

文章编号: 1001-0920(2007)12-1368-05

一种基于粗糙集理论的最简决策规则挖掘算法

钱进¹, 孟祥萍², 刘大有³, 叶飞跃¹

(1. 江苏技术师范学院 计算机科学与工程学院, 江苏 常州 213001; 2. 长春工程学院
电气与信息学院, 长春 130012; 3. 吉林大学 计算机科学与技术学院, 长春 130012)

摘要: 研究粗糙集理论中可辨识矩阵, 扩展了类别特征矩阵, 提出一种基于粗糙集理论的最简决策规则算法. 该算法根据决策属性将原始决策表分成若干个等价子决策表, 借助核属性和属性频率函数对各类别特征矩阵挖掘出最简决策规则. 与可辨识矩阵相比, 采用类别特征矩阵可有效减少存储空间和时间复杂度, 增强规则的泛化能力. 实验结果表明, 采用所提出的算法获得的规则更为简洁和高效.

关键词: 粗糙集; 类别特征矩阵; 决策规则; 分类

中图分类号: TP301 **文献标识码:** A

A mining algorithm for concise decision rules based on rough sets theory

QIAN Jin¹, MENG Xiang-ping², LIU Dayou³, YE Fei-yue¹

(1. Institute of Computer Science and Engineering, Jiangsu Teachers University of Technology, Changzhou 213001, China; 2. Institute of Electric and Information, Changchun Institute of Technology, Changchun 130012, China; 3. Institute of Computer Science and Technology, Jilin University, Changchun 130012, China. Correspondent: QIAN Jin, E-mail: qjqjlqyf@163.com)

Abstract: By the research of discernibility matrix in rough sets, the extended class feature matrices are presented. A mining algorithm for concise decision rules based on rough set theory is proposed. The decision table is divided into many equivalence decision tables by using the decision attributes, core attributes are extracted and the attributes frequent functions are computed to mine decision rules from the small class feature matrices. Compared with algorithms based on discernibility matrices, the proposed algorithm is of much less space complexity and time complexity and has more generalizing ability. The experiment results on data sets in Rosetta software and comparison show that the algorithm provides more precise and simple decision rules.

Key words: Rough sets; Class feature matrix; Decision rule; Classification

1 引言

提取决策规则是数据挖掘中的一项重要内容, 它从一个已知数据库中挖掘一组规则(规则集)来描述各类数据的特征, 并使用此规则集预测未知数据的类别. 粗糙集(RS)理论是研究不精确、不确定性知识的一种数学工具. 规则生成的基本步骤是先求出知识决策表中的核属性; 然后逐步挑选和扩展属性集合, 以求出一个属性约简; 再从该属性约简中找到决策规则. 然而, 求所有约简已被证明是 NP 问题, 因此, 寻求快速高效的属性约简算法和决策规则

挖掘的值约简算法, 便成为粗糙集理论研究的重点之一. 目前, 已有多种基于 Rough 集理论的决策规则提取算法^[1-11], 主要方法有: 基于布尔推理^[2]的最小决策算法, 基于正区域^[3]和基于区分矩阵的属性约简算法^[4-6]等.

本文提出一种基于粗糙集理论的不确定性最简决策规则挖掘算法. 它根据决策属性将大决策表分解为若干个等价类小决策表; 然后生成类别特征矩阵, 借助核属性和属性频率函数直接导出最简决策规则. 通过算法的应用实例与比较分析可知, 该算法能

收稿日期: 2006-08-03; 修回日期: 2007-01-12.

基金项目: 国家自然科学基金项目(60173006); 吉林省科技发展计划项目(20040539); 常州市“831工程”基金项目(KYZ06002).

作者简介: 钱进(1975—), 男, 江苏泰兴人, 讲师, 硕士, 从事数据挖掘、粗糙集等研究; 孟祥萍(1961—), 女, 长春人, 教授, 博士后, 从事人工智能、数据挖掘等研究.

获取简洁而有效的规则.

2 相关基本概念

下面简要介绍本文主要用到的一些 Rough 集的基本概念,详细内容参见文献[1,2,4,6].

定义 1 设 U, A, V, F 是信息系统,如果 A 可以分为条件属性 C 和决策属性 D ,即 $C \cup D = A, C \cap D = \emptyset$,则称该信息系统为决策信息系统或决策表,其中 D 一般只含有一个属性 d . 记

$$R_c = \{(x_i, x_j) \mid f_c(x_i) = f_c(x_j) (a_i \in C)\},$$

$$R_d = \{(x_i, x_j) \mid d(x_i) = d(x_j)\}. \quad (1)$$

若 $R_c \subseteq R_d$,则称该决策表为相容决策表;否则称为不相容决策表.

定义 2 可辨识矩阵定义为 $M^c = (c_{ij})_{n \times n}$, 其中

$$c_{ij} = \begin{cases} \{c \in C: f_c(x_i) = f_c(x_j)\}, & d(x_i) = d(x_j); \\ \emptyset, & \text{其他}; \end{cases}$$

$$i, j = 1, 2, \dots, n. \quad (2)$$

3 类别特征矩阵

从定义 2 可以看出,可辨识矩阵中存在许多空集元素,这显然浪费了大量存储空间. 所产生的约简包含了对所有类别数据起分类作用的属性,这使得针对每一类别数据的值约简有可能在一个较大的属性集上进行,给决策规则的提取带来了困难. 显然,相同决策对象的比较结果没有必要保存下来,由此提出了关于属性的类别特征矩阵. 为构造类别特征矩阵,首先将决策表中的对象按其类别进行划分. 如果有多个决策属性,则可将多个决策属性看成一个决策属性,而多个决策属性组合值为该属性的取值范围. 不失一般性,假设仅有一个决策属性 d . 共有 M 类对象,其决策属性值映射为 $1, 2, \dots, M$. 这样,将整个决策表分为 M 个子决策表 T_1, T_2, \dots, T_M , 每个子决策表中包含同一类别的对象,其对象个数分别为 $|T_1|, |T_2|, \dots, |T_M|$.

定义 3 设 $c \in C$,第 i 类对象与第 j 类对象关于属性 c 的类间特征矩阵定义为 $M_{ij}^c = \{c_{ij}\}$, 其中

$$c_{ij} = \begin{cases} 1, & f_c(t_i) = f_c(t_j), d(t_i) = d(t_j); \\ 0, & f_c(t_i) \neq f_c(t_j), d(t_i) = d(t_j); \\ & t_i \in T_i, t_j \in T_j. \end{cases} \quad (3)$$

从定义 3 可以看出,关于条件属性 c 的类间特征矩阵删除了可辨识矩阵中大量无用的空集元素,对于多数大容量的决策表而言,类间特征矩阵仅占用了较小的存储空间,且容易分析各属性的重要性.

定义 4 类间特征矩阵 M_{ij}^c 的每个元素定义为 $C_{ij} = 1 \ 2 \ \dots \ k$, 其中

$$i = \begin{cases} 0, & f_c(t_i) = f_c(t_j); \\ 1, & f_c(t_i) \neq f_c(t_j); \end{cases} \quad (4)$$

$l = 1, 2, \dots, k; c$ 为 l 对应的 $c_l; t_i \in T_i, t_j \in T_j$.

根据定义 3 和定义 4,可得到如下性质:

性质 1^[7] 对于决策表 T , 设 $C = (c_1, c_2, \dots, c_k)$, 有下式成立:

$$M_{ij}^C = M_{ij}^{c_1} (+) M_{ij}^{c_2} (+) \dots (+) M_{ij}^{c_k}. \quad (5)$$

由定义 3 和定义 4 可直接证得,此略.

定义 5 关于属性 c 的第 i 类对象的 M_i^c 定义为

$$M_i^c = (M_{ij}^c), \quad (6)$$

其中 $j = 1, 2, \dots, i - 1, i + 1, \dots, M$.

性质 2 设 $c \in C$,若 M_i^c 中所有元素都为 1,则 c 能够辨识第 i 类对象与其他类的对象.

证明(反证法) 假设 c 不能辨识第 i 类对象与其他类的对象,则存在第 i 类某一对象 x 的属性 c 与其他类中某一对象 y 的属性 c 的值相同. 根据 M_i^c 的定义, x 所对应的行和 y 所对应的列的交叉处其元素为 0. 这与 M_i^c 中所有的元素都为 1 相矛盾.

推论 1 若 M_i^c 中某些行的所有的元素都为 1, 则 c 能将这些行所对应的对象与其决策不同的对象区分开.

若 M_i^c 某些行上存在 0,则说明属性 c 不能将这些行所对应的对象与其决策不同的对象区分开,这时需要加入其他属性才能区分.

定义 6^[7] 设 $c \in C, v \in C, v$ 相对于 M_i^c 的互补元素为 $V_{ij} = \{m_{ij}(v) : m_{ij}(c) = 0 \text{ 且 } m_{ij}(v) = 1\}$, $m_{ij}(c)$ 和 $m_{ij}(v)$ 分别为 M_i^c 和 M_i^v 的第 i 行第 j 列的元素. v 相对于 c 的互补元素越多,说明 v 与 c 的互补性越大.

若 M_i^c 中某些行上元素为 0,而 M_i^v 中相对应的位置上元素为 1,则属性组合 cv 能将这些行上所有元素都出现 1,这说明 cv 能将这些行所对应的对象与其决策不同的对象区分开. 如果 M_i^v 剩余行中还存在某个元素没有出现 1,则继续寻找 M_i^v 的互补元素,直到合并后矩阵中所有元素都出现 1.

4 基于粗糙集理论的最简规则提取算法

对于大型数据集而言,求互补元素要反复读取数据集,这样势必增加额外的系统读取数据的开销,进而造成算法运行时间变长. 为解决这个问题,引进文献[8]的算法思想:如果一个属性在可辨识矩阵中出现得越频繁,则它的潜在分类能力就越大. 利用不同属性之间区分能力的互补作为启发信息,即考虑约简中已选中的属性还不能区分那些对象中属性的区分能力,体现为出现频率. 令 $P(c_k)$ 是 M 中属性 c_k 的频率函数,定义为属性 c_k 在类别特征矩阵中出

现的次数.如何选取互补元素的主要思想体现在下面的算法 1 和算法 2 中.

本文主要对决策表提取确定性的决策规则,但决策表中可能存在冗余和不相容的数据,从这些数据中不仅不会提取出更多的确定性规则,而且还会在规则提取中增加计算量,因此需要清洗冗余和不相容数据.冗余数据处理可参见文献[3,9]中算法 1;对于不相容数据,可采用文献[3,10]中算法 1,将所有不相容数据归为一类,这样不相容决策表就变成了相容决策表,挖掘规则时将不挖掘这一类决策规则.根据决策属性分别生成对应的类别特征矩阵.于是,基于粗糙集理论的挖掘决策规则的算法主要分为两步:第 1 步,在类别特征矩阵中提取核属性和计算各属性出现的次数(算法 1);第 2 步,借助核属性集合和各属性频率函数对类别特征矩阵反复进行规则提取(算法 2).

算法 1 提取核属性和计算各属性频率函数.

输入: M_i^c ;

输出: 核属性集和各属性出现的次数.

- 1) for $i = 1$ to $|T_i|$
- 2) for $j = 1$ to $|U| - |T_i|$
- 3) CountNum($c_k, \text{Num}[i][k]$)
// 统计每行中各属性出现的次数
- 4) CountElement($m_j(c), s$)
- 5) // 统计每个元素中出现 1 的个数
- 6) if $s = 1$ then Core = Core $\cup \{c_i\}$
- 7) next j
- 8) next i
- 9) for $j = 1$ to k
- 10) for $i = 1$ to m
- 11) $\text{Num}[m+1][j] = \text{Num}[m+1][j] + \text{Num}[i][j]$

为了获取最简决策规则,生成的规则要尽可能泛化更多的对象.算法 2 提取决策规则时,首先考虑核属性,若没有核属性,就选择非核属性中出现次数最多的属性.只要单个属性或属性组合使得某些行上所有元素出现 1,就生成决策规则;然后在剩余的行中继续利用算法 2 生成决策规则.

算法 2 提取类别特征矩阵中决策规则.

输入: 核属性集合 Core, 各属性出现的次数 Num[[]], T_i 和 M_i^c ;

输出: 决策规则集 Rules.

- 1) 核属性集不为空,选择出现次数最多的核属性;否则选择出现次数最多的非核属性.
- 2) 根据各属性出现的次数,生成决策规则:
如果 $\text{Num}[m+1][k] = m * n$,则 $C = C$

c_k ,根据 T_i 生成决策规则 $C = d_i$,算法结束;

如果某些行 $\text{Num}[i][k] = n$,则 $C = C \cup c_k$,根据 T_i 生成这些行的决策规则 $C = d_i$,转 3);

如果不满足 或 ,则寻找互补元素 v ,将类别特征矩阵中包含已选择过的属性的元素置空,调用算法 2,重新计算核属性和各属性出现次数,并选择出现次数最多的属性 v , $C = C \cup v$;如果 M_i^c 还不满足 或 ,则重复 ;否则根据 T_i 生成 $C = d_i$.

3) 如果类别特征矩阵中还有剩余行,则递归调用算法 2;否则算法结束.

5 实例研究

5.1 算法应用实例

为考察算法的有效性,选择文献[6,11]中一个已知规则的决策表(见表 1)进行对比分析.表 1 中: a_1, a_2, a_3, a_4 为条件属性,分别代表 Outlook, Temperature, Humidity, Windy; d 为决策属性.

表 1 关于气象信息的信息系统

U	a_1	a_2	a_3	a_4	d
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

根据定义 4 和定义 5 生成 $M_{d=P}^c$ 类别特征矩阵.决策属性仅两个取值, $M_{d=N}^c$ 是 $M_{d=P}^c$ 的转置矩阵.

首先提取 $M_{d=P}^c$ 决策规则,步骤如下:

对于 $M_{d=P}^c$,各属性出现次数分别为 33,30,27,24,核属性为 a_1, a_4 ,但 a_1 使 $M_{d=P}^c$ 中第 1,4,8,9 行上所有元素都出现了 1,则生成决策规则为

$$a_1 = \text{Overcast} \quad d = P.$$

对于剩余的 5 行(第 2,3,5,6,7 行),计算各属性出现次数为 13, 17, 17, 14,核属性为 a_1, a_4 ,单个属性 a_4 无法使任何行上所有元素出现 1,于是核属性 a_1 优先作为互补元素.因为 a_1 和 a_4 使 $M_{d=P}^c$ 中第 2,3 和 6 行上所有元素都出现 1,故生成决策规则为

$$a_1 = \text{Rain}, a_4 = \text{False} \quad d = P.$$

经过处理,最后只剩下 2 行(第 5 和 7 行),计算各属性出现次数为 4,7,8,5,没有核属性.选择单个属性 a_3 ,但无法使任何行上所有元素出现 1,于是将包含 a_3 的元素置空,重新计算其余属性出现次数为 2,1,1.选择 a_1 作为互补元素,结果 a_1 和 a_3 使 $M_{d=P}^C$ 中所有行上的元素都出现了 1,则生成决策规则为

$$a_1 = \text{Sunny}, a_3 = \text{Normal} \quad d = P.$$

$$M_{d=P}^C = \begin{pmatrix} 1000 & 1001 & 1111 & 1100 & 1101 \\ 1100 & 1101 & 0111 & 1000 & 0001 \\ 1110 & 1111 & 0001 & 1110 & 0111 \\ 1111 & 1110 & 1000 & 1111 & 1110 \\ 0110 & 0111 & 1001 & 0110 & 1111 \\ 1110 & 1111 & 0101 & 1010 & 0011 \\ 0111 & 0110 & 1100 & 0011 & 1010 \\ 1101 & 1100 & 1110 & 1001 & 1000 \\ 1010 & 1011 & 1101 & 1110 & 1111 \end{pmatrix}.$$

其次,提取 $M_{d=N}^C$ 中决策规则. $M_{d=N}^C$ 为

$$M_{d=N}^C = \begin{pmatrix} 1000 & 1100 & 1110 & 1111 & 0110 \\ 1001 & 1101 & 1111 & 1110 & 0111 \\ 1111 & 0111 & 0001 & 1000 & 1001 \\ 1100 & 1000 & 1110 & 1111 & 0110 \\ 1101 & 0001 & 0111 & 1110 & 1111 \\ 1110 & 0111 & 1101 & 1010 \\ 1111 & 0110 & 1100 & 1011 \\ 0101 & 1100 & 1110 & 1101 \\ 1010 & 0011 & 1001 & 1110 \\ 0011 & 1010 & 1000 & 1111 \end{pmatrix}.$$

在 $M_{d=N}^C$ 中,各属性出现次数分别为 33,30,27,24,核属性为 a_1 和 a_4 .因为单个属性 a_1 不能使任何行上所有元素出现 1,所以将核属性 a_4 优先作为互补元素.由于 a_1 和 a_4 使 $M_{d=N}^C$ 中第 3 行和第 5 行上所有元素都出现了 1,则生成决策规则为

$$a_1 = \text{Rain}, a_4 = \text{True} \quad d = N.$$

对于剩余的 3 行(第 1,2 和 4 行),各属性出现次数分别为 21,19,18,12,核属性为 a_1 .因为 a_1 不能使任何行上所有元素出现 1,所以寻找互补元素.将 $M_{d=N}^C$ 中包含 a_1 的元素置空,再次计算其余属性出现次数,分别为 5,6,3.选择 a_3 作为互补元素,由于 a_1 和 a_3 使 $M_{d=N}^C$ 中所有行(第 1,2 和 4 行)上的元素都出现了 1,生成决策规则为

$$a_1 = \text{Sunny}, a_3 = \text{High} \quad d = N.$$

5.2 分析与比较

5.2.1 性能分析

在不考虑对称矩阵压缩存储情况下,可辨识矩阵中元素个数为 $|U| \times |U|$;而采用类别特征矩阵,

其元素个数为 $\sum_{i=1}^M |T_i| \times (|U| - |T_i|)$,节省了 $\sum_{i=1}^M |T_i| \times |T_i|$ 个元素的存储空间.

在可辨识矩阵中,要生成决策规则,需不断地对整个决策表进行处理.而本文算法采用分而治之的方法,只需对单个类别特征矩阵进行处理就能生成决策规则.这对于大型决策表而言,可以并行操作来提取规则,从而节省挖掘时间.

两个对象比较是否相等,计算复杂度为 $O(|C| + |D|)$.利用文献[3]中算法 1,按决策属性将 $|U|$ 个对象进行排序,需要 $|D| |U| \log |U|$ 次比较,计算复杂度为 $O(|D| |U| \log |U|)$.如果决策属性由多个属性组成,则将这些属性的组合看成一个决策属性,其各属性的组合值看成该决策属性的取值.然后按条件属性将 $|U|$ 个对象进行排序,需要 $(|C| + 1) |U| \log |U|$ 次比较,计算复杂度为 $O((|C| + 1) |U| \log |U|)$.生成一个包含 $|T_i| \times (|U| - |T_i|)$ 个元素的类别特征矩阵,计算复杂度为 $O((|C| + 1) |T_i| \times (|U| - |T_i|))$.

对于算法 1,经过简单计算,计算的复杂度为 $O((|C| + 1) |T_i| \times (|U| - |T_i|))$,为多项式算法.算法 2 采用分而治之的方法,主要是调用计算核属性和各属性频率函数的算法 1 来寻找互补元素.对于一个 $|T_i| \times (|U| - |T_i|)$ 类别特征矩阵,计算复杂度为 $O(|C| (|C| + 1) |T_i| \times (|U| - |T_i|))$.对于算法 2,最坏情况下时间复杂度为 $O(|C|^2 |U|^2)$.

通过以上分析可知,整个算法的计算复杂度在最坏情况下是 $O(|C|^2 |U|^2)$,是多项式算法,避免了属性个数呈指数级增长,降低了计算工作量.

5.2.2 与其他算法比较

文献[11]以条件属性子集的分类一致性来度量属性的重要性,逐步加入重要的属性,当选择的属性子集能够正确分类时,则获取到决策规则.但在生成规则后,仍需对规则集进行规则约简,最终能得到 5 条最简规则.与文献[11]中提到的 RITIO 算法和 LEM2 算法相比,本文算法在规则数目和规则总长度上更小.

文献[6]先通过属性约简得到 $\{a_1, a_2, a_4\}$ 和 $\{a_1, a_3, a_4\}$;然后分别进行值约简.以 $\{a_1, a_2, a_4\}$ 为条件属性得到的规则有 7 条,以 $\{a_1, a_3, a_4\}$ 为条件属性得到的规则为 5 条.该算法需要通过计算判断才能得出 5 条最简规则.而本文利用类别特征矩阵,借助核属性和属性频率函数能够直接导出最简决策规则为 5 条,无需进行规则比较分析.

5.3 实验结果



为进一步验证该算法,对波兰华沙大学与挪威科技大学联合开发的 Rosetta 软件中 Iris, Australian 和 HSV 等数据集进行了测试.利用 Rosetta 软件中的 BOrthogonalScaler (简称 BROS 方法)和 Equal Frequency Scaler (简称 EFS 方法)对连续属性数据进行离散化.利用 JohnsonReducer 约简方法对离散化后的数据集进行约简;然后生成规则.对 Rosetta 软件挖掘出的确定性规则还需要进行值约简,经过比较,与本文算法挖掘出的规则相同. Rosetta 软件的确定性规则长度是指没经过值约简的规则长度.由此可见,用本文算法挖掘出的决策规则无论是数量上还是长度上都要简洁得多.

表2 规则挖掘算法比较

决策表	实例数	离散方法	相容	Rosetta 软件		本文算法	
				数量	长度	数量	长度
Iris	150	BROS	是	19	76	9	28
			否	21	84	9	20
Australian	690	BROS	是	601	3 606	236	733
			否	658	7 238	393	1 264
HSV	122	BROS	是	114	1 026	76	561
			否	113	904	81	316

6 结 语

本文充分考虑了可辨识矩阵的特性,利用类别特征矩阵和单个属性的重要性,提出了基于粗糙集理论的最简决策规则挖掘算法.该算法具有如下特点:1)利用类别特征矩阵节省了存储空间;2)借助核属性和属性频率函数求互补元素,避免了组合爆炸;3)采用分而治之的方法,不需要对整个决策表进行反复操作与大量运算.实验结果表明,所获得的规则最简,规则集的规模小,具有较好的可理解性和较强的泛化能力.

参考文献(References)

- [1] Pawlak Z. Rough sets [J]. Int J of Computer and Information Science, 1982, 11(5): 341-356.
- [2] Pawlak Z. Rough sets: Theoretical aspects of reasoning about data [M]. Dordrecht: Kluwer Academic Publishers, 1991.
- [3] 刘少辉, 盛秋戩, 吴斌, 等. Rough 集高效算法研究[J]. 计算机学报, 2003, 26(5): 524-529.
(Liu Shao-hui, Sheng Qi-ji, Wu Bin, et al. Research on efficient algorithms for rough set methods [J]. Chinese J of Computer, 2003, 26(5): 524-529.)
- [4] Skowron A, Rauszer C. The discernibility matrices and functions in information system[C]. Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publishers, 1992: 331-362.
- [5] Wang Jue, Wang Ju. Reduction algorithms based on discernibility matrix: The ordered attributes method[J]. J of Computer Science and Technology, 2001, 16(6): 489-504.
- [6] 常梨云, 王国胤, 吴渝. 一种基于 Rough Set 理论的属性约简及规则提取方法[J]. 软件学报, 1999, 10(11): 1206-1211.
(Chang Li-yun, Wang Guo-ying, Wu Yu. An approach for attribute reduction and rule generation based on rough set theory [J]. J of Software, 1999, 10(11): 1206-1211.)
- [7] 赵卫东, 戴伟辉. 基于特征矩阵的决策表约简研究[J]. 系统工程理论与实践, 2003, (3): 65-69.
(Zhao Wei-dong, Dai Wei-hui. Feature reduct of decision tables based on feature of matrix [J]. System Engineering Theory and Practice, 2003, (3): 65-69.)
- [8] Hu X, Cercone N. Learning in relational databases: A rough set approach [J]. Int J of Computational Intelligence, 1995, 11(2): 323-338.
- [9] 谭天乐, 宋执环, 李平. 信息系统数据清洗、规则提取的矩阵算法[J]. 信息与控制, 2003, 32(4): 289-294.
(Tan Tian-le, Song Zhi-huan, Li Ping. Matrix computation for data cleaning and rule extraction in information system [J]. J of Information and Control, 2003, 32(4): 289-294.)
- [10] 刘启和, 李凡, 闵帆, 等. 一种基于新的条件信息熵的高效知识约简算法[J]. 控制与决策, 2005, 20(8): 878-882.
(Liu Qi-he, Li Fan, Min Fan, et al. An efficient knowledge reduction algorithm based on new conditional information entropy [J]. Control and Decision, 2005, 20(8): 878-882.)
- [11] 代建华, 潘云鹤. 一种基于分类一致性的决策规则获取算法[J]. 控制与决策, 2004, 19(10): 1086-1090.
(Dai Jian-hua, Pan Yun-he. Algorithm for acquisition of decision rules based on classification consistency rate [J]. Control and Decision, 2004, 19(10): 1086-1090.)