]利用可辨识向量组构造了一种基于粗糙集理论规则提取方法。上述方法需要构建决策表的布尔矩阵或差别矩阵,且计算过程复杂,导致基于差别矩阵的不一致决策表规则获取算法的计算效率并不理想,特别是当处理大规模数据时将面临瓶颈,为此设计面向不一致决策表的快速规则获取算法具有重要的理论意义和应用价值。

到稿日期:2012-08-22 返修日期:2012-11-18 本文受国家重点基础研究发展计划项目(973计划)(2009CB522701),国家自然科学基金项目(60963008,60875029),科技部创新方法专项项目(2010IM020900)资助。

钱文彬(1984—),男,博士生,主要研究方向为粗糙集理论、数据挖掘, E-mail: qianwenbin1027@126.com; 杨炳儒(1943—),男,教授,博士生导师,主要研究方向为人工智能、数据挖掘、柔性建模; 徐章艳(1972—),男,博士,教授,主要研究方向为粗糙集、模糊集、数据挖掘; 谢永红(1970—),女,博士,副教授,主要研究方向为数据挖掘、计算智能。

差别矩阵作为粗糙集理论中规则获取的重要技术之一,可从决策表中提取出有效的决策规则,但由于差别矩阵所需的存储空间限制了该技术的广泛应用,为此本文提出了一种从不一致决策表中快速获取规则的算法。该算法首先引入简化决策表思想,删除决策表中可能存在的许多重复对象,为构造差别矩阵降低存储空间做预处理,然后对简化的决策表构造不同决策类之间的子差别矩阵,进一步压缩了算法的存储空间,并采用启发式向后贪心搜索策略,求解完备属性约简;算例分析和实验结果进一步验证了该算法能够从不一致决策表中挖掘出有效的决策规则。

2 相关知识

定义 1 设五元组 $S=(U, C, D, V, f)$ 是一个决策表系统(简称决策表),其中 U 表示非空有限的对象集,简称为论域; C 表示非空有限的条件属性集; D 表示非空有限的决策属性集,有时为了便于计算,设定 D 中仅含一个决策属性,同时 $C \cap D = \emptyset$; $V = \bigcup_{a \in C \cup D} V_a$, V_a 是属性 a 的值域, $f: U \times C \cup D \rightarrow V$ 是信息函数,它为每个对象的属性指定属性值,即 $\forall a \in C \cup D, x \in U$, 有 $f(x, a) \in V_a$ 。

定义 2 在决策表 $S=(U, C, D, V, f)$ 中,对于 $\forall B \subseteq C$, 对应的不可区分关系 $IND(B)$ 为: $IND(B) = \{(x, y) \in U \times U \mid \forall b \in B, f(x, b) = f(y, b)\}$ 。

定义 3 在决策表 $S=(U, C, D, V, f)$ 中,对于 $\forall c_i \in C$, $x, y \in U$, 如果存在 $f(x, c_i) = f(y, c_i)$, 但 $f(x, D) \neq f(y, D)$, 则称该决策表为不一致决策表,否则称一致决策表。

定义 4 在不一致决策表 $S=(U, C, D, V, f)$ 中,设 $U/D = \{D_1, D_2, \dots, D_n\}$ 表示论域 U 在 D 下的决策类划分, $U/C = \{C_1, C_2, \dots, C_m\}$ 表示论域 U 在 C 下的条件等价类划分, 则称 $POS_C(D) = \bigcup_{D_i \in U/D} C_- (D_i)$ 为 C 相对于 D 的正区域。

定义 5^[14] 在不一致决策表 $S=(U, C, D, V, f)$ 中, 设 $U/C = \{[x_1']_C, [x_2']_C, \dots, [x_m']_C\}$, 记 $U' = \{x_1', x_2', \dots, x_m'\}$, 设 $POS_C(D) = [x_{i_1}']_C \cup [x_{i_2}']_C \cup \dots \cup [x_{i_t}']_C$, 则 $[x_{i_t}']_C$ 中的对象属于相同决策类; 记 $U'_{pos} = \{x_{i_1}', x_{i_2}', \dots, x_{i_t}'\}$, $U'_{neg} = U' - U'_{pos}$, 则称 $S=(U', C, D, V, f)$ 为简化的不一致决策表。

定义 6 在不一致决策表 $S=(U, C, D, V, f)$ 中, 基于子差别矩阵的元素定义为:

$$m_{ij} = \begin{cases} c_k, & c_k \in C, f(x_i, c_k) \neq f(x_j, c_k) \wedge x_i \in D_r \\ & \wedge x_j \in D_t \wedge r \neq t, x_i, x_j \in U'_{pos} \\ c_k, & c_k \in C, f(x_i, c_k) \neq f(x_j, c_k) \text{ 且 } x_i, x_j \text{ 一个} \\ & \text{在 } U'_{pos}, \text{ 一个在 } U'_{neg} \text{ 中} \\ \emptyset, & \text{else} \end{cases}$$

定义 7 在不一致决策表 $S=(U, C, D, V, f)$ 中, 设 $Core(C)$ 为核属性, 则对于 $\forall c_k \in Core(C)$, 当且仅当子差别矩阵中存在元素 $m_{ij} = \{c_k\}$ 。

定义 8 在不一致决策表 $S=(U, C, D, V, f)$ 中, 设 $Red(C)$ 为简化的不一致决策表的属性约简, 则 $Red(C)$ 满足:

- (1) 对于 $\forall m_{ij} \in M'_{Red(C)}$, 存在 $\forall m_{ij} \in M_C$;
- (2) 对于 $\forall c_k \subseteq Red(C)$, 使得 $M'_{Red(C)} \neq M_C$ 。

定义 9 在不一致决策表 $S=(U, C, D, V, f)$ 中, 设 $U/D = \{D_1, D_2, \dots, D_n\}$ 表示论域 U 在 D 下的决策类划分, $U/C = \{C_1, C_2, \dots, C_m\}$ 表示论域 U 在 C 下的条件等价类划分, 条件等价类将对应一组决策类, 则决策规则定义为: $\wedge f(x, c_i) \rightarrow$

$\vee f(x, d)$, 其中 $f(x, c_i)$ 为规则前件, $f(x, d)$ 为规则后件。

定义 10 在不一致决策表 $S=(U, C, D, V, f)$ 中, 设 $U/D = \{D_1, D_2, \dots, D_n\}$, $U/C = \{C_1, C_2, \dots, C_m\}$, 对于决策规则 $f(x, C_i) \rightarrow f(x, D_j)$, 其对应的可信度度量 $\delta = |C_i \cap D_j| / |C_i|$ 。当 $\delta=1$ 时, 该规则为确定性规则, 否则称为不确定性规则。

3 不一致决策表的规则获取算法

对不一致决策表中重复或冗余的对象进行删除时, 主要工作应先求解不一致决策表在条件属性集上的等价类划分, 然后根据决策属性对不一致决策表进行简化, 并将其分为不一致性对象集和一致性对象集, 传统方法往往是对不一致决策表中的对象进行两两比较, 这种方法较为耗时, 不利于大型决策表的数据处理, 本文利用文献[16]中的基数排序思想简化决策表, 对不一致决策表进行预处理, 剔除不一致决策表中的重复对象。

在简化的不一致决策表中需考虑对象集中各对象的决策属性值的分布情况, 并根据这个情况计算基于决策属性的子差别矩阵。简化的不一致决策表能将许多重复的对象删除, 为此根据决策类构造的子差别矩阵可有效降低算法对内存空间的依赖, 并针对有些约简算法无法求解出完备的属性约简, 本文采用启发式向后贪心搜索策略求解属性约简。基于此思想, 给出了一种快速的基于差别矩阵的不一致决策表规则获取算法。

算法 1 基于差别矩阵的不一致决策表规则获取算法

输入: 不一致决策表 $S=(U, C, D, V, f)$, 可信度 δ ;

输出: 决策规则集 $Rules(C, D)$;

Step1 利用基数排序思想对不一致决策表 $S=(U, C, D, V, f)$ 进行简化, 得到简化的不一致决策表 $S'=(U', C, D, V, f)$;

Step2 计算简化决策表在决策属性 D 上的划分 $U'/D = \{D_1, D_2, \dots, D_r\}$;

// 初始化子差别矩阵:

$\forall M_p = \emptyset (1 \leq p \leq r(r-1)/2)$;

// 对于 $\forall x_i', x_j' \in U'/D$ 构造子差别矩阵:

for($w=1; w < r; w++$)

for($v=w+1; v \leq r; v++$)

{// $M(w, v)$ 中矩阵元素的计算方法如下:

for($t=1; t \leq m; t++$)

if($(f(x_i', c_t) \neq f(x_j', c_t) \wedge x_i \in D_i \wedge x_j \in D_j)$)

$m'_{ij} = m'_{ij} \cup \{c_t\}$;

$M(w, v) = M(w, v) \cup m'_{ij}$;

}

Step3 在构造的子差别矩阵 $M(1, r), M(2, r), \dots, M(r-1, r)$ 中计算每个条件属性所对应的频率 $Ncount(c_i)$, 并按 $Ncount(c_i)$ 对条件属性升序排列;

Step4 // 对每个子差别矩阵 $M(w, v)$ 执行如下操作:

Step4.1 计算子差别矩阵 $M(w, v)$ 中的矩阵元素是否出现单个属性 c_i , 若出现, 则将此条件属性 c_i 存入约简 $Red(C)$ 中;

Step4.2 在子差别矩阵 $M(w, v)$ 中删除含有属性 c_i 的矩阵元素, 若所有子差别矩阵均为空, 则得到约简 $Red(C)$, 算法转至 Step5, 否则, 算法转至 Step3 执行;

Step4.3 在子差别矩阵 $M(w, v)$ 中删除排序后的属性集中的第一个属性;

Step5 在约简 $Red(C)$ 上对决策表进行规则提取, 将大于可信度 δ 的

规则存入规则集 $Rules(C,D)$;

Step6 对约简后相同的规则进行简化,输出简化的规则集 $Rules(C,D)$,算法结束。

算法的复杂度分析:

算法 Step1 的时间复杂度为 $O(|C||U|)$,算法 Step2 的时间复杂度为 $O(|C|(|U'| |U'_{pos}| - \sum_{i=1}^r |D_i|^2))$,算法 Step3 和算法 Step4 的时间复杂度均为 $O(|C|(|U'| |U'_{pos}| - \sum_{i=1}^r |D_i|^2))$,算法 Step5 的时间复杂度为 $O(|C|^2(|U'| |U'_{pos}| - \sum_{i=1}^r |D_i|^2))$,算法 Step6 的时间复杂度为 $O(|Rules(C,D)|^2)$,由此可知算法的时间复杂度为 $O(|C|^2(|U'| |U'_{pos}| - \sum_{i=1}^r |D_i|^2))$;由于仅需构造不一致决策表的子差别矩阵,显然易知,算法的空间复杂度为 $O(|C|(|U'| |U'_{pos}| - \sum_{i=1}^r |D_i|^2))$ 。而文献[11-13]中算法的时间复杂度和空间复杂度分别为 $O(|C|^2|U|^2)$ 和 $O(|C||U|^2)$,由此可知,本文算法要优于已有的同类算法。当不一致决策表中重复的对象较多和对象分布非平衡时,算法的计算效率将更具有较强优势。

4 算例分析和实验比较

4.1 算例分析

为了分析算法的可行性和有效性,下面以表 1 中的不一致决策表 $S=(U,C,D,V,f)$ 为例进行分析,不一致决策表共 15 个对象、4 个条件属性、1 个决策属性,可信度为 0.6。

表 1 不一致决策表 $S=(U,C,D,V,f)$

U	c_1	c_2	c_3	c_4	d
x_1	0	1	2	0	1
x_2	1	0	0	0	1
x_3	1	2	1	1	2
x_4	1	0	0	0	1
x_5	0	2	1	0	2
x_6	2	1	0	2	1
x_7	2	1	0	2	2
x_8	0	2	1	0	2
x_9	0	1	0	1	2
x_{10}	2	1	0	2	1
x_{11}	1	2	1	1	2
x_{12}	2	0	1	2	2
x_{13}	0	1	0	1	1
x_{14}	2	0	1	2	1
x_{15}	0	1	0	1	2

首先根据算法的 Step1 可得剔除重复对象的简化不一致决策表 $S'=(U',C,D,V,f)$,由于 $U/C=\{\{x_1\},\{x_2,x_4\},\{x_3,x_{11}\},\{x_5,x_8\},\{x_6,x_7,x_{10}\},\{x_9,x_{13},x_{15}\},\{x_{12},x_{14}\}\}$,因此 $U'=\{x_1,x_2,x_3,x_5,x_6,x_9,x_{12}\}$, $U'_{pos}=\{x_1,x_2,x_3,x_5\}$, $U'_{neg}=\{x_6,x_9,x_{12}\}$,并根据算法的 Step2 先将 U'_{pos} 中的对象以决策属性值为划分依据,得到决策类的划分: $U'/\{d\}=\{\{x_1,x_2\},\{x_3,x_5\}\}=\{D_1,D_2\}$;并把 U'_{neg} 中的对象归为一类 $\{x_6,x_9,x_{12}\}=\{D_3\}$,再依据决策类求解出所对应的子差别矩阵,分别为 $M(1,2),M(1,3),M(2,3)$,如表 2—表 4 所列。

表 2 子差别矩阵 $M(1,2)$

	x_3	x_5
x_1	$\{c_1,c_2,c_3,c_4\}$	$\{c_2,c_3\}$
x_2	$\{c_2,c_3,c_4\}$	$\{c_1,c_2,c_3\}$

表 3 子差别矩阵 $M(1,3)$

	x_6	x_9	x_{12}
x_1	$\{c_1,c_3,c_4\}$	$\{c_3,c_4\}$	$\{c_1,c_2,c_3,c_4\}$
x_2	$\{c_1,c_2,c_4\}$	$\{c_1,c_2,c_4\}$	$\{c_1,c_3,c_4\}$

表 4 子差别矩阵 $M(2,3)$

	x_6	x_9	x_{12}
x_3	$\{c_1,c_2,c_3,c_4\}$	$\{c_1,c_2,c_3\}$	$\{c_1,c_2,c_4\}$
x_5	$\{c_1,c_2,c_3,c_4\}$	$\{c_2,c_3,c_4\}$	$\{c_1,c_2,c_4\}$

然后根据算法 Step3,统计出子差别矩阵中各个条件属性出现的频率分别为 $Ncount(c_1)=12,Ncount(c_2)=13,Ncount(c_3)=12,Ncount(c_4)=13$;为了保证所求解的属性约简的完备性,算法 Step4 中采用向后贪心启发式搜索策略计算属性约简,依据 $Ncount(c_i)$ 排列的属性顺序为 $\{c_1,c_3,c_2,c_4\}$,可知先在子差别矩阵集中删除属性 $\{c_1\}$,子差别矩阵集中并未出现单个属性的情况,为此需继续删除属性 $\{c_3\}$,子差别矩阵的情况如表 5—表 7 所列,此时矩阵出现了单个属性,分别为 $\{c_2\},\{c_4\}$,此时将属性 $\{c_2,c_4\}$ 存入约简 $Red(C)$ 中,并同时将子差别矩阵中所有含属性 $\{c_2\}$ 或 $\{c_4\}$ 的差别矩阵元素全部删除,最后可知所有子差别矩阵均为空,则得属性约简 $Red(C)=\{c_2,c_4\}$ 。

表 5 子差别矩阵 $M(1,2)$

	x_3	x_5
x_1	$\{c_2,c_4\}$	$\{c_2\}$
x_2	$\{c_2,c_4\}$	$\{c_1,c_2,c_3\}$

表 6 子差别矩阵 $M(1,3)$

	x_6	x_9	x_{12}
x_1	$\{c_4\}$	$\{c_4\}$	$\{c_2,c_4\}$
x_2	$\{c_2,c_4\}$	$\{c_2,c_4\}$	$\{c_4\}$

表 7 子差别矩阵 $M(2,3)$

	x_6	x_9	x_{12}
x_3	$\{c_2,c_4\}$	$\{c_2\}$	$\{c_2,c_4\}$
x_5	$\{c_2,c_4\}$	$\{c_2,c_4\}$	$\{c_2,c_4\}$

最后根据算法的 Step5 和 Step6,在不一致决策表中输出可信度大于 0.6 的规则,并对提取的规则集进行简化,得到简化的规则集 $Rules(C,D)$,如表 8 所列,其中 R_1 和 R_2 是确定性规则, R_3 和 R_4 是不确定性规则,且可知 R_3 和 R_4 的规则可信度为 0.67。由此可知算法能够从不一致决策表中获取有效的决策规则。

表 8 简化后的规则集 $Rules(C,D)$

ID	规则描述	可信度
R_1	$f(c_2,2) \rightarrow f(d,2)$	1
R_2	$f(c_4,0) \rightarrow f(d,1)$	1
R_3	$f(c_2,1) \wedge f(c_4,2) \rightarrow f(d,1)$	0.67
R_4	$f(c_2,1) \wedge f(c_4,1) \rightarrow f(d,2)$	0.67

4.2 实验比较

为进一步验证本文提出的算法与同类算法的性能,我们选取了 UCI 机器学习数据库中的 5 个数据集作为测试数据进行实验,为了便于算法的实验比较,将文献[13-15]中的算法简记为 A 算法,把本文算法简记为 B 算法。硬件配置为: Intel Core2,CPU E7400 2.8GHz 和 2G 内存,在开发平台 Visual Studio 2005 中利用 C++ 编程实现。实验数据集情况如表 9 所列。其中,数据集 Lymphography 的决策类分布 $U/D=\{2,81,61,4\}$,Vote 的决策类分布 $U/D=\{184,116\}$,Vehicle 的

决策类分布 $U/D=\{240,240,240,226\}$, Car 的决策类分布 $U/D=\{1210,384,69,65\}$, Mushroom 的决策类分布 $U/D=\{4208,3916\}$ 。

表 9 算法的实验数据集

数据集	实例数	属性数	类别数
Lymphography	148	18	4
Vote	300	16	2
Vehicle	946	18	4
Car	1728	6	4
Mushroom	8124	22	2

实验从算法的属性约简个数、平均分类精度和计算时间等方面进行比较,其中,算法的平均分类精度采用十折交叉法验证,B算法的规则可信度设置为 0.7,Red 表示属性约简的个数,Acc 表示平均分类精度,实验结果如表 10 所列;算法的执行时间如图 1 所示。

表 10 算法的约简结果和分类精度

数据集	A 算法		B 算法	
	Red	Acc	Red	Acc
Lymphography	5	78.65%	5	79.30%
Vote	9	87.02%	9	89.49%
Vehicle	11	65.62%	10	71.83%
Car	6	93.05%	6	94.54%
Mushroom	6	85.10%	5	85.68%

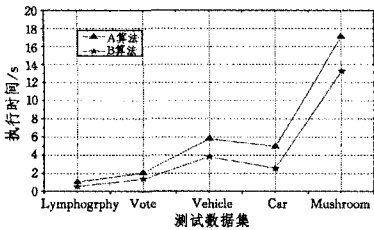


图 1 算法的执行时间比较

从表 10 中可以看出,B算法在属性约简和平均分类精度上均取得了较好结果。与 A 算法相比,B算法在数据集 Vehicle 和 Mushroom 上求解的属性约简个数相对更少,A 算法得到的约简存在冗余,这主要是由两种算法的搜索策略不同所导致的。B算法的平均分类精度优于 A 算法,其主要原因是 B 算法引入了规则的可信度阈值,获取的规则相对更多,如当可信度阈值为 0.7 时,数据集 Vote 获取的规则为 84 条,并随着可信度阈值的增大,满足条件的决策规则将会减少;而 A 算法提取的规则均为确定性规则,即可信度为 100%,这样使得部分合理的决策规则无法被提取,且有些规则的支持度较低,导致分类的精度不理想。

从图 1 中可以发现,B 算法能将不一致决策表中的重复对象删除,且采用了基于决策属性的子差别矩阵,有效降低了算法的搜索空间,提高了算法的计算效率,特别是不一致决策表中对象分布非平衡性时,B 算法的优势更明显,由数据集 Lymphography 和 Car 的计算时间也较好地验证了这一点,特别是数据集 Car 存在较多不一致和重复对象时,简化后的对象个数为 972。而数据集 Mushroom 中的对象都是一致的、不能简化,且对象分布较平衡,使得两种算法的计算性能相当。实验结果表明,本文算法所提取出的决策规则,在保证算法计算效率的同时,能有效地提高规则的分类精度,因此本文

算法是一种有效的规则获取算法。

结束语 规则获取是粗糙集理论的重要研究内容,本文针对差别矩阵在提取决策规则上存在的瓶颈,提出一种基于差别矩阵的不一致决策表快速规则获取算法,算法的优势主要体现在:引入简化决策表删除了不一致决策表中的重复对象,基于属性值的子差别矩阵避免了构造决策类相同的对象所形成的差别矩阵,采用向后贪心算法求解相对最小属性约简。算例分析和实验结果表明,算法能从不一致决策表中快速获取有用的决策规则,这为不一致决策表中的知识发现提供了一种有效的技术手段。

参考文献

[1] Pawlak Z, Grzymala-Busse J, Slowinski R. Rough sets[J]. Communications of the ACM, 1995, 38(11): 89-95

[2] 王国胤, 姚一豫, 于洪. 粗糙集理论与应用研究综述[J]. 计算机学报, 2009, 32(7): 1229-1246

[3] Parthala N M, Shen Qiang, Jensen R. A distance measure approach to exploring the rough set boundary region for attribute reduction[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(3): 305-317, 638

[4] 蒋云良, 杨章显, 刘勇. 不协调信息系统快速属性分布约简方法[J]. 自动化学报, 2012, 38(3): 382-388

[5] 钱进, 孟祥萍, 刘大有, 等. 一种基于粗糙集理论的最简决策规则挖掘算法[J]. 控制与决策, 2007, 22(12): 1368-1372

[6] Shi Fu-qian, Sun Shou-qian, Xu Jiang. Employing rough set and association mining in KANSEI knowledge extraction [J]. Information Sciences, 2012, 196: 118-128

[7] Bahga A, Madiseti V K. Analyzing Massive Machine Maintenance Data in a Computing Cloud[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(10): 1831-1843

[8] 胡峰, 王国胤. 基于分治法的快速确定规则获取算法[J]. 模式识别与人工智能, 2010, 23(3): 349-356

[9] Zhao Su-yun, Tsang E C C, Chen De-gang. Building a Rule-Based Classifier—A Fuzzy-Rough Set Approach [J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(5): 624-638

[10] 梁德翠, 胡培. 面向属性归纳下的多层次决策规则获取算法[J]. 信息与控制, 2012, 41(1): 69-74

[11] Huang C-C, Tseng B, Fan Yu-neng. Alternative rule induction methods based on incremental object using rough set theory[J]. Applied Soft Computing, 2013, 13(1): 372-389

[12] 徐怡, 李龙澍, 李学俊. 改进的 LEM2 规则提取算法[J]. 系统工程理论与实践, 2010, 30(10): 1841-1849

[13] 黄兵, 周献中. 不一致决策表中规则提取的矩阵算法[J]. 系统工程与电子技术, 2005, 27(3): 441-445

[14] Chen C-H, Rao Zhi-ming. MRM: A matrix representation and mapping approach for knowledge acquisition [J]. Knowledge-Based Systems, 2008, 21(4): 284-293

[15] 鄂旭, 邵良杉, 张毅智, 等. 一种基于粗糙集理论的规则提取方法[J], 计算机科学, 2011, 38(1): 232-235

[16] 徐章艳, 刘作鹏, 杨炳儒, 等. 一个复杂度 $\max(O(|C||U|), O(|C|^2|U/C|))$ 快速属性约简算法[J]. 计算机学报, 2006, 29(3): 391-399