文章编号: 1001-0920 (2015) 01-0143-06 **DOI:** 10.13195/j.kzyjc.2014.0126

基于粒计算的最简决策规则挖掘算法

陈泽华, 张 裕, 谢 刚

(太原理工大学 信息工程学院，太原 030024)

摘 要: 传统的规则挖掘算法通常先约简属性再约简属性值. 该方法存在冗余计算, 当样本集增大时, 复杂性急剧增加. 对此提出一种基于粒计算的最简决策规则挖掘算法. 首先, 在不同粒度空间下计算条件粒与决策粒之间的粒关系矩阵; 然后, 将粒关系矩阵中隐含的信息 H_1、H_2 作为启发式算子, 按信息粒约简属性值; 最后, 去除冗余属性并设置终止条件, 实现决策规则的快速挖掘. 理论分析和实验结果表明, 所提出的算法可以获得更简洁的规则, 且规则的泛化能力更强.

关键词: 决策规则; 粒计算; 粒度; 粒矩阵

中图分类号: TP273 **文献标志码:** A

Mining algorithm for concise decision rules based on granular computing

CHEN Ze-hua, *ZHANG Yu*, *XIE Gang*

(School of Information Engineering，Taiyuan University of Technology，Taiyuan 030024，China．Correspondent：CHEN Ze-hua，E-mail：zehuachen@163.com)

Abstract: The traditional rule mining algorithm includes attribute reduction and attribute value reduction, which incorporates redundant computation. The complexity of the algorithm will increase dramatically as the sample dataset increases. Therefore, the granular computing(GrC) method is adopted. Firstly, the granular-relation matrices between condition granules and decision granules in different granular spaces are computed. Then the attribute value is reduced according to H_1 and H_2 which are hidden in the granular-relation matrices. Furthermore, redundant attributes are removed and the termination condition is set, which can accelerate the mining of decision rules. The theoretical analysis and experimental results show that proposed algorithm can acquire more concise rules, and the rules have better generalizing ability.

Keywords: decision rules; granular computing; granularity; granular matrix

0 引 言

规则挖掘是数据挖掘的一项重要内容, 传统的基于粗糙集理论[1]的规则挖掘方法是先求决策信息系统 (决策表) 的属性约简, 再通过逐行约简属性值得到决策规则. 典型的规则挖掘算法主要有[2-4]: 基于布尔推理的最小决策算法[2], 基于正区域[3]和基于区分矩阵的属性约简算法[4]等. 利用粗糙集理论挖掘规则的关键在于属性约简, 属性约简结果直接影响规则挖掘的结果, 因此许多研究者重点对属性约简算法[3]进行了研究. 先约简属性再约简属性值的规则挖掘算法是常见算法, 但在属性约简和属性值约简的过程中存在冗余计算, 并且随着样本集的增大, 按决策规则逐行约简属性值算法的复杂性逐渐增大.

近年来, 不少学者避开了对属性约简的研究, 直接研究决策表的规则挖掘[5-7]. 文献 [5] 以条件属性子集的分类一致性来度量属性的重要性, 当选择的属性子集能正确分类时, 获取决策规则, 但仍需对生成的规则集进行简化; 文献 [6] 在研究可辨识矩阵的基础上提出了类别特征矩阵的概念, 将原始决策表分成若干个等价子决策表, 并借助核属性和属性频率函数对各类别特征矩阵挖掘决策规则, 但生成的规则并非最简; 文献 [7] 从分析属性约简的粒度原理出发, 指出了传统的规则挖掘方法存在的弊端, 并借此提出一种基于最大粒的规则获取算法, 通过在不同的知识粒空间中进行粒子合取运算寻找最大粒, 最终获取决策规则, 但当样本集增大时, 算法复杂性会急剧增加. 上述算法都有意避开了属性约简这一过程, 与基于粗糙集理论的规则挖掘算法相比, 降低了算法的复杂性, 但规

收稿日期: 2014-01-20; **修回日期:** 2014-04-17.

基金项目: 国家自然科学基金项目(61402319); 山西省回国留学人员科研项目(2013-031).

作者简介: 陈泽华(1974−), 女, 副教授, 博士, 从事粒计算、智能信息处理等研究; 张裕(1989−), 男, 硕士生, 从事粒计算、粗糙集理论及应用的研究.

则数量和规则的简洁性还有待提高. 因此, 如何从算法复杂性、规则的简洁程度、规则数量和规则的泛化能力等角度来提高规则挖掘算法的性能和效果, 是近年来研究者需要思考的问题.

粒计算[8]的核心思想是对待求解的问题进行粒化, 在多个粒度空间对问题进行分析和求解, 进而合成原始问题的解, 符合人类从多角度分析问题、求解问题的认知规律, 并受到了研究者的关注.

本文将属性约简和属性值约简过程合二为一, 以知识粒为单位挖掘规则. 先对决策信息系统分层粒化, 在不同粒度的知识空间下计算粒关系矩阵, 并从中获取启发式信息 H_1、H_2. 根据启发式信息确定信息粒的属性值约简顺序, 在此基础上去除冗余属性, 并设定终止条件, 实现决策规则的快速挖掘. 理论分析和UCI数据集的测试结果表明, 该算法能获得所有最简规则.

1 基本概念

定义 1 一般地, 信息系统可以用一个四元组 $S=(U,A,V,f)$ 来表示. 其中: $U=\{u_1,u_2,\cdots,u_l\}$ 是非空有限对象集, 称为论域; $A=\{a_1,a_2,\cdots,a_n\}$ 是属性集合; $V=\bigcup\limits_{a\in A}V_a$ 是属性值的集合, V_a 是属性 a 的值域; $f:U\times A\to V$ 是信息函数, 它指定 U 中每一个对象的属性值. 若 $A=C\bigcup D$ 且 $C\bigcap D=\varnothing$, 则称该信息系统为决策信息系统, 也称为决策表[1-2]. 其中: C 为条件属性, D 为决策属性. 决策信息系统中的每一行代表一条决策规则.

对于任意的条件属性 $C'\subseteq C$, 定义一个 U 上的不可分辨关系:

$$\mathrm{IND}(C')=\{(x,y)\in U\times U|\forall_{a\in C'}(f_a(x)=f_a(y))\}.$$

其中: $\mathrm{IND}(C')$ 是 U 上的等价关系, 所有等价类的集合记为 $U/\mathrm{IND}(C')$. 如果 $U/\mathrm{IND}(C)\subseteq U/\mathrm{IND}(D)$, 则称 S 是一致决策信息系统.

定义 2 设 $C'\subseteq C$, C' 对论域的划分 $U/\mathrm{IND}(C')=\{X_1,X_2,\cdots,X_m\}$, 则 $U/\mathrm{IND}(C')$ 的知识粒度[9]定义为

$$G(U/\mathrm{IND}(C'))=\frac{\sum\limits_{i=1}^{m}|X_i|^2}{|U|^2},$$

其中 $|X_i|$ 表示集合 X_i 的势. 显然有 $G(U/\mathrm{IND}(C))\leqslant G(U/\mathrm{IND}(C'))$.

定义 3 条件属性 C' 和决策属性 D 对论域 U 的划分

$$U/\mathrm{IND}(C')=\{X_1,\cdots,X_i,\cdots,X_m\},$$
$$1\leqslant i\leqslant m\leqslant l;$$
$$U/\mathrm{IND}(D)=\{Y_1,\cdots,Y_j,\cdots,Y_s\},$$
$$1\leqslant j\leqslant s\leqslant l.$$

定义等价类 X_i、Y_j 为信息粒[10], 分别用一个长度为 l 的二进制向量表示, 即

$$X_i=(a_{i1},a_{i2},\cdots,a_{ik},\cdots,a_{il}). \tag{1}$$

$$Y_j=(b_{j1},b_{j2},\cdots,b_{jk},\cdots,b_{jl}). \tag{2}$$

$$a_{ik}=\begin{cases}1,\ u_k\in X_i,\ 1\leqslant k\leqslant l;\\0,\ u_k\notin X_i,\ 1\leqslant k\leqslant l.\end{cases} \tag{3}$$

$$b_{jk}=\begin{cases}1,\ u_k\in Y_j,\ 1\leqslant k\leqslant l;\\0,\ u_k\notin Y_j,\ 1\leqslant k\leqslant l.\end{cases} \tag{4}$$

定义3将原始的以个体为单位的决策信息系统转换为以信息粒(等价类)为单位的决策信息系统, 系统的知识粒度变粗.

在决策信息系统中, 令 $1\leqslant w\leqslant n$, w 表征当前系统的粒度, n 为条件属性个数. 这样系统对应有 n 种粒度, 并且粒度 w 越小, 系统的知识粒度越粗. 假设此时系统的粒度为 w, 则在同一粒度下将产生 C_n^w 个知识空间[11]. 下文将在不同粒度的知识空间中挖掘规则.

2 粒矩阵与粒关系矩阵

定义 4 定义粒矩阵[10]$\mathrm{GrM}=\{X_{m\times l},Y_{s\times l}\}$, 其中

$$X_{m\times l}\triangleq X=\begin{bmatrix}X_1\\X_2\\\vdots\\X_m\end{bmatrix}=\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1l}\\a_{21}&a_{22}&\cdots&a_{2l}\\\vdots&\vdots&\ddots&\vdots\\a_{m1}&a_{m2}&\cdots&a_{ml}\end{bmatrix}, \tag{5}$$

$$Y_{s\times l}\triangleq Y=\begin{bmatrix}Y_1\\Y_2\\\vdots\\Y_s\end{bmatrix}=\begin{bmatrix}b_{11}&b_{12}&\cdots&b_{1l}\\b_{21}&b_{22}&\cdots&b_{2l}\\\vdots&\vdots&\ddots&\vdots\\b_{s1}&b_{s2}&\cdots&b_{sl}\end{bmatrix}. \tag{6}$$

称矩阵 X 为条件粒矩阵(条件信息粒), 称矩阵 Y 为决策粒矩阵(决策信息粒).

定义 5 在定义4的基础上定义粒关系矩阵

$$C_{m\times s}\triangleq C=XY^{\mathrm{T}}=\begin{bmatrix}X_1Y_1&X_1Y_2&\cdots&X_1Y_s\\X_2Y_1&X_2Y_2&\cdots&X_2Y_s\\\vdots&\vdots&\ddots&\vdots\\X_mY_1&X_mY_2&\cdots&X_mY_s\end{bmatrix}=(c_{ij})_{m\times s}. \tag{7}$$

其中: 元素 $c_{ij}=\sum_{k=1}^{l}(a_{ik}b_{kj})$ 反映了信息粒 X_i 包含在信息粒 Y_j 中元素的个数; Y^{T} 表示矩阵 Y 的转置; $C_{m\times s}$ 为条件信息粒 X 与决策信息粒 Y 的粒关系矩阵, 反映了信息粒 X_i 与信息粒 Y_j 之间的包含关系. 令 $\mathrm{NE}(i)=1$ 表示 $C_{m\times s}$ 中第 i 行非零元素个数为 1, $\mathrm{NE}(j)=1$ 表示 $C_{m\times s}$ 中第 j 列非零元素个数为 1.

性质 1　若 $\mathrm{NE}(i)=1$, 则信息粒 X_i 完全包含于信息粒 Y_j 中, 信息粒 X_i 能区分信息粒 Y_j 的部分 (或全部).

性质 2　若 $\mathrm{NE}(j)=1$, 则信息粒 Y_j 完全包含于信息粒 X_i 中, 信息粒 Y_j 能够完全被信息粒 X_i 的部分 (或全部) 区分.

根据性质 1 和性质 2 有以下推论:

推论 1　在决策信息系统中, 决策属性 D 完全依赖于条件属性 C' 的充要条件是: 对于矩阵 $C_{m\times s}$, 存在 $\mathrm{NE}(i)=1$ 对于所有 $1\leqslant i\leqslant m$ 都成立.

推论 1 表明, 若矩阵 $C_{m\times s}$ 中存在 $\mathrm{NE}(i)=1$ 对于所有 $1\leqslant i\leqslant m$ 都成立, 则说明条件属性 C' 对论域划分的信息粒能完全区分决策属性 D, 这时其他属性 $C-C'$ 可被约简.

性质 3　若 $\mathrm{NE}(i)=1$ 且 $\mathrm{NE}(j)=1$, 则信息粒 X_i 全部包含于信息粒 Y_j 中, 且信息粒 Y_j 能够完全被信息粒 X_i 区分.

证明　由 $\mathrm{NE}(i)=1$ 可知, 信息粒 X_i 完全包含于信息粒 Y_j 中, 同理, 当 $\mathrm{NE}(j)=1$ 时, 信息粒 Y_j 完全包含于信息粒 X_i 中. 由性质 1 和性质 2 可知, 信息粒 X_i 可以完全区分信息粒 Y_j. □

性质 1 ~ 性质 3 是本文进行规则挖掘的理论基础.

定义 6　在粒度小于 w 下所有能约简的规则记为 Rr_{w-1}, 当前粒度 w 下条件属性 C' 能约简的规则记为 Rr, 定义

$$H_1=|\mathrm{Rr}-\mathrm{Rr}\bigwedge\mathrm{Rr}_{w-1}|. \tag{8}$$

从定义 6 可以看出, H_1 越大, 粒度 w 下对应信息粒的区分能力越强.

定义 7　判断粒关系矩阵 $C_{m\times s}$ 中是否存在 $\mathrm{NE}(i)=1$ 且 $\mathrm{NE}(j)=1$, 记为

$$H_2=\begin{cases}1,\ \mathrm{NE}(i)=1,\mathrm{NE}(j)=1;\\ 0,\ \text{otherwise.}\end{cases}$$

$$1\leqslant i\leqslant m,\ 1\leqslant j\leqslant s. \tag{9}$$

由定义 6 和定义 7 定义的 H_1、H_2 是两个启发式算子, 本文利用这两个启发式算子对决策信息系统进行规则挖掘.

3　基于粒计算的最简决策规则挖掘算法

对决策信息系统挖掘规则的传统方法是先求属性约简, 再逐行提取规则, 中间包含了很多冗余计算, 最后的结果也取决于属性约简结果的好坏, 并且随着样本集的增大, 算法复杂性将大大增加. 文献 [7] 对属性约简进行了粒度原理分析并指出, 对决策信息系统进行属性约简得到的知识划分空间是极大近似划分空间, 但该知识空间的知识粒并不一定是整个知识空间中最 "粗" 的粒. 本文考虑在不同粒度层次的知识空间中挖掘规则. 为便于算法说明, 先给出符号定义.

3.1　符号定义

为了不失一般性, 假设决策信息系统有 n 个条件属性, 1 个决策属性. w 为条件属性 C' 所含条件属性的个数, 表征系统的粒度, $1\leqslant w\leqslant n$; G_w 为粒度 w 下的所有条件属性 C', 这样的条件属性有 C_n^w 个; X_w 为 G_w 中某一条件属性对应的条件粒矩阵; Y 为决策属性对应的决策粒矩阵; $C_{m\times s}$ 为粒关系矩阵.

3.2　算法描述

基于粒计算的最简决策规则挖掘算法.

输入: 决策信息系统;

输出: 所有最简决策规则.

1) 生成决策粒矩阵 Y 并取粒度 $w=1$.

2) 对 G_w 中每一个条件属性求条件粒矩阵 X_w 和粒关系矩阵 C_w, 计算 H_1、H_2, 保存相应数据并做以下处理:

① 寻找是否存在 $H_2=1$. 若存在, 则由性质 3 可知, 对应信息粒可以完全区分某一决策类, 约简过程中优先考虑, 这样可以保证在区分能力不变的情况下得到的规则最少, 约简相应的信息粒得到决策规则, 否则转 ②;

② 若不存在 $H_2=1$, 则对 H_1 值的大小进行比较, H_1 值越大, 对应信息粒的区分能力越大, 同样可以保证在区分能力不变的情况下得到的规则最少. 根据 H_1 值的大小确定信息粒的约简顺序, 通过约简信息粒得到决策规则, 转 ③;

③ 为了保证得到的规则是完整的 (即保持决策信息系统的分类能力不变)、无缺失, 在完成步骤 ① 和 ② 后, 需要判断约简得到的规则是否覆盖论域. 若覆盖, 则转 4), 否则转 3).

3) $w=w+1$, 对相应数据做初始化处理, 判断

w 是否小于 n. 若小于, 则转 2), 否则转 4).

4) 输出所有最简决策规则, 算法结束.

为了获取最简决策规则, 使得到的规则能泛化更多的对象, 算法在挖掘规则时先从粒度最“粗”的空间中进行, 从粗到细, 循序渐进, 直到挖掘到所有规则, 覆盖论域为止, 这样既能保证挖掘的规则最简, 也能保证规则是完整、无缺失的.

4 实验研究

4.1 算法实例与分析

4.1.1 算法实例

为了考察算法的有效性, 选择文献 [4] 中已知规则的决策信息系统进行对比分析, 决策信息系统如表 1 所示. 在表 1 中: $\{a,b,c,d\}$ 为条件属性, $\{e\}$ 为决策属性.

表 1 决策信息系统

U	a	b	c	d	e
1	1	0	0	1	1
2	1	0	0	0	1
3	0	0	0	0	0
4	1	1	0	1	0
5	1	1	0	2	2
6	2	1	0	2	2
7	2	2	2	2	2

根据本文提出的算法对表 1 进行规则挖掘, 步骤如下:

Step 1 根据定义 4 生成决策粒矩阵

$$Y=\begin{bmatrix}1&1&0&0&0&0&0\\0&0&1&1&0&0&0\\0&0&0&0&1&1&1\end{bmatrix}.$$

Step 2 粒度由粗到细, 在不同粒度下挖掘规则.

首先, 取粒度 $w=1$, 则 $G_1=\{\{a\},\{b\},\{c\},\{d\}\}$, 分别计算 G_1 中每一个条件属性的条件粒矩阵和粒关系矩阵

$$X_a=\begin{bmatrix}1&1&0&1&1&0&0\\0&0&1&0&0&0&0\\0&0&0&0&0&1&1\end{bmatrix},$$

$$X_b=\begin{bmatrix}1&1&1&0&0&0&0\\0&0&0&1&1&1&0\\0&0&0&0&0&0&1\end{bmatrix},$$

$$X_c=\begin{bmatrix}1&1&1&1&1&1&0\\0&0&0&0&0&0&1\end{bmatrix},$$

$$X_d=\begin{bmatrix}1&0&0&1&0&0&0\\0&1&1&0&0&0&0\\0&0&0&0&1&1&1\end{bmatrix};$$

$$C_a=X_aY^{\mathrm{T}}=\begin{bmatrix}2&1&1\\0&1&0\\0&0&2\end{bmatrix},$$

$$C_b=X_bY^{\mathrm{T}}=\begin{bmatrix}2&1&0\\0&1&2\\0&0&1\end{bmatrix},$$

$$C_c=X_cY^{\mathrm{T}}=\begin{bmatrix}2&2&2\\0&0&1\end{bmatrix},$$

$$C_d=X_dY^{\mathrm{T}}=\begin{bmatrix}1&1&0\\1&1&0\\0&0&3\end{bmatrix}.$$

计算 H_1、H_2 如表 2 所示.

表 2 粒度 $w=1$

条件属性	H_1	H_2
$\{a\}$	3	0
$\{b\}$	1	0
$\{c\}$	1	0
$\{d\}$	3	1

考察 H_1、H_2, 对于条件属性 $\{d\}$, $H_2=1$, 由矩阵 C_d 提供的信息可知, 属性 $\{d\}$ 的第 3 个信息粒可以完全区分属性 $\{e\}$ 的第 3 个信息粒, 约简这 3 条规则可得决策规则 $d=2\rightarrow e=2$ (覆盖规则 $\{5,6,7\}$).

另外, 属性 $\{a\}$ 的第 2 个信息粒还能部分区分属性 $\{e\}$ 的第 2 个信息粒, 约简得到决策规则 $a=0\rightarrow e=0$ (覆盖规则 $\{3\}$).

综上所述, 粒度 $w=1$ 下能约简的规则为 $\{3,5,6,7\}$, 未能覆盖整个论域, 继续考察粒度 $w=2$.

$G_2=\{\{ab\},\{ac\},\{ad\},\{bc\},\{bd\},\{cd\}\}$, 依次考察每个条件属性

$$X_{ab}=\begin{bmatrix}1&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\end{bmatrix},\ C_{ab}=\begin{bmatrix}2&0&0\\0&1&0\\0&1&1\\0&0&1\\0&0&1\end{bmatrix},$$

$$X_{ac}=\begin{bmatrix}1&1&0&1&1&0&0\\0&0&1&0&0&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\end{bmatrix},\ C_{ac}=\begin{bmatrix}2&1&1\\0&1&0\\0&0&1\\0&0&1\end{bmatrix},$$

$$X_{ad}=\begin{bmatrix}1&0&0&1&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&0&1&1&0\\0&0&0&0&0&0&1\end{bmatrix},\ C_{ad}=\begin{bmatrix}1&1&0\\1&0&0\\0&1&0\\0&0&1\\0&0&2\end{bmatrix},$$

$$X_{bc}=\begin{bmatrix}1&1&1&0&0&0&0\\0&0&0&1&1&1&0\\0&0&0&0&0&0&1\end{bmatrix},\ C_{bc}=\begin{bmatrix}2&1&0\\0&1&2\\0&0&1\end{bmatrix},$$

$$X_{bd}=\begin{bmatrix}1&0&0&0&0&0&0\\0&1&1&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&1&0\\0&0&0&0&0&0&1\end{bmatrix},\ C_{bd}=\begin{bmatrix}1&0&0\\1&1&0\\0&1&0\\0&0&2\\0&0&1\end{bmatrix},$$

$$X_{cd}=\begin{bmatrix}1&0&0&1&0&0&0\\0&1&1&0&0&0&0\\0&0&0&0&1&1&0\\0&0&0&0&0&0&1\end{bmatrix},\ C_{cd}=\begin{bmatrix}1&1&0\\1&1&0\\0&0&2\\0&0&1\end{bmatrix}.$$

H_1、H_2 的信息如表 3 所示.

表 3　粒度 $w=2$

条件属性	H_1	H_2
$\{ab\}$	2	0
$\{ac\}$	0	0
$\{ad\}$	1	0
$\{bc\}$	0	0
$\{bd\}$	2	0
$\{cd\}$	0	0

分析表 3 可知, 不存在 $H_2=1$, 因此只需考察 H_1. 由 H_1 的大小确定规则约简顺序为 $\{ab\}\to\{bd\}\to\{ad\}$. $\{ab\}$ 能约简规则 $\{1,2\}$, $\{bd\}$ 能约简规则 $\{1,4\}$, $\{ad\}$ 能约简规则 $\{2\}$, 因为 $\{ab\}$ 已能约简规则 $\{2\}$, 故 $\{ad\}$ 无需参与约简. 先约简条件属性 $\{ab\}$, 可得决策规则 $a=1\bigwedge b=0\to e=1$ (覆盖规则 $\{1,2\}$). 约简条件属性 $\{bd\}$, 可得决策规则 $b=1\bigwedge d=1\to e=0$ (覆盖规则 $\{4\}$).

至此, 规则已覆盖整个论域, 规则挖掘过程结束.

Step 3　输出决策规则

$$a=1\bigwedge b=0\to e=1,\ a=0\to e=0,$$

$$b=1\bigwedge d=1\to e=0,\ d=2\to e=2.$$

更进一步地, 属性 $\{c\}$ 对于论域中每个对象的属性值在约简过程中均被约去, 因而属性 $\{c\}$ 可被约去, 由此可得表 1 的属性约简为 $\{a,b,d\}$, 与文献 [4] 的结果一样.

4.1.2　算例分析

文献 [4] 先对表 1 进行属性约简, 得到 $\{a,b,d\}$, 再逐条对每条规则进行值约简, 最终得到 4 条最简规则. 与文献 [4] 相比, 本文算法无需进行属性约简, 直接进行规则约简并生成最简规则.

文献 [6] 按决策属性分类将表 1 分成等价的 3 个子决策表, 借助核属性和属性频率函数分析各类别特征矩阵, 最终导出 4 条最简规则. 与文献 [6] 提出的基于粗糙集理论的最简决策规则挖掘算法相比, 本文算法在规则总长度上更小.

文献 [7] 从条件属性构成的不完备格出发, 在全粒空间上寻找最大粒, 最终也能得到 4 条最简规则. 但在全粒空间上寻找最大粒, 增加了不必要的计算量. 与文献 [7] 提出的基于最大粒的规则获取算法相比, 本文算法利用启发式信息减少了更多的冗余计算量, 在细粒度下结合 H_1、H_2 提供的信息优先约简区分能力大的信息粒, 这样可以保证规则挖掘在最小的属性集上进行.

4.2　UCI 数据集测试与分析

4.2.1　UCI 数据集测试

为了进一步验证本文算法的有效性, 对 Rosetta 软件中的 Iris, Australian 和 HSV 等 UCI 数据集进行测试. 首先, 利用 Rosetta 软件中的 BROrthogonalScaler (BROS) 和 Equal Frequency Scaler (EFS) 对数据集中连续属性数据进行离散化; 然后, 利用本文提出的规则挖掘算法对经过离散化处理的数据集挖掘规则, 并与文献 [5-7] 中的算法进行比较, 结果如表 4 所示.

从表 4 可以看出, 本文算法与文献 [5-7] 中的算法相比, 挖掘的规则在规则数量和规则长度上都要简洁得多, 其主要原因是本文算法是一种例化方向的算法, 规则挖掘是在一个粒度由粗到细的粒度空间上进行的, 获得的规则在数量和长度上都有一定的优势.

表 4　规则挖掘算法比较

数据集	实例数	离散方法	相容	文献 [5] 算法		文献 [6] 算法		文献 [7] 算法		本文算法	
				数量	长度	数量	长度	数量	长度	数量	长度
Iris	150	BROS	是	10	25	9	28	12	26	10	22
		EFS	否	6	11	9	20	7	11	6	9
Australian	690	BROS	是	243	1 005	236	733	282	969	229	789
		EFS	否	213	987	393	1 264	226	816	198	713
HSV	122	BROS	是	76	303	76	561	82	267	73	238
		EFS	否	73	282	81	316	81	227	72	201

4.2.2 算法复杂性分析

本文算法主要考虑如何提高现有算法的计算效率, 包括如何减少冗余计算, 如何提高搜索效率, 如何减少存储空间.

本文按照启发式信息H_1、H_2对信息粒进行约简, 同时去掉冗余属性, 减少了传统先约简属性再约简属性值时的冗余计算. 在同一粒度空间下进行搜索时使用启发式算子对不同知识空间进行选择和排序, 提高了搜索效率. 在最坏的情况下需要搜索$2^{|C|}$次, 而在实际情况中, 当数据本身的冗余性很大时, 搜索空间要远远小于$2^{|C|}$, 因为在该算法中加入启发式信息, 同时设置终止条件, 算法收敛更快. 本文使用的矩阵是布尔稀疏矩阵, 空间复杂度在最坏情况下为$O(|C||U|)$.

5 结 论

本文提出了一种基于粒计算的最简决策规则挖掘算法, 与传统的先约简属性再提取规则的方法不同, 首先定义了粒矩阵和两个启发式算子, 然后在不同粒度的知识空间中依据启发式信息对不同信息粒进行约简并设定终止条件, 最终得到所有最简决策规则. 理论分析、算例和UCI数据集测试结果表明, 该算法在规则数量和规则长度上有较好的优势.

本文的创新之处在于: 1) 不需要对决策信息系统进行属性约简, 也不需要逐行进行属性值约简, 而是直接对信息粒进行规则约简; 2) 在挖掘规则的过程中建立一种分层递阶的知识粒度空间; 3) 根据粒矩阵提供的信息能一次性地完成规则的挖掘, 不需要进行反向检查, 而且矩阵运算简便; 4) 在规则挖掘的过程中能同时得到属性约简结果. 不足之处在于, 这种算法目前只适用于一致决策信息系统. 如何将其拓展到不完备、多目标决策以及相容决策信息系统, 当样本数据集增大时如何减小存储空间, 以及如何在不同粒层的知识空间中找到一种快速的搜索策略, 进一步减小算法复杂性将是下一步的工作方向.

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: 71-78.

[3] 刘少辉, 盛秋戬, 吴斌, 等. Rough集高效算法的研究[J]. 计算机学报, 2003, 26(5): 524-529.
(Liu S H, Sheng Q J, Wu B, et al. Research on efficient algorithms for rough set methods[J]. Chinese J of Computer, 2003, 26(5): 524-529.)

[4] 常犁云, 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 L Y, Wang G Y, Wu Y. An approach for attribute reduction and rule generation based on rough set theory[J]. J of Software, 1999, 10(11): 1206-1211.)

[5] 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 J H, Pan Y H. Algorithm for acquisition of decision rules based on classification consistency rate[J]. Control and Decision, 2004, 19(10): 1086-1090.)

[6] 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.
(Qian J, Meng X P, Liu D Y, et al. A mining algorithm for concise decision rules based on rough sets theory[J]. Control and Decision, 2007, 22(12): 1368-1372.)

[7] 张清华, 王国胤, 刘显全. 基于最大粒的规则获取算法[J]. 模式识别与人工智能, 2012, 25(3): 388-396.
(Zhang Q H, Wang G Y, Liu X Q. Rule acquisition algorithm based on maximal granule[J]. Pattern Recognition and Artificial Intelligence, 2012, 25(3): 388-396.)

[8] Lin T Y. Granular computing: Practices, theories, and future directions[M]. New York: Springer, 2009: 4339-4355.

[9] 苗夺谦, 范世栋. 知识的粒度计算及其应用[J]. 系统工程理论与实践, 2002, 22(1): 48-56.
(Miao D Q, Fan S D. The calculation of knowledge granulation and its application[J]. System Engineering-Theory & Practice, 2002, 22(1): 48-56.)

[10] Chen Zehua, Xie Gang, Yan Gaowei, et al. Application of a matrix-based binary granular computing algorithm in RST[C]. Proc of IEEE Int Conf on Granular Computing. Beijing: IEEE Press, 2005: 409-412.

[11] 陈泽华, 曹长青, 谢刚. 基于粒矩阵的多变量真值表快速约简算法[J]. 模式识别与人工智能, 2013, 26(8): 745-750.
(Chen Z H, Cao C Q, Xie G. Granular matrix based rapid reduction algorithm for multivarible truth table[J]. Pattern Recognition and Artificial Intelligence, 2013, 26(8): 745-750.)

(责任编辑: 闫 妍)